



مقرر مقدمة في البرمجة (ET 212)

برنامج تكنولوجيا التعليم الرقمي
المستوى الثاني - الفصل الدراسي الأول

إعداد أ.م.د. / محمد ضاحي محمد توني

أستاذ مساعد تكنولوجيا التعليم

كلية التربية النوعية - جامعة المنيا

اسم الكتاب:

مقدمة في البرمجة

(ET 2012)

إعداد:

أ.م.د / محمد ضاحي توني

أستاذ مساعد تكنولوجيا التعليم

كلية التربية النوعية

جامعة المنيا

رقم الإيداع والترقيم الدولي:

I.S.B.N:

جميع الحقوق محفوظة

جميع حقوق الطبع محفوظة لا يجوز نشر أي جزء من هذا الكتاب أو إعادة
طبعه أو تصويره أو اختزان مادته العلمية بأي صورة دون موافقة كتابية

لائحة مقرر مقدمة في البرمجة

أولاً - لائحة المقرر:

التفاصيل	البند
ET 212	الرمز الكودي
مقدمة في البرمجة	اسم المقرر
المستوى الثاني	الفرقة / المستوى
تكنولوجيا التعليم الرقمي	القسم
مركز التعليم المدمج بالتعاون مع كلية التربية النوعية	الكلية
جامعة المنيا	الجامعة
الجانب النظري: 1 ساعة (1 ساعة معتمدة)	عدد ساعات التدريس / الوحدات الدراسية
الجانب العملي: 4 ساعة (2 ساعة معتمدة)	
المجموع: 5 ساعات (3 ساعات معتمدة)	
أعمال السنة (تفعيل المقرر الالكتروني): 30 درجة	توزيع الدرجات
العملي / التطبيقي: 30 درجة	
التحريري: 40 درجة	
المجموع: 100 درجة	
ساعتين	زمن الإمتحان

ثانياً - وصف المقرر والهدف العام:

يهدف المقرر إلى تزويد المتعلم بالمعارف والمهارات الأساسية للبرمجة باللغات عالية المستوى، والمهارات اللازمة لبناء البرنامج (البرمجة)، مع التطبيق على أحد لغات البرمجة، والمساهمة في تطوير المهارات والقدرات الذاتية للمتعلمين فيما يخص استيعاب التطور السريع في مجال الحاسبات أثناء الخدمة

ثالثاً - الأهداف العامة للمقرر:

- 1- إكساب الطلاب المعارف والمفاهيم الأساسية الخاصة بالبرمجة
- 2- إكساب الطلاب المعارف والمفاهيم الأساسية الخاصة بالتعامل مع بيئات التطوير البرمجية لأحد لغات المستوى العالي
- 3- إكساب الطلاب المعارف والمفاهيم الأساسية الخاصة بكتابة الأكواد البرمجية لأحد لغات المستوى العالي
- 4- إكساب الطلاب المعارف والمفاهيم الأساسية الخاصة ببرمجة البيئات الرسومية لأحد لغات المستوى العالي

رابعاً - متطلبات دراسة المقرر:

- 1- يتطلب دراسة المقرر الإلمام بالمهارات الأساسية للتعامل مع نظام تشغيل نظام ويندوز.

- 2- اجتياز مقرر ET121 (أنظمة الحاسب - المستوى الأول)

مقدمة عامة للمقرر والكتاب

يهدف هذا المقرر في مجمله إلى اكساب طالب تكنولوجيا التعليم الرقمي المهارات الأساسية في مجال البرمجة، والتي تعد مكوناً أساسياً من مكونات إعداده في سوق العمل، سواء عمل في مجال التدريس في المدارس، أو في مجال التدريب في أحد المراكز المتخصصة، أو كأخصائي تكنولوجيا معلومات IT في أي مؤسسة، ولتحقيق هذا الهدف يتناول المقرر ما يلي:

الباب الأول - مقدمة في البرمجة:

يتناول الباب الأول المعارف والمفاهيم الأساسية الخاصة بالبرمجة، يتكون الباب من فصل واحد على النحو التالي:

• الفصل الأول - مقدمة في البرمجة:

يتناول الفصل مجموعة من المفاهيم الأساسية في البرمجة وهي: (مفهوم البرمجة، أنواع لغات البرمجة، أهمية تعلم لغات البرمجة في مجال تكنولوجيا التعليم، أشهر لغات البرمجة في العالم ، مقدمة في لغة بايثون ، مميزات لغة بايثون في البرمجة ، مميزات لغة بايثون في تكنولوجيا التعليم ، متطلبات العمل بلغة بايثون).

الباب الثاني - بيئات التطوير البرمجية:

يتناول الباب الثاني المعارف والمفاهيم الأساسية الخاصة بالتعامل مع بيئات التطوير البرمجية لأحد لغات المستوى العالي، يتكون الباب من الفصلين التاليين:

• الفصل الأول - بيئة تطوير بايثون في نظام ويندوز:

يتناول الفصل الأول من الباب الثاني مهارات تسطيب لغة بايثون في نظام تشغيل ويندوز، وكيفية التعامل مع بيئة تطوير PyCharm والتي تمكن المبرمج من برمجة المشروعات بلغة بايثون.

• الفصل الثاني - بيئة تطوير بايثون في نظام أندرويد:

يتناول الفصل الثاني من الباب الثاني المعارف والمهارات الخاصة بانتاج مشروعات برمجية في لغة بايثون باستخدام جهاز الموبايل وبدون استخدام جهاز الكمبيوتر، من خلال الامام بكيفية تسطيب بيئة التطوير Pydroid والتي تحتوي بداخلها على مفسر للغة البايثون وبيئة تطوير متكاملة لا تقل جودة عن بيئات التطوير في أجهزة الكمبيوتر.

الباب الثالث – برمجة واجهات CLI:

يتناول الباب الثالث المعارف والمفاهيم الأساسية الخاصة بكتابة الأكواد البرمجية لأحد لغات المستوى العالي، حيث يبدأ أي مبتدئ في مجال البرمجة بالتعامل مع بيئة CLI (Command Line Interface) وهي بيئة برمجية يتم فيها كتابة الاكواد على شكل أسطر متتالية ويكون الناتج تطبيق Console يشبه شاشة نظام التشغيل DOS، يتكون هذا الباب من السبع فصول التالية:

• الفصل الأول – المتغيرات البرمجية **Variables**:

يتناول الفصل الأول من الباب الثالث المفاهيم والمهارات الخاصة بكتابة أكواد المتغيرات البرمجية في لغة بايثون، وهي أولى خطوات تعلم أي لغة برمجة.

• الفصل الثاني – الإدخال والإخراج البرمجي **Input/Output**:

يتناول الفصل الثاني من الباب الثالث المفاهيم والمهارات الخاصة بكتابة أكواد الإدخال والإخراج في لغة بايثون، حيث يتكون أي برنامج من ثلاث مكونات (إدخال – عمليات – إخراج).

- الفصل الثالث - العوامل الحسابية وعوامل الاسناد

:Assingment Operations

يتناول الفصل الثالث من الباب الثالث المفاهيم والمهارات الخاصة بكتابة أكواد العوامل الحسابية وعوامل الاسناد في لغة بايثون، وهي هامة في تنفيذ المعادلات الحسابية والمنطقية في العمليات البرمجية.

- الفصل الرابع - الشرط والتفرع البرمجي **:Branching**

يتناول الفصل الرابع من الباب الثالث المفاهيم والمهارات الخاصة بكتابة أكواد الشرط والتفرع البرمجي في لغة بايثون، حيث يعد الكود البرمجي (IF) كود عالمي موجود في جميع لغات البرمجة بكافة أشكالها.

- الفصل الخامس - الحلقات البرمجية **:Loops**

يتناول الفصل الخامس من الباب الثالث المفاهيم والمهارات الخاصة بكتابة أكواد الحلقات التكرارية والتكرار البرمجي في لغة بايثون، حيث يعد الكود البرمجي (For) كود عالمي - اسوة بـ If - موجود في جميع لغات البرمجة بكافة أشكالها.

• الفصل السادس – الدوال البرمجية **Functions**:

يتناول الفصل السادس من الباب الثالث المفاهيم والمهارات الخاصة بكتابة أكواد الدوال البرمجية في لغة بايثون، وهي مهارة متقدمة في أي لغة برمجة.

• الفصل السابع – البرمجة الكائنية الشيئية **Object**

:Oriented

يتناول الفصل السابع من الباب الثالث المفاهيم والمهارات الخاصة بكتابة أكواد الدوال البرمجية في لغة بايثون، وهي تعد أعلى مهارات البرمجة، ففيها يتحول المبرمج من مستوى المبتدئ إلى مستوى المتوسط في اللغة التي يدرسها.

الباب الرابع – برمجة واجهات **GUI**:

يتناول الباب الرابع المعارف والمفاهيم الأساسية الخاصة ببرمجة البيئات الرسومية لأحد لغات المستوى العالي، وهي الخطوة التالية لبرمجة واجهات CLI، حيث يتم اكساب المتعلم مهارات توظيف الأدوات الرسومية مثل (الأزرار، مربعات النص ...) في إنتاج برنامج متكامل يظهر للمستخدم بشكل رسومي GUI (Graphical User Interface) ، يتكون هذا الموديول من الدروس التالية :

- **الفصل الأول - مقدمة في الواجهات الرسومية:**

يتناول الفصل الأول من الباب الرابع المفاهيم والمهارات الأساسية الخاصة بالعمليات على الواجهات الرسومية، مع الإلمام بالأكواد الأساسية الخاصة بالأداة Form الأساسية لكتابة أي مشروع برمجي

- **الفصل الثاني - أدوات الإدخال والإخراج:**

يتناول الفصل الثاني من الباب الرابع المفاهيم والمهارات الأساسية الخاصة بالعمليات على أدوات الإدخال والإخراج الأساسية في الواجهات الرسومية، وهي الأدوات التي لاغني عنها في أي مشروع برمجية، مثل أداة مربع النص Entry الخاصة بالإدخال، أداة الأخراج Label، أداة Button لتنفيذ الحدث البرمجي

- **الفصل الثالث - الأحداث البرمجية:**

يتناول الفصل الثالث من الباب الرابع المفاهيم والمهارات الأساسية الخاصة بالأحداث البرمجية في لغة بايثون، وبعد الانتهاء من هذا الدرس سيتمكن المتعلم من إنتاج مشروع برمجي متكامل من حيث الاكواد البرمجية والواجهات الرسومية.

الباب الأول

مفاهيم أساسية في البرمجة

الفصل الأول

مفاهيم أساسية في البرمجة

مفهوم البرمجة

أنواع لغات البرمجة

أهمية تعلم لغات البرمجة في تكنولوجيا التعليم

أشهر لغات البرمجة في العالم

ماهية لغة بايثون

مميزات لغة بايثون

مميزات لغة بايثون في تكنولوجيا التعليم

متطلبات العمل بلغة بايثون

مقدمة الفصل

أولاً - الهدف العام:

إكساب طلاب تكنولوجيا التعليم الرقمي المفاهيم المفاهيم للبرمجة.

ثانياً - الأهداف التعليمية:

بعد الانتهاء من هذا الفصل /الدرس، يتوقع من الطالب أن يكون قادراً على أن:

- 1- يذكر مفهوم البرمجة
- 2- يشرح أنواع لغات البرمجة
- 3- يفسر أهمية تعلم لغات البرمجة في مجال تكنولوجيا التعليم
- 4- يعدد أشهر لغات البرمجة في العالم
- 5- يشرح ماهية لغة بايثون
- 6- يوضح مميزات لغة بايثون
- 7- يفسر مميزات لغة بايثون في مجال تكنولوجيا التعليم
- 8- يوضح متطلبات العمل بلغة بايثون

مفاهيم أساسية في البرمجة

مفهوم البرمجة

هي عملية كتابة تعليمات للكمبيوتر بهدف توجيهه لتنفيذ أوامر معينة ، لكل لغة برمجة قواعدها الخاصة بها ، بحيث إذا قام المبرمج بكتابة التعليمات بطريقة المحددة والصحيحة ؛ فإن الكمبيوتر سيقوم بما هو متوقع ومطلوب منه وهو تنفيذ التعليمات والوصول للنتائج المطلوبة.

أنواع لغات البرمجة

أولاً - حسب الإستخدام:

1 - لغة الآلة

هي اللغة التي يفهمها الكمبيوتر مباشرة ، وهي تتكون من رقمين (0) و (1) فقط ، وأي لغة يستخدمها البرمج تترجم في النهاية إلى لغة الآلة

2 - لغة التجميع (أسمبلي):

هي لغة وسط بين لغة الآلة وبين لغات الإنسان الطبيعية ، وهي تتضمن كافة الأوامر التي يفهمها الكمبيوتر ، وتتيح التحكم الكامل فيه.

3 - لغات المستوى العالي :

هي باقي لغات البرمجة المعروفة ، التي تستخدم لغة أقرب ما تكون للغة الإنسان الطبيعية ، تتميز بالسهولة عن لغة التجميع في الكتابة ، إلا أنها أقل من لغة التجميع في التحكم بالكمبيوتر .

ثانياً - حسب المستوى:

1 - Desktop Application

لغات تستخدم لكتابة البرامج الخاصة بالعمل على أجهزة الكمبيوتر المكتبية.

2 - Web Application

لغات تستخدم لتصميم المواقع الإلكترونية التفاعلية

3 - Mobile Application

لغات تستخدم لتصميم تطبيقات الأجهزة المحمولة

4 - Data_Base Application

لغات تستخدم لتصميم قواعد البيانات الإلكترونية

5 - Data Application

لغات تستخدم للتعامل مع البيانات مثل التنقيب والبيانات الضخمة

أهمية تعلم لغات البرمجة في تكنولوجيا التعليم

- تأهيل خريج تكنولوجيا التعليم بتدريس البرمجة ضمن مقررات الحاسب الآلي في المدارس
- تأهيل خريج تكنولوجيا التعليم للعمل كـ (IT) في الشركات والمؤسسات
- تأهيل خريج تكنولوجيا التعليم مؤهلاً لتدريب البرمجة في المعاهد ومراكز التدريب
- إكساب خريج تكنولوجيا التعليم المهارات الأساسية لتصميم البرمجيات التعليمية على مختلف المنصات ونظم التشغيل

أشهر لغات البرمجة في العالم

يوجد في مجال الكمبيوتر الكثير من لغات البرمجة ، ومن أشهرها ما يلي:

1 - لغة جافا JAVA

- من لغات البرمجة عالية المستوى ، إبتكرها James Gosling عام 1991
- تعمل على جميع نظم التشغيل مثل (ويندوز ، لينكس ، أندرويد)
- تم تطويرها لبناء أي تطبيقات مثل : الكمبيوتر والويب والموبايل والسيرفرات والروبوتات
- تقوم بالإشراف على تطويرها شركة Sun Microsystems

2 - لغة سي بلس بلس C++

- من لغات البرمجة عالية المستوى ، إبتكرها Bjarne Stroustrup عام 1979
- من أقوى اللغات والتي تستخدم في برمجة أي تطبيق لأي غرض
- إستخدمت في برمجة الأجهزة المحمولة ، تطبيقات أدوبي ، تطبيقات جوجل ، مايا
- لغة مجانية ولها إصدارات مفتوحة المصدر
- تقوم بالإشراف على تطويرها لجنة معايير C++

3- لغة سي شارب C#

- من لغات البرمجة عالية المستوى ، تم ابتكارها بقيادة Anders Hejlsberg عام 2000
- من أقوى اللغات والتي تستخدم في برمجة أي تطبيق لأي غرض
- هي الخيار الأول لمطوري تطبيقات ويندوز ، وتستخدم في قواعد البيانات وبرمجة الويب
- لغة مجانية ، ولكن قد تحتاج لدفع تكلفة للاستخدام التجاري
- تأتي ضمن حزمة ميكروسوفت فيجوال ستوديو التابعة لشركة Microsoft

4- لغة سكوال SQL

- صمم مبادئها العالم Edgar Frank عام 1970 ، وأنتجت IBM أول بيئة تطوير
- مصممة خصيصا لبناء قواعد البيانات ، ولا تستخدم في بناء التطبيقات التقليدية
- من أشهر بيئات تطويرها : Oracle ، MySql ، Ms SQL Server ، IBMDB2
- MySQL مجانية ومفتوحة المصدر ، أما باقي البيئات تحتاج لرخصة استخدام
- تصدر اللجنة الأمريكية للمعايير إصدارات سنوية للغة آخرها الإصدار SQL2008

ماهية لغة بايثون

- إبتكرها Guido Van Rossum عام 1986 ، وظهر الإصدار المتداول عام 1991
- تعمل على جميع نظم التشغيل مثل (ويندوز ، لينكس ، أندرويد)
- تم تطويرها لبناء أي تطبيقات مثل : الكمبيوتر والويب والموبايل والسيرفرات والروبوتات
- لغة مجانية ومفتوحة المصدر

- تولى فان روسن إدارة مشروع بايثون حتى عام 2018 ، وأصبحت لغة بايثون تدار من قبل مجلس توجيه مكون من خمس أشخاص.

مميزات لغة بايثون

- مجانية تماماً ، فهي مفتوحة المصدر
- يمكن البرمجة بها حتى بإستخدام أجهزة الكمبيوتر القديمة ، وحتى بإستخدام الموبايل
- من أسهل لغات البرمجة تعلماً
- تستخدم في برمجة أي نوع من التطبيقات (كمبيوتر ، ويب ، موبايل ، قواعد بيانات) ، لدرجة أن موقع يوتيوب الشهير مصمم أساساً بإستخدام لغة بايثون
- أغلب لغات البرمجة تدعم بايثون ، أي أن المبرمج يمكنه كتابة كود بايثون داخل بيئة أخرى مثل جافا

مميزات لغة بايثون في تكنولوجيا التعليم

- لغة سهلة القواعد ، ولا تتطلب مجهوداً كبيراً في تعلمها
- لغة معتمدة في برمجة البيانات الضخمة Big Data
- لغة معتمدة في برمجة التعلم العميق Deep Learning
- يمكن إستخدامها في بناء تطبيقات تعليمية على أي منصة وعلى أي نظام تشغيل

- الكثير من المعاهد والمدارس تقوم بتدريس لغة بايثون لطلابها

متطلبات العمل بلغة بايثون

يقوم بتحويل الكود المصدري (Source Code) الذي كتبه المبرمج بإستخدام أي محرر نصوص (Word، Notepad) والمحفوظ بالإمتداد .py ، إلى لغة الآلة (Byte Code) .

أ - مترجم اللغة / مشغل بايثون (Python Interpreter):

- يقوم بتحويل الكود المصدري (Source Code) الذي كتبه المبرمج بإستخدام أي محرر نصوص (Word ، Notepad) والمحفوظ بالإمتداد .py ، إلى لغة الآلة (Byte Code) .
- يتم تحميله من الموقع الرسمي لبايثون.
- يعد الاصدار 3.9.0 هو الأحدث حتى الآن (عام 2020).
- يبلغ حجمه حوالي 27 ميغا بايت.
- يتم تحميل المترجم مجاناً بدون أي قيود أو ترخيص من أي نوع.

ب - بيئة تطوير:

- برامج توفر للمبرمجين إمكانية كتابة الكود المصدري ثم حفظه في ملف بالإمتداد .py.

- تتميز عن أي محرر نصوص تقليدي : بأنها توفر أدوات متكاملة لكتابة الكود وتصحيحه وتجربته وبناء الواجهات الرسومية بشكل أسهل وأكثر احترافية .
- يتوفر العديد من البيئات ، ومن أشهرها بيئة (PyCharm) الخاصة بنظم تشغيل ويندوز ، وبيئة (Pydroid 3) الخاصة بنظم تشغيل أندرويد ، وهما بيئات التطوير التي سندرسها في المحاضرات القادمة

نشاط تعليمي على مفاهيم أساسية في البرمجة

البيان	الوصف
رقم النشاط	1
عنوان النشاط	المقارنة بين لغة فيجوال بيسيك نت وبين لغة بايثون وأهمية كل منهما في مجال تكنولوجيا التعليم الرقمي
وصف النشاط	يتم فتح باب المناقشة في المقارنة بين لغة فيجوال بيسيك نت وبين لغة بايثون وأهمية كل منهما في مجال تكنولوجيا التعليم الرقمي
توقيت التنفيذ	خلال المحاضرة
طريقة التنفيذ	تعليم تعاوني - مجموعات صغيرة

تكليفات برمجية

قم بتنفيذ التكليف التالي، ثم تواصل مع أستاذ المادة لتسليمه ومناقشته في المحاضرات التالية:

التكليف:

اكتب تقرير موجز عن أهمية لغة بايثون في مجالي البيانات الضخمة والتعلم الآلي

مصادر إثرائية من بنك المعرفة المصري



Egyptian Knowledge Bank
بنك المعرفة المصري

من خلال المصادر الالكترونية التالية والموجودة ببنك المعرفة المصري والمكتبة الرقمية؛ يمكنك الاستزادة حول المحتوى العلمي للفصل:

م	عنوان المصدر	الرابط
1	كتاب عن البرمجة والتخطيط	https://0810gvb69-1103-y-https-search-mandumah-com.mplbci.ekb.eg/Record/82232
2	بحث عن توظيف البرمجة في مجال الامتحانات والتقييم	https://0810gvb69-1103-y-https-search-mandumah-com.mplbci.ekb.eg/Record/1008457

قاموس المصطلحات الواردة بالفصل

م	المصطلح بالعربية	المصطلح بالإنجليزية	الوصف
1	البرمجة	Programing	هي عملية كتابة تعليمات للكمبيوتر بهدف توجيهه لتنفيذ أوامر معينة
2	لغة الآلة	Machine Language	هي اللغة التي يفهمها الكمبيوتر مباشرة ، وهي تتكون من رقمين (0) و (1) فقط ، وأي لغة يستخدمها البرمج تترجم في النهاية إلى لغة الآلة
3	لغة التجميع	Assembly	هي لغة وسط بين لغة الآلة وبين لغات الإنسان الطبيعية ، وهي تتضمن كافة الأوامر التي يفهمها الكمبيوتر ، وتتيح التحكم الكامل فيه
4	لغات المستوى العالي	High Level Languages	هي باقي لغات البرمجة المعروفة ، التي تستخدم لغة أقرب ما تكون للغة الإنسان الطبيعية ، تتميز بالسهولة عن لغة التجميع في الكتابة ، إلا أنها أقل من لغة التجميع في التحكم بالكمبيوتر

اختبر نفسك

اختر الإجابة الصحيحة من بين البدائل الموجودة:

1		م
هي عملية كتابة تعليمات للكمبيوتر بهدف توجيهه لتنفيذ أوامر معينة		العبارة
برامج الكمبيوتر	أ	البدائل
برمجة الكمبيوتر	ب	
برنامج الورد	ج	
برامج الحاسب الآلي	د	
2		م
اللغة التي يفهمها الكمبيوتر مباشرة		العبارة
لغة الآلة	أ	البدائل
لغة التجميع	ب	
لغات المستوى العالي	ج	
لغة الانسان	د	
3		م
لغات برمجة تستخدم في كتابة البرامج لمواقع الانترنت		العبارة
Desktop Application	أ	البدائل
Data_Base Application	ب	
Mobile Application	ج	
Web Application	د	

4		م
من لغات البرمجة وتعمل على أي نظام تشغيل		العبرة
C#	أ	البدائل
JAVA	ب	
Visual Basic	ج	
ASP	د	
5		م
أي مما يلي ليس من إمكانيات لغة بايثون		العبرة
تعمل على أي نظام تشغيل	أ	البدائل
مجانية	ب	
مفتوحة المصدر	ج	
مخصصة لقواعد البيانات	د	
6		م
أي مما يلي من مميزات لغة بايثون		العبرة
تنتج قواعد بيانات	أ	البدائل
تعتمد على السيرفرات	ب	
تعتمد على برمجة الويب	ج	
تعمل على أي نظام تشغيل	د	

7		م
من التطبيقات التعليمية عالية المستوى في بايثون		العبارة
Deep Learning	أ	البدائل
Data Base	ب	
Web App	ج	
Mobile App	د	
8		م
يقوم بتحويل كود بايثون إلى لغة الآلة		العبارة
بيئة تطوير بايثون	أ	البدائل
مترجم لغة البايثون	ب	
مضاد الفيروسات	ج	
نظام التشغيل	د	

مفتاح الإجابة

م	العبارة	الإجابة الصحيحة
1	هي عملية كتابة تعليمات للكمبيوتر بهدف توجيهه لتنفيذ أوامر معينة	برمجة الكمبيوتر
2	اللغة التي يفهمه الكمبيوتر مباشرة	لغة الآلة
3	لغات برمجة تستخدم في كتابة البرامج لمواقع الانترنت	Web Application
4	من لغات البرمجة وتعمل على أي نظام تشغيل	JAVA
5	أي مما يلي ليس من إمكانيات لغة بايثون	مخصصة لقواعد البيانات
6	أي مما يلي من مميزات لغة بايثون	تعمل على أي نظام تشغيل
7	من التطبيقات التعليمية عالية المستوى في بايثون	Deep Learning
8	يقوم بتحويل كود بايثون إلى لغة الآلة	مترجم لغة البايثون

الباب الثاني

بيئات التطوير البرمجية

الفصل الأول

بيئة تطوير بايثون في نظام ويندوز

متطلبات بايثون في ويندوز

تسطيب مشغل لغة بايثون

التأكد من نجاح تسطيب بايثون

تسطيب بيئة تطوير بايثون PyCharm

فتح برنامج PyCharm للمرة الأولى

مكونات واجهة PyCharm

اغلق مشروع

إنشاء مشروع جديد في بداية فتح البرنامج

إنشاء مشروع جديد أثناء فتح مشروع آخر

حفظ مشروع لأول مرة

حفظ التعديلات الدورية

تجربة تنفيذ كود برمجي

ملاحظات على كتابة الكود البرمجي

تبدال القالب التصميمي لبيئة PyCharm

كتابة أول برنامج بلغة بايثون وتجربته

مقدمة الفصل

أولاً - الهدف العام:

إكساب طلاب تكنولوجيا التعليم الرقمي المفاهيم والمفاهيم والمهارات الأساسية الخاصة بالعمل ببيئة بتطوير لغة بايثون الخاصة بنظام التشغيل ويندوز.

ثانياً - الأهداف التعليمية:

بعد الانتهاء من هذا الفصل /الدرس، يتوقع من الطالب أن يكون قادراً على أن:

1. يشرح متطلبات لغة بايثون في نظام تشغيل ويندوز
2. يقوم بتثبيت مشغل لغة بايثون في نظام ويندوز
3. ينفذ إجراءات التأكد من نجاح تثبيت بايثون في ويندوز
4. يقوم بتثبيت بيئة تطوير بايثون PyCharm
5. يفتح برنامج PyCharm للمرة الأولى
6. يميز بين مكونات واجهة PyCharm
7. يقوم بإغلاق مشروع في PyCharm
8. ينشئ مشروع جديد في بداية فتح البرنامج
9. ينشئ مشروع جديد أثناء فتح مشروع آخر
10. يحفظ مشروع لأول مرة
11. يحفظ التعديلات الدورية على مشروع
12. يقوم بتجربة تنفيذ كود برمجي
13. يشرح قواعد كتابة الكود البرمجي في بايثون
14. يقوم بتعديل القالب التصميمي لبيئة PyCharm
15. يكتب أول برنامج بلغة بايثون ويجريه

بيئة تطوير بايثون في نظام ويندوز

متطلبات العمل بلغة بايثون في ويندوز

لتنفيذ المشروعات البرمجية بلغة البايثون يتطلب توفير المكونات التاليين في نظام تشغيل ويندوز:

1- مترجم اللغة / مشغل بايثون (Python Interpreter)

- يقوم بتحويل الكود المصدري (Source Code) الذي كتبه المبرمج باستخدام أي محرر نصوص (Word، Notepad) والمحفوظ بالإمتداد py، إلى لغة الآلة (Byte Code).
- يتم تحميله من الموقع الرسمي لبايثون.
- يعد الاصدار 3.9.0 هو الأحدث حتى الآن (عام 2020).
- يبلغ حجمه حوالي 27 ميغا بايت.
- يتم تحميل المترجم مجاناً بدون أي قيود أو ترخيص من أي نوع.

2- بيئة تطوير

- برامج توفر للمبرمجين إمكانية كتابة الكود المصدري ثم حفظه في ملف بالإمتداد py.
- تتميز عن أي محرر نصوص تقليدي : بأنها توفر أدوات متكاملة لكتابة الكود وتصحيحه وتجربته وبناء الواجهات الرسومية بشكل أسهل وأكثر احترافية .

- يتوفر العديد من البيئات ، ومن أشهرها بيئة (PyCharm) والتي سيتم استخدامها في المحاضرات القادمة.
- يتم تحميلها من الموقع الرسمي لـ PyCharm ، والذي يوفر نسخة مجانية (Community) لدارسي البرمجة ، ونسخة مدفوعة (Professional) كاملة الخصائص

تسطيب مشغل لغة بايثون Python Interpreter

يتم تسطيب مترجم البايثون وفق الخطوات التالية

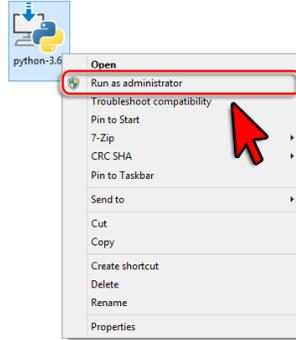
1- الدخول على رابط تحميل المترجم من موقع بايثون الرسمي

<https://www.python.org/downloads/>

2- تحميل الإصدار الاحدث



3- تسطيب البرنامج في وضع Administrator



4- الموافقة على جميع الخيارات والاعدادات الافتراضية التي يقترحها

ملف التسطيب



5- الانتهاء من عملية التسطيب



التأكد من نجاح تسطيب بايثون

يتم تنفيذ هذه الخطوة احتياطياً للتأكد من نجاح تسطيب لغة بايثون بشكل فعل وذلك كما يلي:

1- الدخول على برنامج Command Prompt

```
Command Prompt
Microsoft Windows [Version 6.3.9600]
(c) 2013 Microsoft Corporation. All rights reserved.
C:\Users\user harmash>
```

2- كتابة كلمة Python ثم الضغط على زر Enter

```
Command Prompt - python.exe
> python
Python 3.6.4 (v3.6.4:d48eceb, Dec 19 2017, 06:04:45) [MSC v.1900 32 bit (Intel)] on win32
Type "help", "copyright", "credits" or "license" for more information.
>>>
```

إذا ظهرت رسالة توضح اصدار البايثون الموجود هذا يعني نجاح تسطيب اللغة، وإذا لم تظهر يدل هذا على وجود مشكلة او عدم تسطبيه من الأساس.

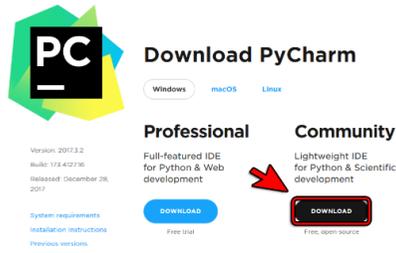
تسطيب بيئة تطوير بايثون

يتم تسطيب بيئة تطوير البايثون (تطبيق PyCharm) وفق الخطوات التالية:

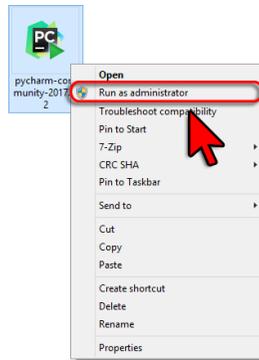
1- الدخول على رابط تحميل البيئة من موقع PyCharm الرسمي:

<https://www.jetbrains.com/pycharm/download/>

2- تحميل الإصدار المجاني Community

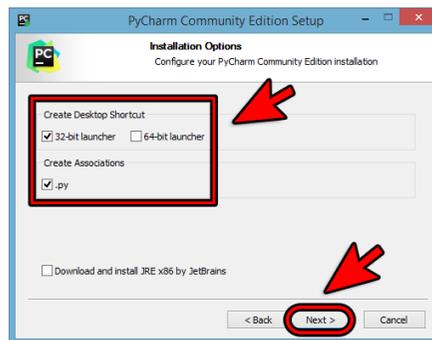


3- تسطيب البرنامج في وضع Administrator



4- الموافقة على جميع الخيارات والاعدادات الافتراضية التي يقترحها

ملف التسطيب



5- الانتهاء من عملية التسطيب



فتح برنامج PyCharm للمرة الأولى

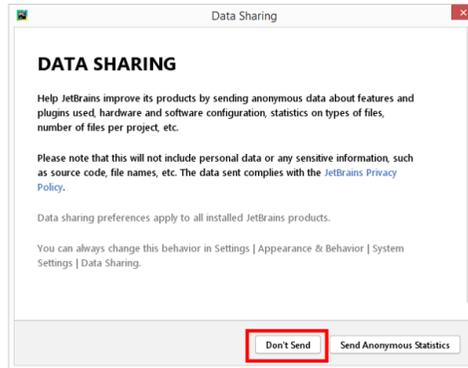
عند فتح التطبيق للمرة الأولى (سواء باستخدام ايقونة البرنامج على ديسكتوب أو من قائمة Start) تطلب البيئة من المستخدم تحديد مجموعة من الخيارات، ولا تسأل عنها البيئة مرة أخرى ، جميع الخيارات يمكن تعديلها فيما بعد من الاعدادات داخل التطبيق فلاتوجد مشكلة في أن يغير المستخدم رأيه في أي مما اختاره عند الفتح للمرة الأولى

1- حقوق الملكية الفكرية



هي رسالة توجد في كل التطبيقات ، تعطي المستخدم معلومات شاملة عن حقوق الملكية الفكرية للتطبيق

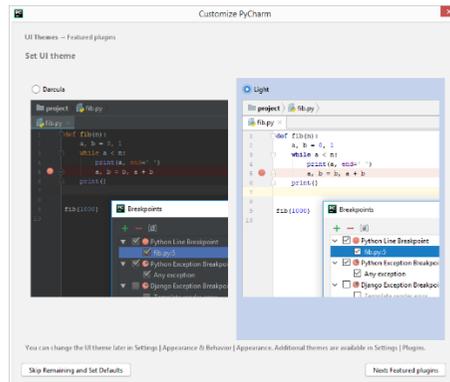
2- طلب تحديد موقف المستخدم من مشاركة البيانات



نظرا لأن التطبيق مفتوح المصدر ومجاني، فإنه يشجع المستخدم على مشاركة معلوماته الشخصية ، وإذا لم تكن تمتلك فكرة واضحة عن مشاركة المعلومات

البيانات قم بالضغط على زر Don't send

3- طلب تحديد قالب التصميم المفضل

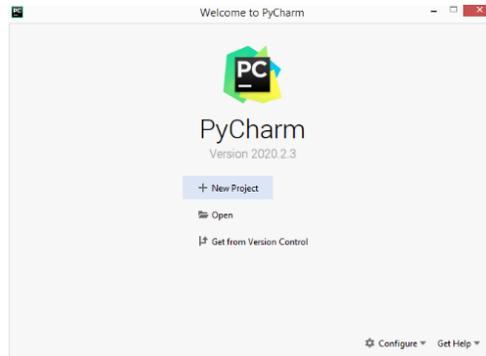


يتيح التطبيق قالبين رئيسين للتصميم ، أحدهما ذو خلفية بيضاء وكتابة سوداء ،
والآخر ذو خلفية سوداء وكتابة بيضاء ، يقوم المستخدم بإختيار القالب المفضل
، ويمكنه تعديله في أي وقت فيما بعد من داخل اعدادات التطبيق.

4- ظهور شعار البيئة معلومات عن الإصدار

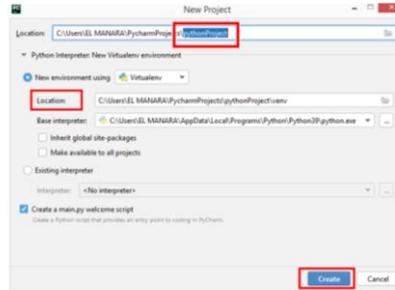


5- ظهور نافذة البداية



لا يتم دخول البيئة إلا بعد أن يحدد المستخدم مايريده من الخيارات الموجودة ،
وهي نافذة تظهر فيما بعد في جميع مرات التشغيل (وليس المرة الأولى فقط)،
لدخول البيئة اضغط **New Project**.

6- ظهور نافذة حفظ المشروع

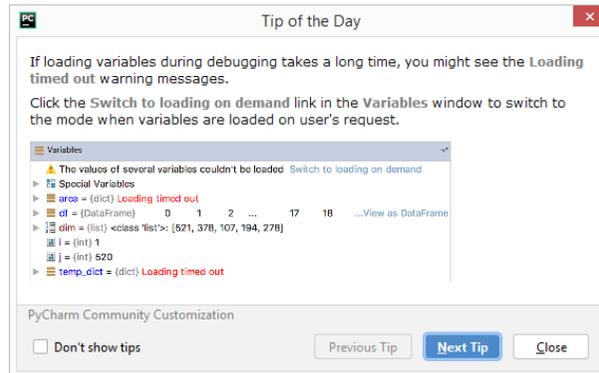


يرتبط أمر **New Project** بإجبار المستخدم على:

- تحديد اسم المشروع البرمجي
- تحديد مكان حفظ ملفات المشروع البرمجي **Location**

بعد تحديدهما، اضغط زر **Create**

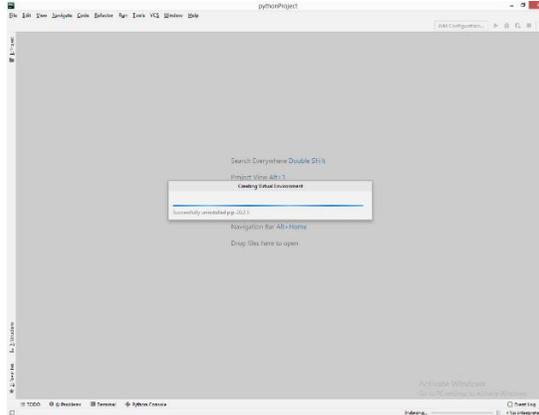
7- ظهور نافذة ملاحظات اثرائية



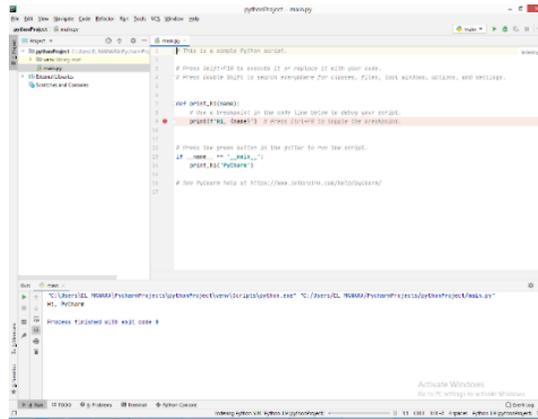
هي نافذة تظهر في كل مرة يتم فيها تشغيل البرنامج ، هي تعطي المستخدم معلومات برمجية وملاحظات اثرائية ، في حالة عدم الرغبة في ظهورها مرة

أخرى يتم الضغط على (Don't Show tips) ، ويمكن ارجاع هذا الاختيار من اعدادات البرنامج الداخلية

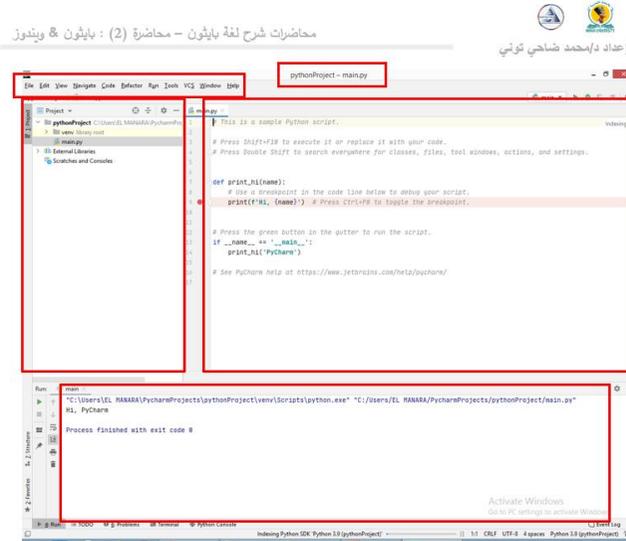
8- انتظار تحميل البيئة (حسب قوة وسرعة جهاز الكمبيوتر)



9- ظهور الشاشة الافتتاحية Interface



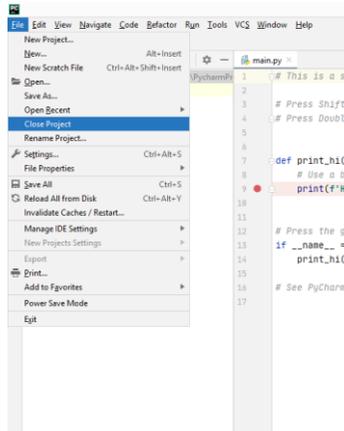
مكونات واجهة PyCharm



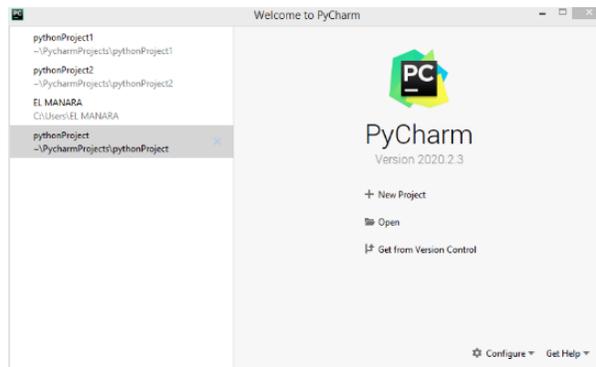
تتكون واجهة بيئة PyCharm من العناصر التالية:

- شريط العنوان (title bar) : أعلى الواجهة ، يظهر به اسم الملف الحالي
- شريط القوائم (File , Edit) : أسفل شريط العنوان ، وبه جميع أوامر البيئة
- شريط الأدوات : أسفل شريط العنوان ، وبه بعض أوامر البيئة
- نافذة (Project) : على اليسار ، يظهر بها جميع ملفات المشروع
- مساحة العمل : على اليمين ، يتم فيها كتابة الكود البرمجي.
- نافذة النتائج : من الأسفل ، يظهر بها ناتج تنفيذ البرنامج اذا تم كتابته بشكل صحيح ، أو رسالة بالاطءاء البرمجية إذا تمت بشكل خاطئ.

اغلاق مشروع



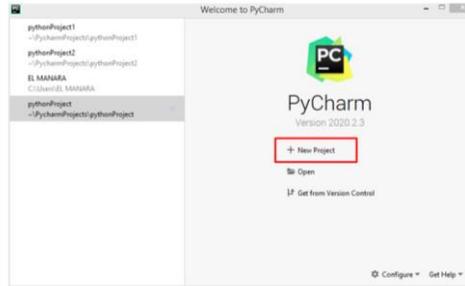
يتم اغلاق مشروع من خلال فتح قائمة File ثم اختيار الامر Close Project ، ستختفي واجهة البرنامج ويحل محلها نافذة البداية



انشاء مشروع جديد

يمكن انشاء مشروع جديد بطريقتين

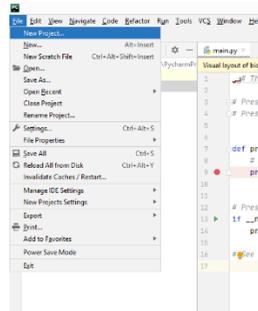
1 - انشاء مشروع جديد عن فتح البرنامج



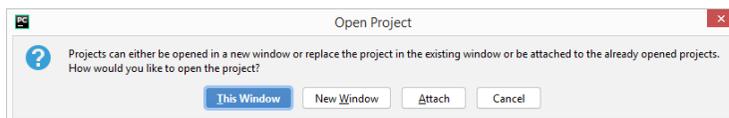
وذلك من خلال نافذة البداية (التي تظهر اجبارياً) عند تشغيل البيئة ، يتم تأكيد انشاء مشروع جديد من خلال اختيار New Project

1 - انشاء مشروع جديد عن اثناء فتح مشروع آخر

اذا كان المستخدم يعمل بالفعل في مشروع برمجي داخل البيئة ، يتم انشاء المشروع الجديد من خلال فتح قائمة file ثم اختيار امر New project



وستظهر نافذة تغير المستخدم اذا كان يريد فتح المشروع في نافذة جديدة (new window) أو فتح المشروع ليحل محل المشروع الحالي في النافذة الحالية (This window)



حفظ مشروع لأول مرة

يتم حفظ المشروع لأول مرة بشكل تلقائي عند انشاء المشروع نفسه من Project ، حيث لاتسمح بالبدء في المشروع إلا بعد أن يقوم المستخدم بتحديد اسم المشروع ومكان التخزين Location .

حفظ التعديلات الدورية على مشروع

لحفظ التعديلات التي يقوم بها المستخدم بشكل دوري على المشروع ، توجد عدة طرق أسهلها الضغط على (Ctrl + S)

تجربة كود برمجي

يتم تجربة كود برمجي بعدة طرق هي:

- الضغط على (Shift + F10)
- فتح قائمة Run ثم اختيار الامر Run
- الضغط على زر Run من شريط القوائم



قواعد كتابة الكود البرمجي في بايثون

توجد بعض القواعد التي تميز كتابة الكود البرمجي في لغة بايثون ، وهي قد تتشابه أو تختلف مع قواعد الكتابة في لغات برمجية أخرى ، وذلك على النحو التالي:

1 - حساسية الأحرف Sensitivity Case

- تقوم لغة بايثون بالترقية بين الكلمات في وضع (Captial - الأحرف الكبيرة) وبين الكلمات في وضع (Small - الأحرف الصغيرة).
- بالتالي كلمة (Go) لا تساوي في بايثون كلمة (go)

2 - عدد سطور كتابة الأوامر

- يفترض بايثون أن الأمر الواحد يكتب في سطر واحد ، أي أن بمجرد الضغط على زر Enter هذا يعني أن السطر الحالي من الكود إنتهى ويتم البدء في سطر جديد من الكود (الحد الأقصى لعدد حروف السطر (79)

$$A = 1$$

$$B = 2$$

3 - كتابة عدة أسطر كود في سطر واحد

- إذا أردت كتابة عدة أسطر كود في سطر واحد ، قم بوضع فاصلة منقوطة (;) بين أسطر الكود

$$A = 1 ; B = 2$$

4 - كتابة كود واحد في عدة أسطر

- إذا أردت كتابة سطر كود واحد وتوزيعه على عدة أسطر فعلية، قم بوضع باك سلاش (\) بين أسطر الكود

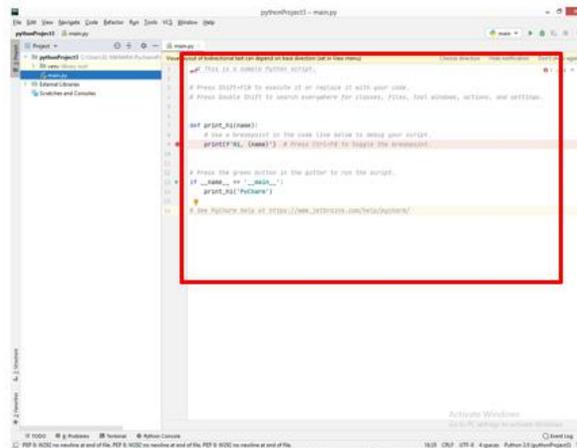
```
A = B + \
+ \      C
+ \      D
```

4 - كتابة التعليقات

- إذا أردت كتابة ملاحظات شخصية حول الكود الذي تكتبه ليزكرك بوظيفة الكود عند الرجوع إليه ، قم بوضع علامة (#) وبعدها أكتب ما تشاء ، سيعرف البرنامج أنها ملاحظة ولن يدخلها في الكود البرمجي هذا تعليق ولا يضاف للكود #

5 - كتابة مشروع جديد:

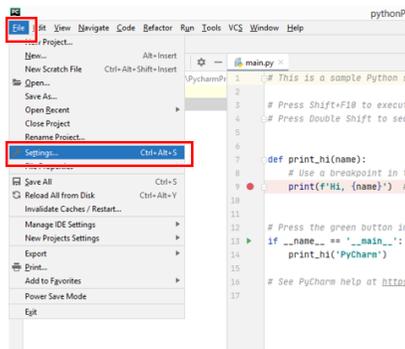
- عند إنشاء مشروع جديد تقوم البيئة بكتابة كود تجريبي في نافذة كتابة الكود ، لذلك يجب مسح الكود التجريبي الموجود لكتابة الكود الجديد المطلوب.



تبديل القالب التصميمي (Theme) واجهة بايثون

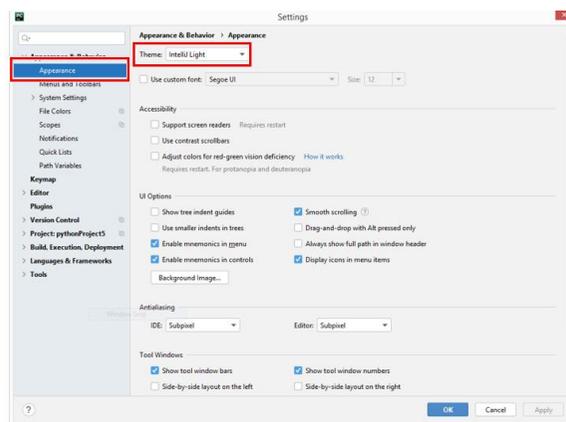
يتم سؤال المستخدم في أول مرة يتم فيها تشغيل البيئة عن القالب التصميمي المفضل، ومن خلال الخطوات التالية يمكن للمستخدم تعديل الاختيار الذي قرره في تلك المرة

1 - فتح قائمة File ثم اختيار أمر Settings

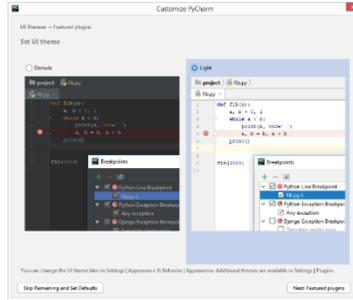


2 - من نافذة Settings يتم اختيار التبويب Appearance ثم

Theme



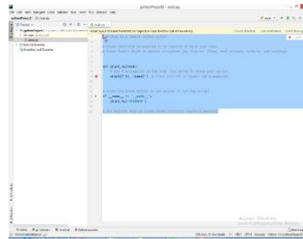
3 - اختيار القالب التصميمي المطلوب



قواعد كتابة الكود البرمجي في بايثون

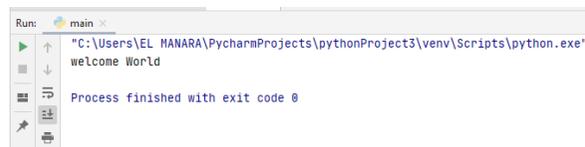
يتم كتابة كود برمجي وتجربته من خلال الخطوات الرئيسية التالية:

1 - مسح الكود التجريبي الموجود بشكل افتراضي في واجهة البيئة



2 - كتابة الكود التالي والذي سيظهر رسالة ترحيبية للمستخدم:

```
print (" Welcome World ")
```

3 - تجربة الكود بالضغط على **Shift + F10**:

نشاط تعليمي على بيئة تطوير بايثون في نظام التشغيل ويندوز

الوصف	البيان
1	رقم النشاط
المقارنة بين بيئة PyCharm وبيئة تطوير ميكروسوفت فيجوال ستوديو	عنوان النشاط
يتم فتح باب المناقشة في الفرق بين بيئة PyCharm وبيئة تطوير ميكروسوفت فيجوال ستوديو	وصف النشاط
خلال المحاضرة	توقيت التنفيذ
تعليم تعاوني - مجموعات صغيرة	طريقة التنفيذ

تكاليفات برمجية

قم بتنفيذ التكليف التالي، ثم تواصل مع أستاذ المادة لتسليمه ومناقشته في المحاضرات التالية:

التكليف:

اكتب تقرير موجز عن بيئات أخرى لبرمجة بايثون في ويندوز بخلاف بيئة PyCharm.

مصادر إثرائية من بنك المعرفة المصري



Egyptian Knowledge Bank
بنك المعرفة المصري

من خلال المصادر الالكترونية التالية والموجودة ببنك المعرفة المصري والمكتبة الرقمية؛ يمكنك الاستزادة حول المحتوى العلمي للفصل:

م	عنوان المصدر	الرابط
1	كتاب عن نظام تشغيل ويندوز 7	https://0811nksi2-1106-y-https-ekbjun-ethraadl-com.mplbci.ekb.eg/Record/6763
2	كتاب عن نظام تشغيل ويندوز 8	https://0811nksi2-1106-y-https-ekbjun-ethraadl-com.mplbci.ekb.eg/Record/8265
3	كتاب عن نظام تشغيل ويندوز 8.1	https://0811nksi2-1106-y-https-ekbjun-ethraadl-com.mplbci.ekb.eg/Record/13668

قاموس المصطلحات الواردة بالفصل

م	المصطلح بالعربية	المصطلح بالإنجليزية	الوصف
1	مترجم اللغة	Interpreter	يقوم بتحويل الكود المصدري (Source Code) الذي كتبه المبرمج باستخدام أي محرر نصوص (Word ، Notepad) ، إلى لغة الآلة (Byte Code) .
2	بيئة تطوير	Programing Envirnomnet	برامج توفر للمبرمجين إمكانية كتابة الكود المصدري ثم حفظه في ملف بالإمتداد الخاص بمترجم اللغة
3	مترجم بايثون	Python Interpreter	مترجم اللغة الخاص بلغة بايثون ويتم تحميله من موقع بايثون الرسمي
4	بايتشارم	PyCharm	بيئة تطوير لها اصدار مجاني ، توفر إمكانية كتابة الكود المصدري بالإمتداد py الذي يستطيع مترجم بايثون تحويله للغة الآلة

اختبر نفسك

اختر الإجابة الصحيحة من بين البدائل الموجودة:

1	م
يقوم بتحويل كود بايثون إلى لغة الآلة في نظام تشغيل ويندوز	العبارة
مترجم Python	أ
بيئة تطوير Pycharm	ب
بيئة تطوير Pydroid	ج
نظام Windows	د
2	م
يتم تحميل لغة بايثون من	العبارة
موقع Pycharm	أ
موقع Windows	ب
موقع Android	ج
موقع Python	د
3	م
يتم التأكد من نجاح تسطيب بايثون عن طريق	العبارة
Start Menu	أ
Command Prompt	ب
Windows	ج
Restart	د

4		م
ما هو الإصدار المجاني الخاص ببيئة Pycharm		العبارة
Professional	أ	البدائل
Community	ب	
Free	ج	
Trial	د	
5		م
من متطلبات تشغيل البرنامج للمرة الأولى، خاص بمشاركة البيانات		العبارة
Open	أ	البدائل
Policy	ب	
Theme	ج	
Data Sharing	د	
6		م
من مكونات واجهة Pycharm ويظهر فيه اسم الملف الحالي		العبارة
شريط الادوات	أ	البدائل
Title Bar	ب	
نافذة Project	ج	
مساحة العمل	د	

7		م
يمكن اغلاق مشروع في Pycharm من خلال قائمة		العبارة
Edit	أ	البدائل
Run	ب	
File	ج	
Close	د	
8		م
عند بداية فتح بيئة Pycharm يتم ما يلي		العبارة
غلق مشروع	أ	البدائل
مشاركة البيانات	ب	
تصدير مشروع	ج	
انشاء مشروع جديد	د	
9		م
يتم انشاء مشروع في Pycharm من خلال امر		العبارة
File	أ	البدائل
New	ب	
Edit	ج	
Run	د	

10		م
عند بداية فتح بيئة Pycharm يتم ما يلي		العبرة
انشاء مشروع وحفظه	أ	البدائل
ضبط مشاركة الاكواد	ب	
غلق جميع الملفات	ج	
ضبط نوع الخط	د	
11		م
يتم حفظ تعديلات مشروع في Pycharm من خلال		العبرة
CTRL + C	أ	البدائل
CTRL + V	ب	
CTRL + S	ج	
CTRL + P	د	
12		م
يتم تجربة كود برمجي في Pycharm من خلال		العبرة
Shift + F5	أ	البدائل
Shift + F10	ب	
CTRL + S	ج	
CTRL + C	د	

13		م
ينتهي سطر الكود في Pycharm بعد أن يتم		العبارة
انتهاء عدد السطور	أ	البدائل
كتابة فاصلة عادية	ب	
كتابة فاصلة منقوطة	ج	
الضغط على Enter	د	
14		م
يتم تبديل القالب التصميمي في Pycharm من خلال أمر		العبارة
Desktop	أ	البدائل
Interface	ب	
Theme	ج	
Graphics	د	
15		م
أول خطوة في كتابة كود في بيئة Pycharm هي		العبارة
مسح الكود التجريبي	أ	البدائل
تبديل القالب التصميمي	ب	
انشاء مجلد جديد	ج	
فتح نافذة مشروع مخصص	د	

مفتاح الإجابة

م	العبارة	الإجابة الصحيحة
1	يقوم بتحويل كود بايثون إلى لغة الآلة في نظام تشغيل ويندوز	مترجم Python
2	يتم تحميل لغة بايثون من	موقع Python
3	يتم التأكد من نجاح تسطيب بايثون عن طريق	Command Prompt
4	ما هو الإصدار المجاني الخاص ببيئة Pycharm	Community
5	من متطلبات تشغيل البرنامج للمرة الأولى، خاص بمشاركة البيانات	Data Sharing
6	من مكونات واجهة Pycharm ويظهر فيه اسم الملف الحالي	Title Bar
7	يمكن اغلاق مشروع في Pycharm من خلال قائمة	File
8	عند بداية فتح بيئة Pycharm يتم ما يلي	انشاء مشروع جديد
9	يتم انشاء مشروع في Pycharm من خلال امر	New
10	عند بداية فتح بيئة Pycharm يتم ما يلي	انشاء مشروع وحفظه
11	يتم حفظ تعديلات مشروع في Pycharm من خلال	CTRL + S
12	يتم تجربة كود برمجي في Pycharm من خلال	Shift + F10
13	ينتهي سطر الكود في Pycharm بعد أن يتم	الضغط على Enter
14	يتم تبديل القالب التصميمي في Pycharm من خلال أمر	Theme
15	أول خطوة في كتابة كود في بيئة Pycharm هي	مسح الكود التجريبي

الفصل الثاني

بيئة تطوير بايثون في نظام اندرويد

متطلبات العمل بلغة بايثون في اندرويد

لغة وبيئة تطوير بايثون في اندرويد

مجلد حفظ المشاريع

انشاء مشروع جديد

حفظ التعديلات الدورية

اغلق مشروع

مشاهدة المشاريع الاخوة

القالب التصميمي

الاعمال التلقائي

اظهار الأخطاء والتعديلات

كتابة كود برمجي

التراجع عن كتابة كود

تجربة تنفيذ كود برمجي

كتابة أول برنامج بلغة بايثون وتجربته

مقدمة الفصل

أولاً - الهدف العام:

إكساب طلاب تكنولوجيا التعليم الرقمي المفاهيم والمفاهيم والمهارات الأساسية الخاصة بالعمل ببيئة بتطوير لغة بايثون الخاصة بنظام التشغيل اندرويد على أجهزة الموبايل.

ثانياً - الأهداف التعليمية:

بعد الانتهاء من هذا الفصل /الدرس، يتوقع من الطالب أن يكون قادراً على أن:

1. يشرح متطلبات العمل بلغة بايثون في نظام تشغيل أندرويد

2. يقوم بتثبيت بيئة تطوير بايثون 3 Pydroid

3. يحدد مجلد حفظ المشاريع في بيئة Pydroid 3

4. ينشئ مشروع جديد

5. يحفظ تعديلات مشروع

6. يغلق ملف مشروع

7. يقوم بمشاهدة المشاريع الأخيرة

8. يقوم بتبديل قالب التصميمي

9. يفعل خاصية الاكمال التلقائي

10. يفعل خاصية اظهار الأخطاء والتعديلات

11. يقوم بكتابة كود برمجي في بيئة Pydroid 3

12. ينفذ أمر التراجع عن كتابة كود

13. يجرب نتيجة تنفيذ كود برمجي

14. يكتب أول برنامج بلغة بايثون ويجريه

بيئة تطوير بايثون في نظام اندرويد

متطلبات العمل بلغة بايثون في اندرويد

لتنفيذ المشروعات البرمجية بلغة البايثون في أجهزة الموبايل (أسوة بنظام التشغيل ويندوز) يتطلب توفير المكونات التالية في نظام التشغيل:

1- مترجم اللغة / مشغل بايثون (Python Interpreter)

يقوم بتحويل الكود المصدري (Source Code) الذي كتبه المبرمج بإستخدام أي محرر نصوص (Word، Notepad) والمحفوظ بالإمتداد .py ، إلى لغة الآلة (Byte Code) .

2- بيئة تطوير

برامج توفر للمبرمجين إمكانية كتابة الكود المصدري ثم حفظه في ملف بالإمتداد .py.

لغة وبيئة تطوير بايثون في اندرويد

يتوفر في منصات تحميل التطبيقات مثل Google Store العديد من التطبيقات التي تحتوي بداخلها على كلا من المترجم وبيئة التطوير معاً ، ومن أشهرها تطبيق Pydroid ، الذي بمجرد تحميله وتسطيبه على جهاز موبايل اندرويد يتم تحميل المترجم وبيئة التطوير ويجعل الموبايل جاهزاً للحمل على اللغة مباشرة ، يتم تسطيب تطبيق Pydroid من خلال الخطوات التالية:

1- تحميل التطبيق من بلاي ستور



2- انتظار الانتهاء من تسطيب التطبيق

مجلد حفظ المشاريع

على خلاف بيئة PyCharm التي تحدد بشكل تلقائي مكان تخزين المشاريع، يطلب تطبيق Pydroid تحديد المكان من قبل المستخدم، وذلك وفقاً للخطوات التالية:

الخطوة الأولى	الخطوة الثانية	الخطوة الثالثة
		
الضغط على أداة المجلد	اختيار أمر Save as	تحديد المجلد، ثم الضغط على امر select folder

انشاء مشروع جديد

يتم انشاء مشروع جديد (وتحديد الاسم في ذات الوقت) من خلال تنفيذ التالي:

الخطوة الأولى	الخطوة الثانية	الخطوة الثالثة
		
الضغط على أداة المجلد	اختيار أمر New	كتابة اسم المشروع ، ثم الضغط على امر save

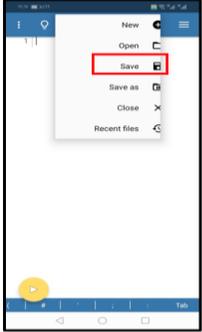
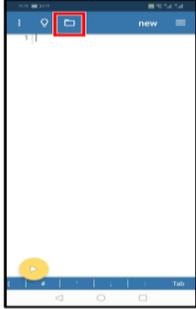
حفظ التعديلات الدورية

يتم حفظ التعديلات الدورية على مشروع من خلال تنفيذ الخطوات التالية:

الخطوة الأولى	الخطوة الثانية
	
الضغط على أداة المجلد	اختيار أمر save

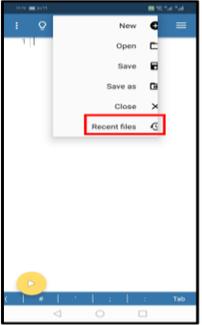
اغلاق مشروع

يتم اغلاق مشروع من خلال تنفيذ الخطوات التالية:

الخطوة الثانية	الخطوة الأولى
 <p data-bbox="340 877 552 915">اختيار أمر Close</p>	 <p data-bbox="793 877 1053 915">الضغط على أداة المجلد</p>

مشاهدة المشاريع الأخيرة

يتم مشاهدة اخر مشاريع تم حفظها من خلال تنفيذ الخطوات التالية:

الخطوة الثانية	الخطوة الأولى
 <p data-bbox="299 1595 598 1634">اختيار أمر Recent files</p>	 <p data-bbox="793 1595 1053 1634">الضغط على أداة المجلد</p>

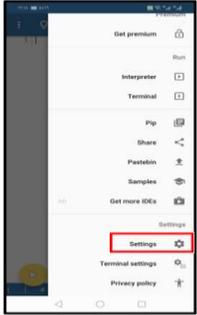
القالب التصميمي

يتم ضبط القالب التصميمي Theme لواجهة التطبيق من خلال التالي:

<p>الخطوة الثانية</p>  <p>اختيار أمر settings</p>	<p>الخطوة الأولى</p>  <p>الضغط على أداة الاعدادات</p>
<p>الخطوة الرابعة</p>  <p>اختيار أمر Editor theme</p>	<p>الخطوة الثالثة</p>  <p>اختيار أمر appearance</p>

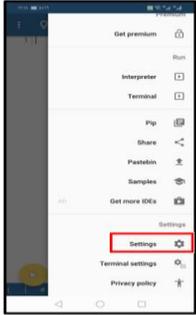
الاكمال التلقائي

يتم تنشيط وظيفة اكمال الكود بشكل تلقائي دون الحاجة لكتابته كاملة من خلال الخطوات التالية:

<p style="text-align: center;">الخطوة الثانية</p>  <p style="text-align: center;">اختيار أمر settings</p>	<p style="text-align: center;">الخطوة الأولى</p>  <p style="text-align: center;">الضغط على أداة الاعدادات</p>
<p style="text-align: center;">الخطوة الرابعة</p>  <p style="text-align: center;">تنشيط الأمر Enable autocomplete</p>	<p style="text-align: center;">الخطوة الثالثة</p>  <p style="text-align: center;">اختيار أمر System</p>

اظهار الأخطاء والتعديلات

يتم تنشيط وظيفة اظهار الأخطاء والتعديلات حتى يتم بشكل تلقائي تصحيح الكود المكتوب من خلال الخطوات التالية:

<p>الخطوة الثانية</p>  <p>اختيار أمر settings</p>	<p>الخطوة الأولى</p>  <p>الضغط على أداة الاعدادات</p>
<p>الخطوة الرابعة</p>  <p>تنشيط الأمرين show warning و Show error</p>	<p>الخطوة الثالثة</p>  <p>اختيار أمر Code analysis</p>

كتابة كود برمجي

يتم كتابة الكود البرمجي كما يلي:

الخطوة الثانية	الخطوة الأولى
 <p>كتابة الكود مباشرة باستخدام لوحة مفاتيح الكتابة بالموبايل نفسه</p>	 <p>الضغط على مساحة كتابة الكود</p>

التراجع عن كتابة كود

يتم التراجع عن كتابة كود برمجي ، سواء خطوة للأمام أو للخلف كما يلي:

الخطوة الثانية	الخطوة الأولى
 <p>اختيار الأمر Undo للتراجع خطوة للخلف ، اختيار الأمر Redo للتراجع خطوة للأمام</p>	 <p>الضغط على الأداة (:)</p>

تجربة تنفيذ كود برمجي

يتم تجربة كود برمجي (Run) كما يلي:



كتابة أول برنامج بلغة بايثون وتجربته

يتم كتابة برنامج بايثون وفق الخطوات التالية:

1 - انشاء مشروع جديد (New)

2 - كتابة الكود التالي :

```
print ( " Hello World " )
```

3 - تجربة الكود بالأمر ()

نشاط تعليمي على بيئة تطوير بايثون في نظام تشغيل أندرويد على أجهزة الموبايل

البيان	الوصف
رقم النشاط	1
عنوان النشاط	المقارنة بين بيئة PyCharm وبيئة تطوير ميكروسوفت فيجوال ستوديو
وصف النشاط	يتم فتح باب المناقشة في الفرق بين بيئة PyCharm وبيئة تطوير ميكروسوفت فيجوال ستوديو
توقيت التنفيذ	خلال المحاضرة
طريقة التنفيذ	تعليم تعاوني - مجموعات صغيرة

تكاليفات برمجية

قم بتنفيذ التكليف التالي، ثم تواصل مع أستاذ المادة لتسليمه ومناقشته في المحاضرات التالية:

التكليف:

اكتب تقرير موجز عن عن بيئات أخرى لبرمجة بايثون في ويندوز بخلاف بيئة PyCharm.

مصادر إثرائية من بنك المعرفة المصري



Egyptian Knowledge Bank
بنك المعرفة المصري

من خلال المصادر الالكترونية التالية والموجودة ببنك المعرفة المصري والمكتبة الرقمية؛ يمكنك الاستزادة حول المحتوى العلمي للفصل:

م	عنوان المصدر	الرابط
1	كتاب في نظام تشغيل اندرويد	https://0810gxsjp-1106-y-https-search-mandumah-com.mplbci.ekb.eg/Record/661018
2	دراسة بحثية عن تطوير تطبيق الموبايل لطلاب المدارس	https://0810gxsjr-1106-y-https-search-mandumah-com.mplbci.ekb.eg/Record/915808
3	دراسة بحثية عن تطبيقات الاندرويد في المستشفيات التعليمية	https://0810gxsjr-1106-y-https-search-mandumah-com.mplbci.ekb.eg/Record/831990

قاموس المصطلحات الواردة بالفصل

م	المصطلح بالعربية	المصطلح بالإنجليزية	الوصف
1	بايدرويد	Pydroid	تطبيق موبايل مجاني ، يضم بداخله مترجم بايثون وبيئة تطوير بايثون ، تمكن المبرمجين من برمجة مشاريع بايثون من خلال أجهزة الموبايل التي تعمل بنظام تشغيل اندرويد
2	القالب التصميمي	Theme	التصميم الرسومي لواجهة تشغيل تطبيق وما تحتويه من ألوان للخلفيات والكتابات والعناصر الرسومية الموجودة به
3	الاكمال التلقائي	Auto Complete	خاصية تقوم بإكمال الكلمة المكتوبة بمجرد كتابة أول أحرف منها ، وهذا يوفر وقت المبرمج ويقلل من أخطاء كتابة الكود البرمجي
4	التصحيح التلقائي	Auto Correct	خاصية تقوم بتصحيح الكلمة المكتوبة إذا كانت مكتوبة بشكل خاطئ ، وهذا يقلل من أخطاء كتابة الكود البرمجي بشكل كبير

اختبر نفسك

اختر الإجابة الصحيحة من بين البدائل الموجودة:

1		م
يوفر إمكانية كتابة لكود وتحويله إلى امتداد PY في نظام اندرويد		العبارة
بيئة تطوير Pydroid	أ	البدائل
مترجم Python	ب	
بيئة تطوير Pycharm	ج	
نظام Windows	د	
2		م
يقوم بتسطيب مشغل لغة بايثون في أجهزة الموبايل بنظام اندرويد		العبارة
Play Store	أ	البدائل
بيئة تطوير Pydroid	ب	
نظام اندرويد	ج	
بيئة تطوير Pycharm	د	
3		م
يتم تحديد مجلد حفظ المشاريع في Pydroid من خلال		العبارة
Exit	أ	البدائل
New	ب	
Close	ج	
Save As	د	

4		م
يتم انشاء مشروع جديد في Pydroid من خلال امر		العبارة
Save	أ	البدائل
Undo	ب	
New	ج	
Open	د	
5		م
يتم حفظ التعديلات الدورية مشروع في Pydroid من خلال قائمة ذات شكل		العبارة
مجلد	أ	البدائل
اعدادات	ب	
أداة بحث	ج	
أداة تشغيل	د	
6		م
يمكن اغلاق مشروع في Pydroid من خلال ايقونة ذات شكل		العبارة
مجلد	أ	البدائل
علامة الغلق	ب	
مثث	ج	
دائرة	د	

7		م
يتم مشاهدة المشاريع الأخيرة في Pydroid من خلال		العبارة
Open Files	أ	البدائل
Recent Files	ب	
Show Files	ج	
Last Files	د	
8		م
يمكن التبديل القالب التصميمي في Pydroid من خلال أمر		العبارة
Interface	أ	البدائل
Colors	ب	
Bit 32	ج	
Appearance	د	
9		م
يتم الاكمال التلقائي للكود في Pydroid من خلال		العبارة
Share	أ	البدائل
Pip	ب	
Settings	ج	
Interpreter	د	

10		م
يتم اظهار اخطاء الكود في Pydroid من خلال		العبارة
Code Analysis	أ	البدائل
System	ب	
Editor	ج	
Appearance	د	
11		م
يتم كتابة الكود في Pydroid في		العبارة
ملف مصدر	أ	البدائل
ملف نصي	ب	
مساحة العمل	ج	
مساحة برمجية	د	
12		م
يتم التراجع عن أمر (خطوة للأمام) في Pydroid من خلال		العبارة
Undo	أ	البدائل
File	ب	
Search	ج	
Redo	د	

13		م
يمكن تجربة كود برمجي في Pydroid من خلال		العبارة
Run	أ	البدائل
Share	ب	
Terminal	ج	
Analysis	د	
14		م
آخر خطوة في كتابة كود في بيئة Pydroid هي		العبارة
تجربة الكود	أ	البدائل
مسح الكود التجريبي	ب	
كتابة الكود	ج	
انشاء مشروع	د	

مفتاح الإجابة

م	العبارة	الإجابة الصحيحة
1	يوفر إمكانية كتابة لكود وتحويله إلى امتداد PY في اندرويد	بيئة تطوير Pydroid
2	يقوم بتسطيب مشغل لغة بايثون في أجهزة الموبايل التي تعمل بنظام اندرويد	بيئة تطوير Pydroid
3	يتم تحديد مجلد حفظ المشاريع في Pydroid من خلال	Save As
4	يتم انشاء مشروع جديد في Pydroid من خلال امر	New
5	يتم حفظ التعديلات الدورية مشروع في Pydroid من خلال قائمة ذات شكل	مجلد
6	يمكن اغلاق مشروع في Pydroid من خلال ايقونة ذات شكل	مجلد
7	يتم مشاهدة المشاريع الأخيرة في Pydroid من خلال	Recent Files
8	يمكن التبديل القالب التصميمي في Pydroid من خلال أمر	Appearance
9	يتم الاكمال التلقائي للكود في Pydroid من خلال	Settings
10	يتم اظهار اخطاء الكود في Pydroid من خلال	Code Analysis
11	يتم كتابة الكود في Pydroid في	مساحة العمل
12	يتم التراجع عن أمر (خطوة للأمام) في Pydroid من خلال	Redo
13	يمكن تجربة كود برمجي في Pydroid من خلال	Run
14	آخر خطوة في كتابة كود في بيئة Pydroid هي	تجربة الكود

الفصل الثالث

العوامل البرمجية وخرائط التدفق

العوامل البرمجية

مفهوم العمليات الحسابية في البرمجة

مراحل تنفيذ العمليات الحسابية في البرمجة

العوامل الحسابية في البرمجة

عوامل الإسناد في البرمجة

خرائط التدفق

مفهوم خرائط التدفق

أهمية رسم خرائط التدفق

رموز خرائط التدفق

خطوات رسم خريطة تدفق لعملية برمجية

مقدمة الفصل

أولاً - الهدف العام:

إكساب طلاب تكنولوجيا التعليم الرقمي المفاهيم والمهارات الأساسية الخاصة بالعمليات على العوامل الحسابية وعوامل الإسناد في البرمجة.

ثانياً - الأهداف التعليمية:

بعد الانتهاء من هذا الفصل /الدرس، يتوقع من الطالب أن يكون قادراً على أن:

1. يذكر مفهوم العمليات الحسابية في البرمجة
2. يعدد مراحل تنفيذ العمليات الحسابية في البرمجة
3. يميز بين العوامل الحسابية البرمجية
4. يفرق بين عوامل الإسناد البرمجية
5. يذكر مفهوم خرائط التدفق
6. يوضح أهمية رسم خرائط التدفق للمشاريع البرمجية
7. يفرق بين رموز خرائط التدفق
8. يرسم خريطة تدفق لعملية برمجية

العوامل البرمجية (الحسابية ، الاسناد)

مفهوم العمليات الحسابية في البرمجة

هي أكواد برمجية تستخدم في تنفيذ عملية حسابية يطلبها المستخدم ، وهي أحد المراحل الثلاثة للبرنامج (إدخال - عمليات - إخراج)

مراحل تنفيذ العمليات الحسابية في البرمجة

كما تم الإشارة سابقاً ، تتكون مراحل تنفيذ العمليات في أي برنامج من ثلاث مراحل رئيسية هي:

أولاً - مرحلة الإدخال:

- يقوم فيها المستخدم بإدخال القيم المطلوبة منه والتي يريد نتيجتها.
- مثل إدخال طول نصف قطر دائرة لكي يعرف مساحتها
- كما درسنا من قبل، يستخدم في الإدخال الكود البرمجي input ، والذي يطلب من المستخدم إدخال قيمة معينة وتسكينها في متغير

ثانياً - مرحلة العمليات:

- يقوم فيها البرنامج بعمل الحسابات الرياضية المطلوبة منه ، وذلك بناء على أكواد محددة كتبها المبرمج لتخزين المعادلة داخل البرنامج.
- مثل حساب مساحة دائرة بالمعادلة (ط نق 2)
- يستخدم فيها العوامل الحسابية Arithmetic Operators ، وعوامل

الإسناد Assignment Operators

ثالثاً - مرحلة الإخراج :

- يقوم فيها البرنامج بإخراج رسالة للمستخدم يوضح فيها قيمة العملية الحسابية الناتجة.
- مثل إخراج مساحة الدائرة بشكل فعلي بناء على ما أدخله المستخدم
- كما درسنا من قبل، يستخدم في الإخراج الكود البرمجي print، والذي يظهر نتيجة التنفيذ للمستخدم

العوامل الحسابية البرمجية

الرموز التي تكون أشكال المعادلات الحسابية في المشروع البرمجي ، وهي:

- $a + b$: عملية حسابية تقوم **بجمع** a و b معاً
- $a - b$: عملية حسابية تقوم **بطرح** a ناقص b
- a / b : عملية حسابية تقوم **بقسمة** a على b
- $a // b$: عملية حسابية تقوم **بقسمة** a على b **بدون باقي** قسمة
- $a * b$: عملية حسابية تقوم **بضرب** a و b معاً

عوامل الإسناد البرمجية

الرموز والتي تقوم بإختصار إسناد قيم للمتغيرات ، وشكلها كما يلي:

- $a + = b$: تعني **زيادة** قيمة a **بمقدار** b ($a = a + b$)
- $a - = b$: تعني **تقليل** قيمة a **بمقدار** b ($a = a - b$)
- $a * = b$: تعني **تحويل** قيمة a إلى $a * b$ ($a = a * b$)
- $a / = b$: تعني **تحويل** قيمة a إلى a / b ($a = a / b$)

خرائط التدفق Flow Charts

مفهوم خرائط التدفق

هي تمثيل تخطيطي يعتمد على الرسم بأشكال قياسية لتوضيح ترتيب العمليات اللازمة لحل مسألة أو مشكلة محددة (تستخدم في مجالات أخرى غير البرمجة)

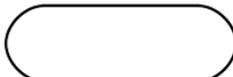
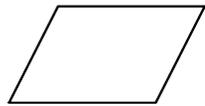
أهمية رسم خرائط التدفق

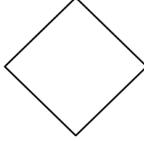
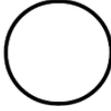
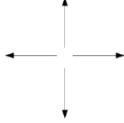
لرسم خرائط التدفق مزايا عديدة، منها التالي:

- تيسر قراءة وفهم المشكلة وتوضح للمبرمج ما يجب عمله
- مفيدة في شرح البرنامج للآخرين
- تساعد خريطة التدفق في توثيق أفضل للبرنامج وخصوصا إذا كان البرنامج معقدا

رموز خرائط التدفق

يوضح الجدول التالي أهم رموز خرائط التدفق والتي قد تستخدم في معظم المشاريع البرمجية:

الوظيفة	الرمز
بداية المشروع ، نهاية المشروع	
اكواد الادخال - اكواد الإخراج	

الوظيفة	الرمز
اكواد العمليات الحسابية	
اكواد تفرع البرنامج إلى مسارات مختلفة بناء على نتيجة عملية منطقية	
اذا كان البرنامج طويلا ولا يكفي لكتابته صفحة كاملة ، يوضع هذا الرمز ليربط بين البرنامج الموزع على صفحات مختلفة	
أسهم توضح اتجاه سير العمليات	

خطوات رسم خريطة تدفق لمشروع برمجي

يتكون أي مشروع برمجي (في أبسط صورة له) من المراحل الخمسة التالية:

- 1 - بداية المشروع
- 2 - مرحلة الادخال
- 3 - مرحلة العمليات
- 4 - مرحلة الإخراج
- 5 - نهاية المشروع

فإذا كان المشروع البرمجي يهدف إلى إيجاد جمع عددين يدخلهم المستخدم ،
ستكون مراحل تنفيذ المشروع على النحو التالي :

1 - بداية المشروع

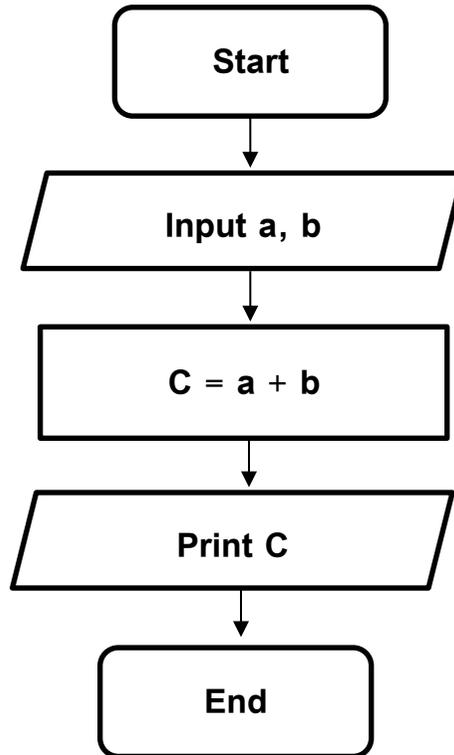
2 - مرحلة الإدخال : ادخال قيمة العددين من قبل المستخدم

3 - مرحلة العمليات : إيجاد حاصل جمع العددين

4 - مرحلة الإخراج : اخراج قيمة حاصل جمع العددين

5 - نهاية المشروع

يوضح الشكل التالي مراحل سير البرنامج على شكل خريطة تدفق



أنشطة تعليمية على العوامل الحسابية وعوامل الاسناد البرمجي

رقم النشاط	عنوان النشاط	وصف النشاط	توقيت التنفيذ	طريقة التنفيذ
1	أكتب الكود البرمجي	اكتب كود العملية الحسابية التي تهدف لإيجاد قيمة الرصيد البنكي بالجنيه بعد تحويله من الدولار الأمريكي ، علما بأن سعر صرف الدولار = 16 جنية	خلال المحاضرة	تعليم تعاوني مجموعات صغيرة
2	إستنتج نتيجة تنفيذ كود برمجي	استنتج نتيجة تنفيذ الكود التالي $x = y * z$		
3	اكتب برنامج بايثون	اكتب برنامج بايثون يطلب فيه من المستخدم حساب مساحة دائرة ، بناء على طول نصف قطر الدائرة المدخل من قبل المستخدم ، علما بأن مساحة		

رقم النشاط	عنوان النشاط	وصف النشاط	توقيت التنفيذ	طريقة التنفيذ
		الدائرة = ط نق 2		
4	إستخرج الخطأ البرمجي	<p>استخرج الخطأ البرمجي في الكود التالي</p> <pre>dollar = float(input("Enter balance :")) pound = dollar * 16 print ("Egyptian Balance=")</pre>		

تكاليف برمجية

قم بتنفيذ التكاليف التالية، ثم تواصل مع أستاذ المادة لتسليمها ومناقشتها في المحاضرات التالية:

م	التكليف
1	اكتب كود العملية الحسابية التي تهدف لإيجاد الوزن بالجرام بعد تحويله من الكيلو جرام ، علماً بأن الكيلو جرام = 1000 جرام
2	استنتج نتيجة تنفيذ الكود التالي rectangle_area = length * width
3	اكتب برنامج بايثون يطلب فيه من المستخدم حساب محيط دائرة ، بناء على طول نصف قطر الدائرة المدخل من قبل المستخدم ، علماً بأن محيط الدائرة = 2 ط نق
4	استخرج الخطأ البرمجي في الكود التالي dollar = input ("Enter balance :") pound = dollar * 16 print ("Egyptian Balance= " , pound)

مصادر إثنائية من بنك المعرفة المصري



Egyptian Knowledge Bank
بنك المعرفة المصري

من خلال المصادر الالكترونية التالية والموجودة ببنك المعرفة المصري والمكتبة الرقمية؛ يمكنك الاستزادة حول المحتوى العلمي للفصل:

م	عنوان المصدر	الرابط
1	كتاب عن مفاهيم البرمجة بين القديم والحديث	https://0811nxsk9-1106-y-https-ekbedu-ethraadl-com.mplbci.ekb.eg/Record/62469
2	دراسة عن استخدام البرمجية في العلاج الطبي	https://0810gxsjr-1106-y-https-search-mandumah-com.mplbci.ekb.eg/Record/834817

قاموس المصطلحات الواردة بالفصل

م	المصطلح بالعربية	المصطلح بالإنجليزية	الوصف
1	العوامل الحسابية	Arithmetic Operators	هي الرموز التي تكون أشكال المعادلات الحسابية ، أشهرها علامات (* ÷ / - +)
2	العوامل الاسنادية	Assigning Operators	هي الرموز التي تقوم باختصار اسناد قيم للمتغيرات ، أشهرها علامات (=+ ، =* ، =/ ، =-)

اختبر نفسك

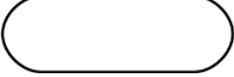
اختر الإجابة الصحيحة من بين البدائل الموجودة:

1		م
أي من التالي لا ينطبق على مفهوم العمليات الحسابية البرمجية		العبارة
اختبار الشرط OR	أ	البدائل
إيجاد ناتج جمع	ب	
تعتمد على متغيرات رقمية	ج	
من مراحل المشروع البرمجي	د	
2		م
من مراحل تنفيذ العمليات الحسابية في البرمجة يتم فيها تسكين قيمة في متغير		العبارة
العمليات	أ	البدائل
الإخراج	ب	
الادخال	ج	
التنفيذ	د	

3		م
من العوامل الحسابية البرمجية ويستخدم في عملية القسمة بدون باقي		العبارة
*	أ	البدائل
&	ب	
Not	ج	
//	د	
4		م
من عوامل الاسناد البرمجية وتعني زيادة قيمة المتغير بمقدار		العبارة
= -	أ	البدائل
= +	ب	
= *	ج	
= /	د	
5		م
أي من التالي من خصائص خرائط التدفق		العبارة
تمثيل تخطيطي لمشروع برمجي	أ	البدائل
إضافة زر رسومي	ب	
إضافة واجهة GUI	ج	
ضبط واجهة المستخدم	د	

6		م
تبرز أهمية خرائط التدفق في		العبارة
ضبط قالب البرنامج	أ	البدائل
تبسيط المشكلة البرمجية	ب	
إضافة كود النافذة	ج	
ضبط الدوال البرمجية	د	
7		م
		العبارة
كود اخراج	أ	البدائل
كود عملية حسابية	ب	
بداية مشروع	ج	
ربط البرنامج بين الصفحات	د	
8		م
الخطوة الأسبق في رسم خريطة تدفق هي		العبارة
العمليات	أ	البدائل
الإخراج	ب	
نهاية المشروع	ج	
الادخال	د	

مفتاح الإجابة

م	العبرة	الإجابة الصحيحة
1	أي من التالي لا ينطبق على مفهوم العمليات الحسابية البرمجية	اختبار الشرط OR
2	من مراحل تنفيذ العمليات الحسابية في البرمجة يتم فيها تسكين قيمة في متغير	الادخال
3	من العوامل الحسابية البرمجية ويستخدم في عملية القسمة بدون باقي	//
4	من عوامل الاسناد البرمجية وتعني زيادة قيمة المتغير بمقدار	+ =
5	أي من التالي من خصائص خرائط التدفق	تمثيل تخطيطي لمشروع برمجي
6	تبرز أهمية خرائط التدفق في	تبسيط المشكلة البرمجية
7		بداية مشروع 
8	الخطوة الأسبق في رسم خريطة تدفق هي	الادخال

الفصل الرابع

الشروط والتفرع البرمجي Branching

مفاهيم في الشروط والتفرع البرمجي

مفهوم الشروط والتفرع المنطقي في البرمجة

مراحل تنفيذ العمليات المنطقية في البرمجة

عوامل المقارنات البرمجة

العوامل المنطقية البرمجية

التفرع الشرطي البسيط

مفهوم التفرع الشرطي البسيط

كود التفرع الشرطي البسيط

التفرع الشرطي الثنائي

مفهوم التفرع الشرطي الثنائي

كود التفرع الشرطي الثنائي

التفرع الشرطي المتعدد

مفهوم التفرع الشرطي المتعدد

كود التفرع الشرطي المتعدد

مقدمة الفصل

أولاً - الهدف العام:

إكساب طلاب تكنولوجيا التعليم الرقمي المفاهيم والمهارات الأساسية الخاصة بالعمليات على الشرط والتفرع البرمجي.

ثانياً - الأهداف التعليمية:

بعد الانتهاء من هذا الفصل /الدرس، يتوقع من الطالب أن يكون قادراً على أن:

1. يذكر مفهوم الشرط والتفرع المنطقي في البرمجة

2. يوضح مراحل تنفيذ العمليات المنطقية في البرمجة

3. يبين عوامل المقارنات البرمجية

4. يميز بين العوامل المنطقية البرمجية

5. يذكر مفهوم التفرع الشرطي البسيط

6. يكتب كود التفرع الشرطي البسيط

7. يذكر مفهوم التفرع الشرطي الثنائي

8. يكتب كود التفرع الشرطي الثنائي

9. يذكر مفهوم التفرع الشرطي المتعدد

10. يكتب كود التفرع الشرطي المتعدد

مفاهيم في الشرط والتفرع البرمجي

مفهوم الشرط والتفرع المنطقي في البرمجة

- هو قيام البرنامج بإختبار شرط معين ، وبناء عليه يتفرع البرنامج إلى مسارات متعددة ، بحيث يتم الدخول في مسار محدد بناء على قيمة هذا الشرط ، وبالتالي فإن المبرمج يقوم بوضع الشرط ووضع المسارات المحتملة ، وعندما يقوم البرنامج أو المستخدم بتحديد قيمة للشرط فإن البرنامج يسري في إتجاه مسار هذه القيمة تحديدا
- على سبيل المثال يقوم المستخدم بإدخال كلمة المرور ، فيقوم البرنامج بإختبار الكلمة فإذا كانت صحيحة سيقوم بالسماح بالدخول ، وإذا لم تكن صحيحة فلن يسمح للمستخدم بالدخول

مراحل تنفيذ العمليات المنطقية في البرمجة

العملية المنطقية مثلها مثل العملية الحسابية يتم تنفيذها من خلال ثلاث مراحل رئيسية هي:

أولاً - مرحلة الإدخال:

- يقوم فيها المستخدم بإدخال القيم المطلوبة منه والتي بناء عليها يتم السير في مسار محدد.
- مثل إدخال كلمة المرور بهدف دخول البرنامج

- كما درسنا من قبل ، يستخدم في الإدخال الكود البرمجي input ، والذي يطلب من المستخدم إدخال قيمة معينة وتسكينها في متغير

ثانياً – مرحلة العمليات:

- يقوم فيها البرنامج بعمل الحسابات المنطقية المطلوبة منه ، وذلك بناء على أكواد محددة كتبها المبرمج ، تحدد المسارات والإحتمالات.
- مثل مقارنة كلمة المرور بكلمة المرور الصحيحة والمخزنة في البرنامج
- يستخدم فيها الجملة if ، وعوامل المقارنات Comparison Operators ، الشروط المنطقية Logical Operators

ثالثاً – مرحلة الإخراج :

- يقوم فيها بتحويل مسار البرنامج للإحتمال الذي يتناسب مع نتيجة العملية المنطقية ، وبالتالي تكون المخرجات وفق هذا المسار.
- مثل السماح بالدخول في حالة كلمة مرور صحيحة ، أو عدم السماح بالمرور إذا كانت كلمة المرور خاطئة
- يتم تحويل المسارات باستخدام (if) و (if else) و (else) بالإضافة لجملة إخراج البيانات (print)

عوامل المقارنات البرمجية

هي الرموز التي تستخدم في العملية المنطقية لإختبار الشرط في المشاريع البرمجية ، فإذا كان الشرط متحقق تكون القيمة المرجعة True ، أو لا تتحقق فتكون False وهي:

- $a == b$: تختبر ما إذا كانت قيمة **a تساوي** **b** أم لا
- $a != b$: تختبر ما إذا كانت قيمة **a لا تساوي** **b** أم لا
- $a > b$: تختبر ما إذا كانت قيمة **a أكبر من** **b** أم لا
- $a < b$: تختبر ما إذا كانت قيمة **a أصغر من** **b** أم لا
- $a >= b$: تختبر ما إذا كانت قيمة **a أكبر من أو تساوي** **b**
- $a <= b$: تختبر ما إذا كانت قيمة **a أقل من أو تساوي** **b**

العوامل المنطقية البرمجية

هي اكواد برمجية مخصصة لاختبار نتيجة عمليتين (منطقيتين نتيجة عوامل مقارنة) أو (حسابيتين نتيجة لعوامل حسابية) ، تتبع علم رياضيات الكمبيوتر وجبر المحددات المنطقية ، من أشهر العوامل المنطقية ما يلي:

أولاً – العامل المنطقي AND :

يستخدم لتنفيذ كود معين إذا تحققت جميع الشروط الموضوعه

- قاعدة كتابة الكود : (condition 1 **and** condition 2)
- نتيجة تنفيذ الكود :

- إذا تحقق condition 1 أي كانت قيمته true ، وأيضاً تحقق condition 2 أي كانت قيمته true ، تصبح نتيجة الكود true
- إذا لم يتحقق احداها (سواء الأول أو الثاني) أي كانت قيمته false ، تصبح نتيجة الكود false
- بالتأكيد إذا لم يتحقق كلاهما ، تصبح نتيجة الكود false
- مثال : **if a == 5 and b == 10**
- ستكون نتيجة العملية True إذا كانت a = 5 ، و b = 10
- فيما عدا ذلك ستكون النتيجة False

ثانياً – العامل المنطقي OR :

يستخدم لتنفيذ كود معين إذا تحققت شرط واحد على الأقل من الشروط
الموضوعة

- قاعدة كتابة الكود : (condition 1 or condition 2)
- نتيجة تنفيذ الكود :
- إذا تحقق condition 1 أي كانت قيمته true ، وأيضاً تحقق condition 2 أي كانت قيمته true ، النتيجة true
- إذا يتحقق احداها على الأقل (سواء الأول أو الثاني) أي كانت قيمته True ، تصبح نتيجة الكود True ، حتى لو كان لم يتحقق التالي وكانت نتيجته False

- بالتأكيد إذا لمن يتحقق كلاهما، تصبح نتيجة الكود false
- مثال : **if a == 5 or b == 10**
- ستكون نتيجة العملية True إذا كانت a = 5 أو قيمة b = 10 ،
واحدة منهم تكفي لجعل النتيجة True
- منطقياً عدم تحقق الاثنين يعني ان النتيجة النهائية false

ثالثاً – العامل المنطقي Not :

يستخدم لنفي نتيجة شرط موضوع

- قاعدة كتابة الكود : (not condition)
- نتيجة تنفيذ الكود :
- اذا تحقق الـ condition أي كانت قيمته true ستكون نتيجة الكود false
- اذا لم تحقق الـ condition أي كانت قيمته false ستكون نتيجة الكود true
- بالتأكيد إذا لمن يتحقق كلاهما ، تصبح نتيجة الكود false
- مثال : **print (not b)**
- ستكون نتيجة العملية True إذا كانت a = false
- ستكون نتيجة العملية false إذا كانت a = True

التفرع الشرطي البسيط

مفهوم التفرع الشرطي البسيط

هي عملية برمجية يتم فيها اختبار الشرط ، إذا تحقق الشرط سيتم تنفيذ الكود ، وإذا لم يتحقق الشرط كأن لم يكن وسيتم الانتقال إلى السطر التالي من الكود

كود التفرع الشرطي البسيط

والذي يتم فيه اختبار الشرط البسيط (تجربة شرط واحد)

قاعدة كتابة كود التفرع الشرطي البسيط	
<code>variable = value</code>	المتغير الذي يتم تجربته منطقياً
<code>If condition :</code>	الجملة المنطقية التي تجرب الشرط
<code>Code</code>	إذا تحقق الشرط المنطقي يتم تنفيذ هذا الكود
مثال على كود التفرع الشرطي البسيط	
<code>x = 100</code>	المتغير (x) هو الذي يتم تجربته منطقياً ، وتم وضع قيمة مبدئية = 100
<code>If x > 50 :</code>	الجملة المنطقية تختبر ما إذا كانت x اكبر من 50
<code>print (" Big Number")</code>	إذا تحقق الشرط (سيتحقق فعلاً) يتم تنفيذ هذا الكود وهو طباعة جملة big number

التفرع الشرطي الثنائي

مفهوم التفرع الشرطي الثنائي

هي عملية برمجية يتم فيها اختبار الشرط ، إذا تحقق الشرط سيتم تنفيذ الكود ، وإذا لم يتحقق الشرط يتم تنفيذ كود آخر بديل.

كود التفرع الشرطي الثنائي (if else)

الذي يتم فيه اختبار الشرط الثنائي (تجربة شرط واحد + التصرف في حالة عدم حدوث هذا الشرط)

قاعدة كتابة كود التفرع الشرطي الثنائي	
variable = value	المتغير الذي يتم تجربته منطقياً
if condition :	الجملة المنطقية التي تجرب الشرط
Code 1	إذا تحقق الشرط المنطقي يتم تنفيذ هذا الكود
else :	كود يقوم بعمله إذا لم يتحقق الكود if
Code 2	الكود الذي سيتم تنفيذه من else نتيجة عدم تحقق if
مثال على كود التفرع الشرطي الثنائي	
x = 100	المتغير (x) هو الذي يتم تجربته منطقياً، ويساوي 100
if x > 1000 :	الجملة المنطقية تختبر ما إذا كانت x اكبر من 10000
print ("Big")	إذا تحقق الشرط (لن يتحقق) يتم تنفيذ كود الطباعة
else:	كود يقوم بعمله إذا لم يتحقق الكود if
print("small")	إذا لم يتحقق الشرط (لن يتحقق) يتم تنفيذ كود الطباعة

التفرع الشرطي المتعدد

مفهوم التفرع الشرطي المتعدد

يقوم بإختبار الشرط ، إذا تحقق ينفذ الكود ، وإذا لم يتحقق ينفذ الشرط الذي يليه ثم الذي يليه وهكذا ، فإذا لم يتحقق أي شرط سيقوم بتنفيذ الكود البديل

كود التفرع الشرطي المتعدد

والذي يتم فيه اختبار الشرط المتعدد (تجربة الشرط الأصلي - تجربة الشروط الأخرى البديلة - التصرف في حالة عدم حدوث أي شرط)

قاعدة كتابة كود التفرع الشرطي المتعدد	
variable = value	المتغير الذي يتم تجربته منطقياً
If condition :	الجملة المنطقية التي تجرب الشرط
Code 1	إذا تحقق الشرط المنطقي يتم تنفيذ هذا الكود
elif:	كود يقوم بعملة إذا لم يتحقق الكود if (يمكن تكراره أي عدد من المرات)
Code 2	كود الشرط التالي
else :	كود يقوم بعمله إذا لم يتحقق الكود أي مما سبق
Code 3	الكود الذي سيتم تنفيذه من else نتيجة عدم تحقق أي مما سبق
مثال على كود التفرع الشرطي المتعدد	
code = 2	المتغير (code) هو الذي يتم تجربته منطقياً، وتم وضع قيمة مبدئية تساوي 2
If code == 1 :	الجملة المنطقية تختبر إذا كان المتغير code يساوي 1

<code>print (" one")</code>	إذا تحقق الشرط (لن يتحقق) يتم تنفيذ هذا الكود وهو طباعة جملة <code>big number</code>
<code>elif code == 2 :</code>	كود يقوم بعمله إذا لم يتحقق الكود <code>if</code> ، ويختبر ما إذا كان المتغير <code>code</code> يساوي 2
<code>print("two")</code>	إذا لم يتحقق الشرط (سيتحقق) يتم تنفيذ هذا الكود وهو طباعة جملة <code>small number</code>
<code>else:</code>	كود يقوم بعمله إذا لم يتحقق الكود أي مما سبق
<code>print("small Number")</code>	الكود الاحتياطي هو طباعة جملة <code>small number</code>

أنشطة تعليمية على الشرط والتفرع البرمجي

أولاً - التفرع الشرطي البسيط

رقم النشاط	عنوان النشاط	وصف النشاط	توقيت التنفيذ	طريقة التنفيذ
1	أكتب الكود البرمجي	اكتب كود الشرط البسيط الذي يهدف لاختبار ما إذا كان المتغير y أكبر من 50 ، فإذا تحقق يتم طباعة كلمة Good	خلال المحاضرة	تعليم تعاوني مجموعات صغيرة
2	إستنتج نتيجة تنفيذ كود برمجي	استنتج نتيجة تنفيذ الكود التالي If $z == 323$: print ("Well Done")		
3	اكتب برنامج بايثون	اكتب برنامج بايثون يطلب فيه من المستخدم إدخال كلمة المرور ، فإذا كانت صحيحة يتم طباعة جملة (Access Allowed) ، (كلمة المرور = 123		

رقم النشاط	عنوان النشاط	وصف النشاط	توقيت التنفيذ	طريقة التنفيذ
4	إستخرج الخطأ البرمجي	استخرج الخطأ البرمجي في الكود التالي samy = 30 adel = 31 If mostafa > omar: print ("omar is older")		

ثانياً - التفرع الشرطي الثنائي

رقم النشاط	عنوان النشاط	وصف النشاط	توقيت التنفيذ	طريقة التنفيذ
1	أكتب الكود البرمجي	اكتب كود الشرط البسيط الذي يهدف لاختبار ما إذا كان المتغير y أكبر من 50 ، فإذا تحقق يتم طباعة كلمة Good ، وإذا لم يتحقق يتم طباعة كلمة bad		
2	إستنتج نتيجة تنفيذ كود برمجي	استنتج نتيجة تنفيذ الكود التالي If z == 323: print ("Well Done") else: print ("Not Well ")	خلال المحاضرة	تعليم تعاوني مجموعات صغيرة
3	اكتب برنامج بايثون	اكتب برنامج بايثون يطلب فيه من المستخدم إدخال كلمة المرور ، فإذا كانت صحيحة يتم طباعة جملة (Access Allowed) ، وإذا لم تكن صحيحة يتم طباعة جملة		

رقم النشاط	عنوان النشاط	وصف النشاط	توقيت التنفيذ	طريقة التنفيذ
		كلمة (Acces Denied) المرور الصحيحة = 123		
4	إستخرج الخطأ البرمجي	استخرج الخطأ البرمجي في الكود التالي Omar = 32 Ali = 31 If ali > omar: print ("Ali is older") else: print ("Omar is older")		

ثالثاً - التفرع الشرطي المتعدد

رقم النشاط	عنوان النشاط	وصف النشاط	توقيت التنفيذ	طريقة التنفيذ
1	أكتب الكود البرمجي	اكتب كود الشرط البسيط الذي يهدف لاختبار ما إذا كان المتغير mark أكبر من أو يساوي 100 يتم طباعة كلمة very good ، وإذا كان أقل من 100 وأكبر من صفر يطبع جملة good ، وإذا كان غير ذلك يطبع جملة bad	خلال المحاضرة	تعليم تعاوني مجموعات صغيرة
2	إستنتج نتيجة تنفيذ كود برمجي	استنتج نتيجة تنفيذ الكود التالي <pre> if age < 18: print ("kid") elif age >= 18 and age <= 40: print ("Young") else: print ("Old") </pre>		

رقم النشاط	عنوان النشاط	وصف النشاط	توقيت التنفيذ	طريقة التنفيذ
3	اكتب برنامج بايثون	اكتب برنامج بايثون يطلب فيه من المستخدم إدخال كود الخدمة ، فإذا كان (1) يطبع جملة (Charge) ، وإذا كان (2) يتم طبع جملة (Transfer) ، وإذا كان (3) يتم طباعة جملة (Change) ، وإذا لم يختار المستخدم أي منهم يتم طباعة جملة (Wrong Code)		
4	إستخرج الخطأ البرمجي	استخرج الخطأ البرمجي في الكود التالي <pre> If code == 1: print ("Charge") elif code == 2: print ("Transfer") elif code == 3: print ("Change </pre>		

تكليفات برمجية

قم بتنفيذ التكليفات التالية، ثم تواصل مع أستاذ المادة لتسليمها ومناقشتها في المحاضرات التالية:

أولاً- التفرع الشرطي البسيط:

م	التكليف
1	اكتب كود الشرط البسيط الذي يهدف لاختبار ما إذا كان المتغير <code>cost</code> أقل من أو يساوي 120 ، فإذا تحقق يتم طباعة جملة (I have enough money)
2	استنتج نتيجة تنفيذ الكود التالي <code>Salem_age = 70</code> <code>If Salem_age < 40:</code> <code>print ("He is Young")</code>
3	اكتب برنامج بايثون يطلب فيه من المستخدم إدخال رصيده، فإذا كان مناسب يتم طباعة (Buy Allowed) ، (الرصيد المطلوب يكون أكثر من 10000)
4	استخرج الخطأ البرمجي في الكود التالي <code>samy = 30</code> <code>adel = 31</code> <code>If mostafa > omar:</code> <code>print ("omar is older")</code>

ثانياً - التفرع الشرطي الثنائي:

م	التكليف
1	اكتب كود الشرط البسيط الذي يهدف لاختبار ما إذا كان المتغير marks أكبر من أو يساوي 50 ، فإذا تحقق يتم طباعة جملة (Successful) ، وإذا لم يتحقق تم طباعة جملة (Unsuccessful)
2	استنتج نتيجة تنفيذ الكود التالي <pre>Salem_age = 70 If Salem_age < 40: print ("He is Young") else: print ("He is old")</pre>
3	اكتب برنامج بايثون يطلب فيه من المستخدم إدخال رصيده، فإذا كان يكفي يتم طباعة (Buy Allowed) ، وإذا لم يكفي يتم طباعة (Buy is not allowed) (الرصيد المطلوب يكون أكثر من 10000)
4	استخرج الخطأ البرمجي في الكود التالي <pre>mostafa = 32 omar = 31 If mostafa > omar: print ("omar is older") else print ("Adel is older")</pre>

ثالثاً - التفرع الشرطي المتعدد:

م	التكليف
1	اكتب كود الشرط البسيط الذي يهدف لاختبار ما إذا كان المتغير x أكبر من 10 يتم طباعة كلمة bigger، وإذا كان يساوي 10 يطبع جملة equal، وإذا كان غير ذلك يطبع جملة smaller
2	استنتج نتيجة تنفيذ الكود التالي <pre> If x < 5 : print ("good") elif x >= 5 and x <=40: print (" a lot") else: print ("wrong number") </pre>
3	اكتب برنامج بايثون يطلب فيه من المستخدم إدخال كود ، فإذا كان (1) يطبع جملة (one) ، وإذا كان (2) يتم طبع جملة (two) ، وإذا كان (3) يتم طباعة جملة (three) ، وإذا لم يختار المستخدم أي منهم يتم طباعة جملة (another code)
4	استخرج الخطأ البرمجي في الكود التالي <pre> If x == 1: print ("Charge") elif code == 2: print ("Transfer") else: print ("Change") </pre>

مصادر إثرائية من بنك المعرفة المصري



من خلال المصادر الالكترونية التالية والموجودة ببنك المعرفة المصري والمكتبة الرقمية؛ يمكنك الاستزادة حول المحتوى العلمي للفصل:

م	عنوان المصدر	الرابط
1	دراسة عن كيفية تنمية مهارات البرمجة بلغة فيجوال بيسيك	https://0810gwg3s-1104-y-https-search-mandumah-com.mplbci.ekb.eg/MyResearch/Home?rurl=%2FRecord%2F990406
2	كتاب عن قواعد البرمجة	https://0811nxsk9-1106-y-https-ekbedu-ethraadl-com.mplbci.ekb.eg/Record/62475

قاموس المصطلحات الواردة بالفصل

م	المصطلح بالعربية	المصطلح بالإنجليزية	الوصف
1	الشرط والتفرع البرمجي	Branching	هو قيام البرنامج بإختبار شرط معين ، وبناء عليه يتفرع البرنامج إلى مسارات متعددة
2	عوامل المقارنات	Comparison Operators	هي الرموز التي تستخدم في العملية المنطقية لإختبار الشرط في بايثون ، فإذا كان الشرط متحقق تكون القيمة المرجعة True ، أو لا تتحقق فتكون False

اختبر نفسك

اختر الإجابة الصحيحة من بين البدائل الموجودة:

1		م
يستخدم الشرط البرمجي في		العبارة
تجزئة مكونات البرنامج	أ	البدائل
تفريع البرنامج لمسارات	ب	
تقليل سطور الكود	ج	
انشاء حدث برمجي	د	
2		م
من مراحل تنفيذ العمليات المنطقية البرمجية يتم فيها اختبار الشرط البرمجي		العبارة
العمليات	أ	البدائل
الادخال	ب	
الإخراج	ج	
التوزيع	د	

3		م
من عوامل المقارنات البرمجية يتم فيها اختبار ما اذا كانت قيمة متغير تساوي متغير آخر		العبارة
=	أ	البدائل
Equal	ب	
“ = “	ج	
= =	د	
4		م
عامل منطقي يستخدم لتنفيذ كود معين اذا تحققت جميع الشروط		العبارة
OR	أ	البدائل
NOT	ب	
AND	ج	
XOR	د	
5		م
اذا كانت العملية البرمجية تتطلب اختبار شرط اذا تحقق سيتم تنفيذ الكود واذا لم يتحقق سينتقل البرنامج تلقائياً للسطور التالية		العبارة
الشرط البسيط	أ	البدائل
الشرط المتعدد	ب	
الشرط المتوان	ج	
الشرط الثنائي	د	

6		م
أي من التالي يعبر كود تفرع شرطي بسيط		العبارة
<pre> If v == n : print (" not yet") else: print ("now") </pre>	أ	البدائل
<pre> If b == u : print (" the sum") elif b == m: print (" not the sum") else: print ("insert another code") </pre>	ب	
<pre> If h == 9: then print (z) </pre>	ج	
<pre> If a == b : print (" good") </pre>	د	

7		م
إذا كانت العملية البرمجية تتطلب اختبار شرط إذا تحقق سيتم تنفيذ الكود وإذا لم يتحقق سيتم تنفيذ كود بديل		العبارة
الشرط البسيط	أ	البدائل
الشرط المتعدد	ب	
الشرط الثنائي	ج	
الشرط البديل	د	

8		م
أي من التالي يعبر كود تفرع شرطي ثنائي		العبارة
<pre>If e == 5 : print (" good") else: print ("bad")</pre>	أ	البدائل
<pre>If f == 7 : print (" seven")</pre>	ب	
<pre>If a == 1 : print (" one") elif a == d: print (" two ") else:</pre>	ج	

print (" three")		
if a == 1		
continue	د	

9		م
اذا كانت العملية البرمجية تتطلب اختبار شرط اذا لم يتحقق سيتم تنفيذ الكود الذي يليه واذا لم يتحقق سينفذ الكود الذي يليه واذا لم يتحقق أي منهم سيتم تنفيذ الكود الاحتياطي		العبارة
الشرط البسيط	أ	البدائل
الشرط الثنائي	ب	
الشرط المتعدد	ج	
الشرط المتوازي	د	

10		م
أي من التالي يعبر كود تفرع شرطي متعدد		العبارة
<pre>If t == m : print (" equal") else: print ("not equal")</pre>	أ	البدائل
<pre>If e == 5 : print (" go") elif e == 6: print (" very good") else: print ("wrong")</pre>	ب	
<pre>If c == x : print (" the same")</pre>	ج	
<pre>if d == bt break</pre>	د	

مفتاح الإجابة

الإجابة الصحيحة	العبرة	م
تفرع البرنامج لمسارات	يستخدم الشرط البرمجي في	1
العمليات	من مراحل تنفيذ العمليات المنطقية البرمجية يتم فيها اختبار الشرط البرمجي	2
= =	من عوامل المقارنات البرمجية يتم فيها اختبار ما اذا كانت قيمة متغير تساوي متغير آخر	3
AND	عامل منطقي يستخدم لتنفيذ كود معين اذا تحققت جميع الشروط	4
الشرط البسيط	اذا كانت العملية البرمجية تتطلب اختبار شرط اذا تحقق سيتم تنفيذ الكود واذا لم يتحقق سينتقل البرنامج تلقائياً للسطور التالية	5
If a == b : print (" good")	أي من التالي يعبر كود تفرع شرطي بسيط	6
الشرط الثنائي	اذا كانت العملية البرمجية تتطلب اختبار شرط اذا تحقق سيتم تنفيذ الكود واذا لم يتحقق سيتم تنفيذ كود بديل	7

الإجابة الصحيحة	العبارة	م
<pre> If e == 5 : print (" good") else: print ("bad") </pre>	<p>أي من التالي يعبر كود تفرع شرطي ثنائي</p>	8
<p>الشرط المتعدد</p>	<p>إذا كانت العملية البرمجية تتطلب اختبار شرط إذا لم يتحقق سيتم تنفيذ الكود الذي يليه وإذا لم يتحقق سينفذ الكود الذي يليه وإذا لم يتحقق أي منهم سيتم تنفيذ الكود الاحتياطي</p>	9
<pre> If e == 5 : print (" go") elif e == 6 : print (" very good") else: print ("wrong") </pre>	<p>أي من التالي يعبر كود تفرع شرطي متعدد</p>	10

الفصل الخامس

الحلقات البرمجية Loops

مقدمة في الحلقات التكرارية

مفهوم الحلقات التكرارية في البرمجة

كود الحلقات التكرارية البرمجية

الدمج بين الحلقات التكرارية وبين الشروط والتنوع

مفهوم الدمج بين الحلقات التكرارية وبين الشروط والتنوع

كود الدمج بين الحلقات التكرارية وبين الشروط والتنوع

إلغاء إحدى لفات حلقة تكرارية

مفهوم إلغاء إحدى لفات حلقة تكرارية

كود إلغاء إحدى لفات حلقة تكرارية

إلغاء جميع لفات حلقة تكرارية

مفهوم إلغاء جميع لفات حلقة تكرارية

كود إلغاء جميع لفات حلقة تكرارية

مقدمة الفصل

أولاً - الهدف العام:

إكساب طلاب تكنولوجيا التعليم الرقمي المفاهيم والمهارات الأساسية الخاصة بالعمليات على الحلقات التكرارية البرمجية.

ثانياً - الأهداف التعليمية:

بعد الانتهاء من هذا الفصل /الدرس، يتوقع من الطالب أن يكون قادراً على أن:

1. يذكر مفهوم الحلقات التكرارية في البرمجة
2. يكتب كود الحلقات التكرارية البرمجية
3. يذكر مفهوم الدمج بين الحلقات التكرارية وبين الشرط والتفرع
4. يكتب كود الدمج بين الحلقات التكرارية وبين الشرط والتفرع
5. يذكر مفهوم إلغاء إحدى لغات حلقة تكرارية
6. يكتب كود إلغاء إحدى لغات حلقة تكرارية
7. يذكر مفهوم إلغاء جميع لغات حلقة تكرارية
8. يكتب كود إلغاء جميع لغات حلقة تكرارية

مقدمة في الحلقات التكرارية

مفهوم الحلقات التكرارية في البرمجة

- هي أكواد برمجية تتيح للمبرمج تنفيذ الكود عدة مرات دون الحاجة لإعادة كتابته
- فعلى سبيل المثال إذا أراد المبرمج طباعة جملة (welcome) عشرة آلاف مرة ، كل ما عليه هو كتابة كود حلقة تتكرر عشرة آلاف مرة (وهذا يتطلب منه سطر كود واحد فقط) ثم وضع جملة الطباعة المطلوبة بداخله ، وعند تنفيذ البرنامج سيتم تكرار الحلقة العدد المطلوب من المرات وطباعة الجملة بداخلها.

كود الحلقات التكرارية البرمجية

توجد طريقتين لكتابة كود الحلقات التكرارية البرمجية في بايثون:

أولاً - الحلقة التكرارية البسيطة

هي حلقة تكرارية، عدد التكرارات فيها يساوي رقم محدد

قاعدة كتابة كود الحلقة التكرارية البسيطة

for variable in range (value) :

code

حلقة تكرارية - تعتمد على متغير variable - سنقوم بعدد من التكرارات - يبدأ من (0) ، ينتهي بقيمة value ناقص واحد

مثال على كود الحلقة التكرارية البسيطة

```
for m in range (5) :
    print (" well done")
```

حلقة تكرارية - تعتمد على المتغير m - ستقوم بعدد من التكرارات - يبدأ من (0) ، ينتهي بقيمة 4 - بالتالي سيتم طباعة جملة (well done) 5 مرات

ثانياً - الحلقة التكرارية ذات المدى

هي حلقة تكرارية، عدد التكرارات فيها لها مدى - يبدأ من القيمة الأولى وينتهي قبل الوصول للقيمة الثانية بفرق واحد صحيح

قاعدة كتابة كود الحلقة التكرارية ذات المدى

```
for variable in range (value1 , value 2) :
    code
```

حلقة تكرارية - تعتمد على متغيرين - ستقوم بعدد من التكرارات - يبدأ من (value 1) ، ينتهي بقيمة value2 ناقص واحد

مثال على كود الحلقة التكرارية ذات المدى

```
for x in range (1 , 10) :
    print (" wellcome")
```

حلقة تكرارية - تعتمد على متغيرين - ستقوم بعدد من التكرارات - يبدأ من (1) ، ينتهي بقيمة 9 - بالتالي سيتم طباعة جملة (well done) 9 مرات

الدمج بين الحلقات التكرارية وبين الشرط والتفرع

مفهوم الدمج بين الحلقات التكرارية وبين الشرط والتفرع

تعتمد الكثير من المسائل البرمجية على الدمج بين : التكرار بالكود for ، ووضع شرط داخل التكرار بالكود if .

كود الدمج بين الحلقات التكرارية وبين الشرط والتفرع

والذي يتم فيه وضع كود if داخل حلقة for

قاعدة كتابة كود دمج بين الحلقات والشرط البرمجي	
for variable in range (value) :	سطر انشاء حلقة تكرارية
if condition :	جملة شرطية غالبا ما تعتمد على تجربة قيمة للمتغير variable
Code	اذا تحقق الشرط سينفذ هذا الكود
مثال على كود دمج بين الحلقات والشرط البرمجي	
for z in range (10 , 15) :	سطر انشاء حلقة تكرارية
if z == 11 :	جملة شرطية تعتمد على تجربة قيمة للمتغير z
print (" Hello")	اذا تحقق الشرط بأن z تساوي 11 (سيتحقق فعلا لأن z ستتراوح قيمتها بين 10 و 14) سيتم طباعة كلمة hello

الغاء إحدى لفات حلقة تكرارية

مفهوم إغفاء إحدى لفات حلقة تكرارية

طريقة برمجية ، تعتمد على الدمج بين (كود الحلقة التكرارية + كود الشرط + الكود continue) ، تستخدم لإيقاف الحلقة (تلغي لفة واحدة) عند تحقق شرط معين ، وتستكمل الحلقة دوراتها بعد ذلك

كود إغفاء إحدى لفات حلقة تكرارية

والذي يتم فيه الغاء لفة من حلقة تكرارية بناء على شرط محدد

قاعدة كتابة كود الغاء لفة من حلقة برمجية	
for variable in range (value) :	سطر انشاء حلقة تكرارية
If condition :	جملة شرطية غالبا ما تعتمد على تجربة قيمة للمتغير variable
continue	كود يلغي عمل السطور التالية داخل الحلقة ، ولمدة لفة واحدة فقط
Code	كود سينفذ في جميع لفات الحلقة ، ماعدا اللفة التي يتحقق فيها الشرط وينفذ فيها كود continue
مثال على كود الغاء لفة من حلقة برمجية	
for r in range (2,7) :	سطر انشاء حلقة تكرارية

If r == 5 :	جملة شرطية تعتمد على تجربة اذا كان المتغير r يساوي 5
continue	كود يلغي عمل السطور التالية داخل الحلقة ، ولمدة لفة واحدة فقط (اذا كانت r تساوي 5 فقط)
print (r)	كود سينفذ في جميع لفات الحلقة (طباعة قيمة r والتي ستتراوح بين 2 و 6) ، ماعدا اللفة التي يتحقق فيها الشرط (r تساوي 5) وينفذ فيها كود continue

الغاء جميع لفات حلقة تكرارية

مفهوم إغناء جميع لفات حلقة تكرارية

طريقة برمجية ، تعتمد على الدمج بين (كود الحلقة التكرارية + كود الشرط + الكود break) ، تستخدم لإيقاف الحلقة (كاملة بجميع لفاتها) عند تحقق شرط معين .

كود إغناء جميع لفات حلقة تكرارية

والذي يتم فيه الغناء جميع لفات حلقة تكرارية بناء على شرط محدد

قاعدة كتابة كود جميع لفات حلقة تكرارية حلقة برمجية	
for variable in range (value) :	سطر انشاء حلقة تكرارية
If condition :	جملة شرطية غالبا ما تعتمد على تجربة قيمة للمتغير variable
break	كود يلغي عمل جميع لفات الحلقة
Code	كود سينفذ في حالة عدم حدوث الشرط ، واذا حدث الشرط في اللفة الأولى لن ينفذ أصلاً
مثال على كود الغناء لفة من حلقة برمجية	
for r in range (2,7) :	سطر انشاء حلقة تكرارية
If r == 5 :	جملة شرطية تعتمد على تجربة اذا كان المتغير r يساوي 5

break	كود يلغي عمل الحلقة كاملة (إذا كانت r تساوي 5)
print (r)	كود سينفذ في حالة عدم حدوث الشرط (سينفذ عندما تكون r تساوي 2 و 3 و 4) وعند وصول قيمة r إلى 5 ستلغى الحلقة

أنشطة تعليمية على الحلقات التكرارية

رقم النشاط	عنوان النشاط	وصف النشاط	توقيت التنفيذ	طريقة التنفيذ
1	أكتب الكود البرمجي	اكتب الكود الذي يقوم بطباعة جملة (No) بعدد 25 مرة	خلال المحاضرة	تعليم تعاوني مجموعات صغيرة
2	إستنتج نتيجة تنفيذ كود برمجي	استنتج نتيجة تنفيذ الكود التالي for f in range (2, 9): print ("ok")		
3	اكتب برنامج بايثون	اكتب برنامج بايثون يتم فيه طباعة الأعداد الزوجية فقط من 1 إلى 10		
4	إستخرج الخطأ البرمجي	استخرج الخطأ البرمجي في الكود التالي for c in range (7) if c == 2 continue print (c)		

تكاليف برمجية

قم بتنفيذ التكاليف التالية، ثم تواصل مع أستاذ المادة لتسليمها ومناقشتها في المحاضرات التالية:

م	التكليف
1	اكتب الكود الذي يقوم بطباعة الأعداد من 100 إلى 1000
2	استنتج نتيجة تنفيذ الكود التالي <pre>for c in range : (7) if c == 2: continue print (c)</pre>
3	اكتب برنامج بايثون يتم فيه طباعة الأعداد الفردية فقط من 1 إلى 10
4	استخرج الخطأ البرمجي في الكود التالي <pre>for s in : (2,11,2) print (s)</pre>

مصادر إثرائية من بنك المعرفة المصري



Egyptian Knowledge Bank
بنك المعرفة المصري

من خلال المصادر الالكترونية التالية والموجودة ببنك المعرفة المصري والمكتبة الرقمية؛ يمكنك الاستزادة حول المحتوى العلمي للفصل:

م	عنوان المصدر	الرابط
1	كتاب عن برمجة الامتحانات وبنوك الأسئلة	https://0810gxsjr-1106-y-https-search-mandumah-com.mplbci.ekb.eg/Record/1008457
2	دراسة عن التدريب على البرمجة باستخدام التدريب الالكتروني	https://0810gxsre-1106-y-https-search-mandumah-com.mplbci.ekb.eg/Record/960166

قاموس المصطلحات الواردة بالفصل

م	المصطلح بالعربية	المصطلح بالإنجليزية	الوصف
1	الحلقات التكرارية	Loop	هي أكواد برمجية تتيح للمبرمج تنفيذ الكود عدة مرات دون الحاجة لإعادة كتابته
2	الحلقات التكرارية البسيطة	Simple Loop	حلقة تكرارية ، عدد التكرارات فيها يساوي الرقم المحدد
3	الحلقة التكرارية ذات المدى	Ranged Loop	حلقة تكرارية ، عدد التكرارات فيها لها مدى يعتمد على قيمتين

اختبر نفسك

اختر الإجابة الصحيحة من بين البدائل الموجودة:

1		م
تستخدم الحلقات التكرارية في البرمجة في		العبارة
تكرار الكود	أ	البدائل
تفرع الكود	ب	
تصحيح الكود	ج	
غلق الكود	د	
2		م
أي من التالي هو مثال على كود حلقة تكرارية بسيط		العبارة
for b in range (2 , 6) : print (" dell")	أ	البدائل
for c in range (1 , 33) : if c == 11 : print (" Yes")	ب	
for b in range (11) : print (" good")	ج	
for h in range (10): if h == 11 : print (" say")	د	

3		م
أي من التالي من استخدامات الدمج بين الشرط والتكرار البرمجي		العبارة
تنفيذ حلقة لها مدى تكراري محدد	أ	البدائل
تنفيذ شرط برمجي ثنائي	ب	
تنفيذ حلقة تكرارية ثنائية	ج	
الغاء احدى لغات حلقة تكرارية بناء على شرط برمجي	د	
4		م
أي من التالي مثال على الكود المستخدم في الدمج بين الشرط والتكرار		العبارة
for c in range (33) : print (" plain")	أ	البدائل
for c in range (1 , 33) : if c == 11 : print (" Yes")	ب	
for z in range (5 , 33) : print (" right")	ج	
for s in range (8 , 11) : print (" left")	د	

5		م
كود يستخدم في الغاء لفة من لفات حلقة تكرارية		العبارة
continue	أ	البدائل
Break	ب	
Range	ج	
For	د	
6		م
أي من التالي مثال على الكود المستخدم في الغاء احدى لفات حلقة تكرارية		العبارة
for a in range (16) : print (" hp")	أ	البدائل
for c in range (7 , 21) : print (" add")	ب	
for s in range (20) : If s == 13 : print (" we")	ج	
for b in range (55) : If b == 13 : continue print (b)	د	

7		م
كود يستخدم في الغاء جميع لفات حلقة تكرارية		العبارة
Continue	أ	البدائل
Break	ب	
In range	ج	
For + if	د	
8		
أي من التالي مثال على الكود المستخدم في الغاء جميع لفات حلقة تكرارية		العبارة
for d in range (77) : print (" win")	أ	البدائل
for b in range (1 , 11) : print (" ram")	ب	
for r in range (30) : if r == 14 : print (" some")	ج	
for g in range (7) : if g == 3 : break print (g)	د	

مفتاح الإجابة

الإجابة الصحيحة	العبرة	م
تكرار الكود	تستخدم الحلقات التكرارية في البرمجة في	1
for b in range(11) : print (" good")	أي من التالي هو مثال على كود حلقة تكرارية بسيط	2
الغاء احدى لغات حلقة تكرارية بناء على شرط برمجي	أي من التالي من استخدامات الدمج بين الشرط والتكرار البرمجي	3
for c in range (1 , 33 : If c == 11 : print (" Yes")	أي من التالي مثال على الكود المستخدم في الدمج بين الشرط والتكرار	4
continue	كود يستخدم في الغاء لفة من لغات حلقة تكرارية	5
for b in range (55) : If b == 13 : continue print (b)	أي من التالي مثال على الكود المستخدم في الغاء احدى لغات حلقة تكرارية	6
Break	كود يستخدم في الغاء جميع لغات حلقة تكرارية	7
for g in range (7) : If g == 3 : break print (g)	أي من التالي مثال على الكود المستخدم في الغاء جميع لغات حلقة تكرارية	8

الفصل السادس

النوال البرمجية Functions

مقدمة في النوال البرمجية

مفهوم النوال في البرمجة

استخدامات النوال في البرمجة

النوال الجاهزة في بايثون

مفهوم النوال الجاهزة

كود النوال الجاهزة

النوال الغير جاهزة في بايثون

مفهوم النوال الغير جاهزة

كود النوال الغير جاهزة

الدالة المختصرة lamda

مفهوم الدالة المختصرة lamda

كود الدالة المختصرة lamda

مقدمة الفصل

أولاً - الهدف العام:

إكساب طلاب تكنولوجيا التعليم الرقمي المفاهيم والمهارات الأساسية الخاصة بالعمليات على الدوال البرمجية.

ثانياً - الأهداف التعليمية:

بعد الانتهاء من هذا الفصل /الدرس، يتوقع من الطالب أن يكون قادراً على أن:

1. يذكر مفهوم الدوال في البرمجة
2. يوضح استخدامات الدوال في البرمجة
3. يذكر مفهوم الدوال الجاهزة
4. يكتب أكواد بعض الدوال الجاهزة
5. يذكر مفهوم الدوال الغير جاهزة
6. يكتب كود الدوال الغير جاهزة
7. يذكر مفهوم الدالة المختصرة lamda
8. يكتب كود الدالة المختصرة lamda

مقدمة في الدوال البرمجية

مفهوم الدوال في البرمجة

- عبارة عن مجموعة مجمعة من الأكواد في مكان واحد ولها وظيفة محددة ، ولها اسم يشير إليها
- عند الحاجة لاستخدام تلك الاكواد يكتفى فقط بكتابة اسمها نيابة عن كتابة جميع سطور الكود التي بداخلها (حتى لو الف سطر)

استخدامات الدوال في البرمجة

يتم استخدام الدوال لتحقيق الفوائد التالية:

- **تقليل عدد سطور الكود:** عند الاحتياج لأكواد الدالة يتم استدعاء الاسم فقط ، ولأي عدد من المرات ، وعند الحاجة في أي وقت ، دون كتابة اكوادها الداخلية
- **تسهيل تصحيح وتعديل الكود :** حيث يكتفى بتعديل اكواد الدالة نفسها ويسري هذا التعديل فوراً في باقي المشروع البرمجي ، وفي كل الأماكن التي تستدعي فيها الدالة

الدوال الجاهزة في بايثون

مفهوم الدوال الجاهزة

دوال تم تصميمها ووضعها داخل مكتبة اللغة البرمجية ، مثل الدوال الرياضية (min , max) والدوال الإحصائية ، وهي تقوم بوظيفتها فور استدعائها ، تزداد قوة لغة البرمجة بقوة الدوال البرمجية الجاهزة التي توفرها للمبرمج

كود الدوال الجاهزة

والذي يتم فيه استدعاء الدالة لتقوم بعملها على متغير أو قيمة محدد :

قاعدة كتابة اكواد تستفيد من الدوال الجاهزة	
<code>variable = function (value)</code>	طريقة 1 : انشاء متغير قيمته تساوي العملية على قيمة
<code>variable = function (another variable)</code>	طريقة 2 : انشاء متغير قيمته تساوي العملية على متغير
<code>print (function (variable))</code>	طريقة 3 : طباعة ناتج دالة على متغير اخر
<code>print (function (value))</code>	طريقة 4 : طباعة ناتج دالة على قيمة معينة مباشرة دون وجود متغيرات
مثال على دوال جاهزة	
<code>print (max (x , y , z))</code>	دالة (max) تقوم بإيجاد القيمة الرقمية الأعلى من بين مجموعة قيم (طريقة 3)
<code>print (min (x , y , z))</code>	دالة (min) تقوم بإيجاد القيمة الرقمية الأقل من بين مجموعة قيم (طريقة 3)
<code>print (sum (x , y , z))</code>	دالة (sum) تقوم بحساب حاصل الجمع

<code>print (pow (x , y , z))</code>	دالة (pow) تقوم بإيجاد قيمة عدد الأول مرفوعا إليه (أس) العدد الثاني (طريقة 3)
<code>print (len (x))</code>	دالة (len) تقوم بإيجاد عدد حروف كلمة (طريقة 3)

الدوال الغير جاهزة في في بايثون

مفهوم الدوال الغير جاهزة

- دالة غير مخزنة في لغة البرمجة ، وبالتالي يضطر المبرمج لتصميمها وكتابتها (في مكان مخصص في المشروع مع باقي الدوال الغير جاهزة)
- تقوم بوظيفتها عند استدعاء اسمها (بعد كتابتها كاملة من المبرمج)
- هي تنقسم لنوعين :
 - الأول لا يحتوي على parameters ويتم استدعاء اسمها فقط
 - الثاني تحتوي على parameters ويتم استدعاؤها لتنفيذ مهام على متغيرات محددة

كود الدوال الغير جاهزة

يتم كتابة كود الدوال الغير جاهزة على النحو التالي:

قاعدة كتابة كود دالة غير جاهزة لا تحتوي على parameters	
def function_name () : code	جزء تعريف الدالة (اسم الدالة + اكوادها) ويتم كتابته في مكان مخصص للدوال
function_name ()	جزء استدعاء الدالة من خلال اسمها فقط
مثال على كود دالة غير جاهزة لا تحتوي على parameters	
def greetning (): print (“ Hello “)	جزء تعريف الدالة - اسم الدالة greeting ، الكود هو طباعة كلمة Hello
greetning ()	استدعاء الدالة (طباعة كلمة Hello)

قاعدة كتابة كود دالة غير جاهزة تحتوي على parameters	
<pre>def function_name (parameters): code return (variable) ;</pre>	<p>جزء تعريف الدالة (اسم الدالة + متغيرتها + parameters اكوادها) ويتم كتابته في مكان مخصص للدوال</p>
<pre>function_name(parameters)</pre>	<p>جزء استدعاء الدالة من خلال اسمها + قيم ومتغيرات تحل محل البارامترز</p>
مثال على كود كود دالة غير جاهزة تحتوي على parameters	
<pre>def multiply (x , y): z = x + y return z ;</pre>	<p>جزء تعريف الدالة - اسم الدالة greeting ، الكود هو طباعة كلمة Hello</p>
<pre>multiply (5 , 6)</pre>	<p>طريقة لاستدعاء الدالة بدلالة قيم</p>
<pre>multiply (a , b)</pre>	<p>طريقة لاستدعاء الدالة بدلالة متغيرات</p>
<pre>print (multiply (a , b))</pre>	<p>في حالة الرغبة في دمج كود الطباعة مع استدعاء الدالة</p>

الدالة المختصرة lamda

مفهوم الدالة المختصرة lamda

هي صورة مختصرة لكتابة الدوال ، تعتمد على وجود القيمة المدخلة والعملية في سطر واحد ، ولا يوجد لها إسم

كود الدالة المختصرة lamda

والذي يتم فيه :

- انشاء دالة (lamda)
- انشاء متغير يحمل قيمة ناتج الدالة (variable)
- وضع متغيرات الدالة (parameter 1 , 2)
- عمل معادلة خاصة بمتغيرات الدالة (formula)
- إيجاد نتيجة الدالة ووضعها داخل المتغير

قاعدة كتابة كود الدالة المختصرة lamda	
<code>variable=lambda parameter_1,parameter_2: formula</code>	التعريف
<code>variable (values or other variables)</code>	الاستخدام
مثال على كود الدالة المختصرة lamda	
<code>multiply= lambda x , y : x + y</code>	التعريف
<code>multiply (a , b)</code>	استخدام 1
<code>multiply (5 , 6)</code>	استخدام 2
<code>print (multiply (a , b))</code>	طباعة

أنشطة تعليمية على الدوال البرمجية

رقم النشاط	عنوان النشاط	وصف النشاط	توقيت التنفيذ	طريقة التنفيذ
1	أكتب الكود البرمجي	اكتب الكود الخاص بدالة تقوم بضرب العدد في 25		
2	إستنتج نتيجة تنفيذ كود برمجي	استنتج نتيجة تنفيذ الكود التالي def account1 (f , v): s = f * v / 2 return s;		
3	اكتب برنامج بايثون	اكتب برنامج بايثون يتم فيه إنشاء دالة تقوم بحساب قيمة الضرائب على الدخل بمعلومية المبلغ الذي يدخله المستخدم ، (الضرائب = 5% من الدخل)	خلال المحاضرة	تعليم تعاوني مجموعات صغيرة
4	إستخرج الخطأ البرمجي	استخرج الخطأ البرمجي في الكود التالي def sell_phones(b , v): s = f * v / 2 return s;		

تكاليف برمجية

قم بتنفيذ التكاليف التالية، ثم تواصل مع أستاذ المادة لتسليمها ومناقشتها في المحاضرات التالية:

م	التكليف
1	اكتب الكود الخاص بدالة تقوم بإيجاد حاصل ضرب 3 أعداد ، بالإضافة إلى كود الاستدعاء
2	استنتج نتيجة تنفيذ الكود التالي <pre>def account1 (f , v): s = f * v / 2 return s ;</pre>
3	اكتب برنامج يتم فيه تجربة كلمة المرور التي يدخلها المستخدم ، فإذا كانت صحيحة يتم طباعة جملة السماح بالدخول ، وإذا لم تكن صحيحة يتم طباعة جملة منع الدخول ، (كلمة السر = 123)
4	استخرج الخطأ البرمجي في الكود التالي <pre>def sc (m , n) return m / n * 2</pre>

مصادر إثنائية من بنك المعرفة المصري



Egyptian Knowledge Bank
بنك المعرفة المصري

من خلال المصادر الالكترونية التالية والموجودة ببنك المعرفة المصري والمكتبة الرقمية؛ يمكنك الاستزادة حول المحتوى العلمي للفصل:

م	عنوان المصدر	الرابط
1	دراسة عن برمجة الروبوت التعليمي	https://0810gxsre-1106-y-https-search-mandumah-com.mplbci.ekb.eg/Record/1108594
2	دراسة بحثية عن أهمية تدريس البرمجة في المدارس	https://0810gxsjr-1106-y-https-search-mandumah-com.mplbci.ekb.eg/Record/252220

قاموس المصطلحات الواردة بالفصل

م	المصطلح بالعربية	المصطلح بالإنجليزية	الوصف
1	الدوال البرمجية	Function	عبارة عن مجموعة مجمعة من الأكواد في مكان واحد ولها وظيفة محددة عندما يتم استدعائها وربطها بمتغير محدد (من خلال سطر كود واحد به اسم الدالة واسم المتغير) ، يتم تنفيذ كافة أكواد الدالة على هذا المتغير
2	المعطيات	Parameters	تعتمد بعض الدوال البرمجية على وجود معطيات داخلية في المعادلة ، قد تكون قيم أو متغيرات
3	لامدا	Lampda	طريقة برمجية لكتابة الدوال في سطر واحد يشمل المعطيات والمعادلة والنتائج معاً

اختبر نفسك

اختر الإجابة الصحيحة من بين البدائل الموجودة:

1	م
مجموعة من الأكواد في مكان محدد داخل المشروع البرمجي ولها وظيفة محددة	
الدوال البرمجية	أ
المتغيرات البرمجية	ب
الحلقات البرمجية	ج
المكتبات البرمجية	د
2	م
أي من التالي من ليس من استخدامات الدوال البرمجية	
تقليل سطور الكود	أ
سهولة تعديل الكود	ب
اختصار وقت المبرمج	ج
انشاء كائنات برمجية	د
3	م
دوال موجودة بمكتبة لغة البرمجة	
الدوال الغير جاهزة	أ
الدوال الجاهزة	ب
الدوال المختصرة	ج
الدوال الرسومية	د

4		م
دالة جاهزة تقوم بإيجاد قيمة العدد الأول مرفوعا إليه (أس) العدد الثاني		العبارة
Max	أ	البدائل
Lamda	ب	
Sum	ج	
Pow	د	
5		م
دالة يضطر المبرمج لكتابة اكوادها كاملة		العبارة
الدوال الجاهزة	أ	البدائل
دالة lamda	ب	
الدالة الهيكلية	ج	
الدوال غير جاهزة	د	
6		م
أي من التالي يعبر عن سطر تعريف دالة برمجية		العبارة
return (x);	أ	البدائل
def abc () :	ب	
Function name:	ج	
A = def (a):	د	

7		م
دالة برمجية ليس لها اسم		العبارة
الدالة الجاهزة	أ	البدائل
الدالة المزدوجة	ب	
الدالة الكائنية	ج	
الدالة المختصرة	د	
8		م
أي من التالي كود دالة تكتب في سكر واحد		العبارة
Lamda	أ	البدائل
Class	ب	
Max	ج	
Odd	د	

مفتاح الإجابة

م	العبارة	الإجابة الصحيحة
1	مجموعة من الأكواد في مكان محدد داخل المشروع البرمجي ولها وظيفة محددة	الدوال البرمجية
2	أي من التالي من ليس من استخدامات الدوال البرمجية	انشاء كائنات برمجية
3	دوال موجودة بمكتبة لغة البرمجة	الدوال الجاهزة
4	دالة جاهزة تقوم بإيجاد قيمة العدد الأول مرفوعا إليه (أس) العدد الثاني	Pow
5	دالة يضطر المبرمج لكتابة اكوادها كاملة	الدوال غير جاهزة
6	أي من التالي يعبر عن سطر تعريف دالة برمجية	def abc () :
7	دالة برمجية ليس لها اسم	الدالة المختصرة
8	أي من التالي كود دالة تكتب في سكر واحد	Lamda

الفصل السابع

البرمجة الكائنية Object Oriented

مفهوم البرمجة الكائنية التوجه / البرمجة الشيئية

أهمية البرمجة الكائنية

مفهوم الفئات Class في البرمجة

مفهوم الكائنات object في البرمجة

الفرق بين الفئات Class وبين الكائنات Objects

كود بسيط لإنشاء كلاس وأوبجكت متفوعه منه

كود متقدم لإنشاء كلاس وأوبجكت متفوعه منه

مقدمة الفصل

أولاً - الهدف العام:

إكساب طلاب تكنولوجيا التعليم الرقمي المفاهيم والمهارات الأساسية الخاصة بالعمليات على البرمجة الكائنية.

ثانياً - الأهداف التعليمية:

بعد الانتهاء من هذا الفصل /الدرس، يتوقع من الطالب أن يكون قادراً على أن:

1. يذكر مفهوم البرمجة الكائنية التوجه / البرمجة الشيئية
2. يوضح أهمية البرمجة الكائنية
3. يذكر مفهوم الفئات Class في البرمجة
4. يذكر مفهوم الكائنات object في البرمجة
5. يفرق بين الفئات Class وبين الكائنات Objects
6. يكتب كود بسيط لإنشاء كلاس وأوبجكت متفرعه منه
7. يكتب كود متقدم لإنشاء كلاس وأوبجكت متفرعه منه

البرمجة الكائنية / الشيئية / Object Oriented

مفهوم البرمجة الكائنية التوجه / البرمجة الشيئية

- هي نمط متقدم من البرمجة يختلف عن البرمجة الهيكلية التقليدية
- في البرمجة الهيكلية يتم كتابة البرنامج في شكل أسطر متتالية
- أما في البرمجة الكائنية يتم تقسيم البرنامج لمجموعة أجزاء (كائنات) ، وكل جزء يحتوي على حزمة من المتغيرات والدوال والأكواد وطرق إدخال وإخراج البيانات ، ويهدف كل كائن لتحقيق إحدى وظائف البرنامج ككل
- بعد تصميم وبرمجة الكائنات ؛ يتم ربطها جميعاً معاً بواجهة الاستخدام

أهمية البرمجة الكائنية

يفيد استخدام البرمجة الكائنية فيما يلي:

- عدم تكرار الكود ، لأنه يكتب مرة واحدة ويتم إستدعاؤه بأي عدد من المرات (نفس فكرة الدوال البرمجية)
- عدم تداخل الكود ، لأن كل جزء يؤدي وظيفته ولا يتداخل في نفس مكان كود آخر يؤدي وظيفة أخرى
- سهولة صيانة وتعديل الجزء المطلوب ، بدلاً من فحص البرنامج ككل في البرمجة الهيكلية

- سندرس في البرمجة الكائنية (الفئات class ، الكائنات object ، الوراثة inheritance) إلا أنه توجد العديد من المفاهيم الأخرى

مفهوم الفئات Class في البرمجة

- هي حاوية برمجية تضم العديد من المتغيرات (اسمها داخل الكلاس property) والدوال والأكواد البرمجية ، ويمكن وضعها في ملف منفصل داخل المشروع البرمجي
- تضم أي لغة برمجة كائنية كلاسات خاصة (built in) يستخدمها المبرمجين في أداء العديد من الوظائف ، وفي لغة بايثون توجد موديولات (modules) تضم بداخلها عدد من الكلاس بهدف أداء وظائف محددة ، مثل : webbrowser للتعامل مع صفحات الويب ، mysql-connector للتعامل مع قواعد البيانات.
- توفر جميع لغات البرمجة الكائنية للمبرمجين إمكانية إنشاء كلاس خاصة بهم لأداء وظائف محددة في المشاريع البرمجية ، برنامج مدرسي يضم كلاس للطلاب وآخر للمدرسين على سبيل المثال
- يتم إنشاء الكلاس مرة واحدة في الكود ، وعند احتياجه يتم إنشاء كائن object منه

مفهوم الكائنات object في البرمجة

- هي نسخة من الكلاس الأصلي ، بمعنى أن المبرمج يقوم بإنشاء كلاس مرة واحدة ، وعند استخدامه يقوم بإنشاء نسخ منه لاستخدامها عدة مرات ، ودون إعادة لكتابة كود الكلاس مرة أخرى
- بمجرد إنشاء أوبجكت من كلاس ، سيضم هذا الأوبجكت جميع المتغيرات (properties) والأكواد الخاصة بالكلاس
- يشبه هذا مفهوم الدوال الذي سبق أن درسناه ، حيث يتم إنشاء الدوال مرة واحدة ثم يتم استدعاء الدالة وربطها بقيم ومتغيرات محددة بمجرد كتابة إسم الدالة ودون الحاجة لإعادة كتابتها مرة أخرى
- كلاً من الكلاس والدوال تطبق مفهوم البرمجة الكائنية ، إلا أن الكلاس أشمل وأعم حيث يمكن أن يحتوي بداخله على عدد كبير من الدوال نفسها

الفرق بين الفئات Class وبين الكائنات Objects

- يتم إنشاء الكلاس مرة واحدة في بداية التصميم ، يتم إنشاء الكائنات المتفرعة من الكلاس بأي عدد من المرات وحسب الحاجة.
- الكلاس الواحد قد يتفرع منه عدة أوبجكتس ، أما الأوبجكت فهو متفرع من كلاس واحد فقط

- القيم الموجود بالكلاس هي قيم إبتدائية ، أما القيم الموجودة في الأوبجكت فهي القيم التي ادخلها المستخدم ، والمطلوب فيها عمليات المعالجة والطباعة
- يمكن وضع الكلاس في ملف منفصل داخل المشروع البرمجي وهذا يسهل تنظيم البرنامج، يتم وضع الأوبجكت في المكان المطلوب وحسب الحاجة
- عند تعديل خصائص الكلاس ، سيتم بشكل تلقائي تعديل خصائص جميع الكائنات المتفرعة منه

كود بسيط لإنشاء كلاس وأوبجكت متفرعه منه

يتم انشاء الكلاس والابجكت (بالطريقة البسيطة) وفق المراحل التالية :

- في جزء الدوال والكلاس ، يتم :
 - انشاء الكلاس وتسميته
 - وضع الاكواد والمتغيرات الخاصة به بداخله
- في هيكل البرنامج الأصلي يتم :
 - انشاء اوبجت في سطر كود
 - وضع قيمة للمتغير الخاصة بالابجكت في سطر كود (لكل متغير سطر كود منفصل)

قاعدة كتابة كود انشاء كلاس بالطريقة البسيطة	
class Class_name :	انشاء كلاس
class_variable1 = standard value class_variable2 = standard value	انشاء متغيرات داخل الكلاس واسناد قيمة مبدئية
قاعدة كتابة كود اوبجت متفرعة من كلاس بالطريقة البسيطة	
Object_name = Class_name ()	انشاء الاوبجت
Object_name.class_variable1=object value1 Object_name.class_variable2=object value2	وضع قيم لمتغير داخل الاوبجت
مثال على كود انشاء كلاس واوبجت متفرعة وقيم داخل الاوبجت	
class car : price = 1000 production = 1900	انشاء كلاس (سيارة) وعمل متغير للسعر
fiat = car ()	انشاء اوبجت (سيارة فيات)
fiat . price = 50000 fiat . color = 2000	وضع قيمة لسعر السيارة فيات
print (fiat.price) print (fiat.color)	في حالة الرغبة في طباعة السعر

كود متقدم لإنشاء كلاس وأوبجكت متفرعه منه

في المشاريع البرمجية يتم يتم انشاء الكلاس والايوبجكت (بالطريقة المتقدمة) ، حيث تضاف دالة `init` للكلاس ، وبالتالي يتم انشاء الاوبجكت وازافة كل قيم متغيراته في سطر كود واحد ، ويتم هذا بالطريقة التالية:

- في جزء الدوال والكلاس ، يتم :

- انشاء الكلاس وتسميته
- انشاء دالة `init` داخل الكلاس ، ويتم وضع جميع المتغيرات وقيمها المبدئية بداخلها
- يمكن للكلاس أن يحتوى على دوال أخرى ، لكل دالة مهمة محددة داخل الكلاس مثل تنفيذ العمليات الحسابية والمنطقية

- في هيكل البرنامج الأصلي يتم :

- انشاء اوبجكت + وضع قيم جميع المتغيرات الخاصة بالايوبجكت في سطر واحد مجمع (نظرا لوجود دالة `init`)

قاعدة كتابة كود انشاء كلاس بالطريقة المتقدمة	
<code>class Class_name :</code>	انشاء كلاس
<code>Def __init__(self , parameter1, parameter2) :</code>	انشاء دالة <code>init</code>
<code>self.parameter1 = parameter1</code> <code>self.parameter2 = parameter2</code>	انشاء متغيرات الكلاس (لابد ان يتماثل اسم المتغير مع قيمته

قاعدة كتابة كود اوبجت متفرعة من كلاس بالطريقة المتقدمة	
Object_name = Class_name (object value1 , object value2)	انشاء الاوبجكت ووضع القيم كلها مرة واحدة
مثال على كود انشاء كلاس واوبجكت متفرعة وقيم داخل الاوبجكت	
class car :	انشاء كلاس
Def __init__(self , price , color) :	انشاء دالة init
self.price = price self.color = color	انشاء متغيرات الكلاس
Object_name = Class_name (50000 ,silver)	انشاء الاوبجكت ووضع القيم كلها مرة واحدة

أنشطة تعليمية على البرمجة الكائنية

أولاً - الفئات Class

رقم النشاط	عنوان النشاط	وصف النشاط	توقيت التنفيذ	طريقة التنفيذ
1	أكتب الكود البرمجي	اكتب الكود الخاص بكلاس إفتراضي لمروحة fan (لها متغيرين model ، price)	خلال المحاضرة	تعليم تعاوني مجموعات صغيرة
2	إستنتج نتيجة تنفيذ كود برمجي	استنتج نتيجة تنفيذ الكود التالي class Processor : def __init__ (self , model, price): self.model = model self.price = price		
3	اكتب برنامج بايثون	اكتب الكود الخاص بكلاس يحسب مساحة مستطيل		
4	إستخرج الخطأ البرمجي	استخرج الخطأ البرمجي في الكود التالي		

رقم النشاط	عنوان النشاط	وصف النشاط	توقيت التنفيذ	طريقة التنفيذ
		<pre>class C_area def circle_area(self): area = 3.14 * n * n return;</pre>		

ثانياً – الكائنات Object

رقم النشاط	عنوان النشاط	وصف النشاط	توقيت التنفيذ	طريقة التنفيذ
1	أكتب الكود البرمجي	اكتب الكود الخاص بأوبجكت متفرع من الكلاس fan (لها متغيرين price ، model) مع وضع قيم من عندك	خلال المحاضرة	تعليم تعاوني مجموعات صغيرة
2	إستنتج نتيجة تنفيذ كود برمجي	استنتج نتيجة تنفيذ الكود التالي pro_1 = Processor ("Core i5, 1500)		
3	اكتب برنامج بايثون	اكتب الكود الخاص بأوبجكت متفرع من كائن يقوم بحساب مساحة مساحة مستطيل		
4	إستخرج الخطأ البرمجي	استخرج الخطأ البرمجي في الكود التالي user_circle = C_area. Circle		

تكاليف برمجية

قم بتنفيذ التكاليف التالية، ثم تواصل مع أستاذ المادة لتسليمها ومناقشتها في المحاضرات التالية:

م	التكليف
1	اكتب الكود الخاص بكلاس إفتراضي لموبايل ثم إنشئ أوبجت منه وضع فيه قيم من عندك
2	استنتج نتيجة تنفيذ الكود التالي class Area : def square_area (self , x): x = x * 2 return area ;
3	اكتب برنامج بايثون يستكون من كلاس لحساب مساحة مربع ، وأوبجت خاص بإدخال طول الضلع
4	استخرج الخطأ البرمجي في الكود التالي class Laptop def __init__(model, price): self.model = model self.price = price

مصادر إثرائية من بنك المعرفة المصري



Egyptian Knowledge Bank
بنك المعرفة المصري

من خلال المصادر الالكترونية التالية والموجودة ببنك المعرفة المصري والمكتبة الرقمية؛ يمكنك الاستزادة حول المحتوى العلمي للفصل:

م	عنوان المصدر	الرابط
1	دراسة عن تعليم البرمجة الكائنية باستخدام المنصات الالكترونية	https://0810gxsre-1106-y-https-search-mandumah-com.mplbci.ekb.eg/Record/216146
2	دراسة عن تعليم البرمجة الكائنية باستخدام مواقع الويب	https://0810gxsre-1106-y-https-search-mandumah-com.mplbci.ekb.eg/Record/1005161

قاموس المصطلحات الواردة بالفصل

م	المصطلح بالعربية	المصطلح بالإنجليزية	الوصف
1	البرمجة الكائنية	Object Oriented Programing	نمط متقدم من البرمجة يتم فيه يتم تقسيم البرنامج لمجموعة أجزاء (كائنات) ، وكل جزء يحتوي على حزمة من الأكواد ، ويهدف كل كائن لتحقيق إحدى وظائف البرنامج ككل
2	الفئات	Class	حاوية برمجية تضم العديد من الاكواد (اسمها داخل الكلاس property) ، ويمكن وضعها في ملف منفصل داخل المشروع البرمجي
3	الكائنات	Object	نسخة من الكلاس الأصلي ، بمعنى أن المبرمج يقوم بإنشاء كلاس مرة واحدة ، وعند استخدامه يقوم بإنشاء نسخ منه لاستخدامها عدة مرات ، ودون إعادة لكتابة كود الكلاس مرة أخرى

اختبر نفسك

اختر الإجابة الصحيحة من بين البدائل الموجودة:

1		م
طريقة برمجية تقسم البرنامج لأجزاء وحزم		العبارة
البرمجة الكائنية	أ	البدائل
الشرط البرمجي	ب	
الادخال البرمجي	ج	
المكتبات البرمجية	د	
2		م
أي من التالي من استخدامات البرمجة الكائنية		العبارة
تكرار الكود عدد من المرات	أ	البدائل
اختبار قيمة متغير	ب	
عدم تكرار الكود	ج	
إضافة أدوات الادخال والايخارج	د	
3		م
أي من التالي ليس من خصائص الفئات Class		العبارة
حاوية برمجية	أ	البدائل
تضم متغيرات واكواد	ب	
تؤدي وظيفة في البرنامج	ج	
متفرعة من الدوال	د	

4		م
أي من التالي ليس من خصائص الكائنات البرمجية Objects		العبارة
تتفرع من Class	أ	البدائل
تضم فئات Class	ب	
حاوية برمجية	ج	
تضم دوال برمجية	د	
5		م
الفرق بين الفئة Class والكائن Object يكمن في		العبارة
الفئة تتفرع منها كائنات	أ	البدائل
الكائن يتفرع منه فئات	ب	
الكائن أشمل من الفئة	ج	
الكائن يكتب في أول الكود	د	
6		م
أي من التالي أول كود في انشاء Class بالطريقة المبسطة		العبارة
a = class a	أ	البدائل
class a = class	ب	
a.class	ج	
class a:	د	

7		م
الكود المميز للكلاس المتقدم هو		العبارة
Class	أ	البدائل
Object	ب	
parameter	ج	
def __init__	د	

مفتاح الإجابة

الإجابة الصحيحة	العبارة	م
البرمجة الكائنية	طريقة برمجية تقسم البرنامج لأجزاء وحزم	1
عدم تكرار الكود	أي من التالي من استخدامات البرمجة الكائنية	2
متفرعة من الدوال	أي من التالي ليس من خصائص الفئات Class	3
تضم فئات Class	أي من التالي ليس من خصائص الكائنات البرمجية Objects	4
الفئة تتفرع منها كائنات	الفرق بين الفئة Class والكائن Object يكمن في	5
class a:	أي من التالي أول كود في انشاء Class بالطريقة المبسطة	6
def __init__	الكود المميز للكلاس المتقدم هو	7

الباب الرابع

برمجة واجهات GUI

الفصل الأول

مقدمة في الواجهات الرسومية

الواجهات الرسومية GUI

مفهوم الواجهات الرسومية GUI

مفهوم واجهات Console / CLI

التفوق بين واجهات GUI و CLI

مفهوم المكتبات البرمجية

مفهوم مكتبة tkinter

العناصر الرسومية (widgets) الأساسية

خصائص الأنواع (Properties)

النوافذ البرمجية Forms

كود استيراد مكتبة tkinter

كود إنشاء نافذة

كود نهاية العمليات و غلق النافذة

كود إضافة نص في شريط العنوان

كود تحديد حجم نافذة ومكان ظهورها في الشاشة

كود تحديد لون نافذة

مقدمة الفصل

أولاً - الهدف العام:

إكساب طلاب تكنولوجيا التعليم الرقمي المفاهيم والمهارات الأساسية الخاصة بالعمليات على الواجهات الرسومية ونوافذ التشغيل.

ثانياً - الأهداف التعليمية:

بعد الانتهاء من هذا الفصل /الدرس، يتوقع من الطالب أن يكون قادراً على أن:

1. يذكر مفهوم الواجهات الرسومية GUI
2. يذكر مفهوم واجهات Console / CLI
3. يفرق بين واجهات GUI و CLI
4. يذكر مفهوم المكتبات البرمجية
5. يوضح ماهية مكتبة tkinter
6. يبين العناصر الرسومية (widgets) الأساسية
7. يشرح خصائص الأدوات (Properties)
8. يكتب كود استيراد مكتبة tkinter
9. يكتب كود انشاء نافذة
10. يكتب كود نهاية العمليات وغلق النافذة
11. يكتب كود إضافة نص في شريط العنوان
12. يكتب كود تحديد حجم نافذة ومكان ظهورها في الشاشة
13. يكتب كود تحديد لون نافذة

الواجهات الرسومية GUI

مفهوم الواجهات الرسومية GUI

The image shows a graphical user interface window titled "tk #2". It features a form with four text input fields. The first field is labeled "Last Name" and contains the text "Klein". The second field is labeled "First Name" and contains "Bernd". The third field is labeled "Job" and contains "Trainer". The fourth field is labeled "Country" and contains "Germany". Below the input fields, there are two buttons: "Show" and "Quit". The window has a standard title bar with a dropdown arrow on the left and minus, plus, and close buttons on the right.

- هي عناصر رسومية تمكن المستخدم من التفاعل مع البرنامج من خلال عناصر تحكم مرئية ، مثل النوافذ (Forms) والأزرار (Buttons)
- يتفاعل المستخدم مع البرنامج من خلال هذه الأدوات ، حيث يستخدم الماوس في الضغط على الأزرار ، ولوحة المفاتيح في كتابة بيانات الإدخال
- مصطلح GUI : هو إختصار للجملة Graphical User Interface

مفهوم واجهات Console / CLI

```
C:\Users\Danielle\Desktop\python> pipenv shell
Creating a virtualenv for this project...
Pipfile: C:\Users\Danielle\Desktop\python\Pipfile
Using C:/python/python.exe (3.7.5) to create virtualenv...
[== ] Creating virtual environment...created virtual environment CPython3.7.5.final.0-64 in
creator CPython3Windows(dest=C:\Users\Danielle\.virtualenvs\python-10dCQzS1, clear=False,
seeder FromAppData(download=False, pip=bundle, setuptools=bundle, wheel=bundle, via=copy,
\AppData\Local\pypa\virtualenv)
added seed packages: pip==20.2.2, setuptools==49.6.0, wheel==0.35.1
activators BashActivator,BatchActivator,FishActivator,PowerShellActivator,PythonActivator,
Successfully created virtual environment!
Virtualenv location: C:\Users\Danielle\.virtualenvs\python-10dCQzS1
Creating a Pipfile for this project...
Launching subshell in virtual environment...
Microsoft Windows [Version 10.0.19041.450]
(c) 2020 Microsoft Corporation. All rights reserved.

(python-10dCQzS1) C:\Users\Danielle\Desktop\python>
```

- هي شاشات (غالباً ذات خلفية سوداء) تعتمد على النص في التعامل مع البرنامج
- حيث يتم كتابة الأوامر كاملة بإستخدام لوحة المفاتيح وبدون إستخدام الماوس أو اللمس (في تطبيقات الموبايل)
- ويتم إخراج البيانات والنتائج للمستخدم على شكل رسائل نصية
- يشار لواجهات (Console) أيضاً بالمصطلح (CLI) وهي إختصار لجملة Command line interface

الفرق بين واجهات GUI و CLI



- يتوفر في كل لغات البرمجة الحديثة واجهتان للبرمجة
- الأولى خاصة ب Console Application وهي غالبا ما يبدأ بها دارس البرمجة في التعامل مع لغة البرمجة ، فعن طريقها يتعلم كتابة المتغيرات والدوال والحلقات والشرط والبرمجة الكائنية
- وبعد أن يتقدم دارس البرمجة في تعلم اللغة ينتقل لتعلم إنتاج البرامج ذات الواجهة الرسومية

المكتبات البرمجية

- لغة بايثون لغة مفتوحة المصدر open source ، بالتالي يسمح لأي مبرمج تعديل وتطوير اللغة بما يتناسب مع المستجدات البرمجية
- لذلك ظهرت العديد من المكتبات التي تسمح لمبرمج البايثون بإنتاج برامج واجهات رسومية والتي انتجتها جهات مختلفة
- مصطلح المكتبة في البرمجة هو ملف يتم تحميله ودمج بمترجم اللغة وبيئة التطوير ، به أكواد إذا استخدمها المبرمج سيتمكن من بناء نوعية محددة من التطبيقات أكثر تقدماً، مثل مكتبات واجهات رسومية

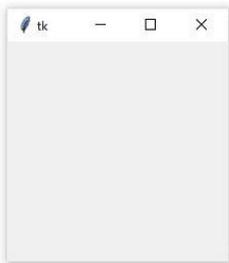
مكتبة tkinter:

- هي مكتبة أكواد واجهات رسومية مجانية ، وهي مدمجة بشكل افتراضي في مترجم بايثون الرسمي ، توفر للمبرمج إمكانية تصميم واجهة رسومية جيدة ، ولاتحتاج لتحميل أي ملفات خارجية. (هي موضوع دراستنا في المحاضرات).
- يتوفر في بايثون أيضاً مكتبة kivy هي مكتبة أكواد واجهات رسومية ، تحتاج لتحميلها من مصادرها ، هي أشمل من مكتبة tkinter لأنها قابلة للعمل على نظم تشغيل أندرويد وأيفون وبها أوامر اللمس ، وبالتالي هي مناسبة جداً لتصميم الألعاب الإلكترونية على الأجهزة المحمولة

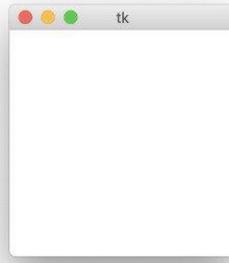
العناصر الرسومية (widgets) الأساسية

يتوفر في مكتبة tkinter العديد من العناصر الرسومية ، أشهرها ما يلي:

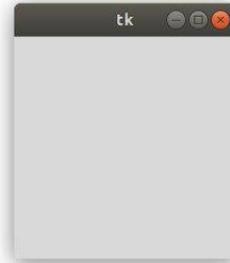
أولاً – النوافذ Frame



(a) Windows



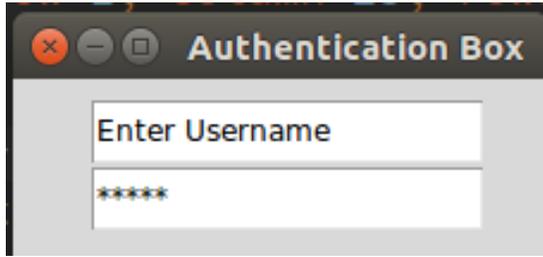
(b) macOS



(c) Ubuntu

- هي النافذة التي تحتوي على جميع الأدوات widgets الأخرى
- يتم فيها ترتيب الأدوات
- لابد أن يحتوي البرنامج على Frame واحد على الأقل

ثانياً – أداة الإدخال Entry



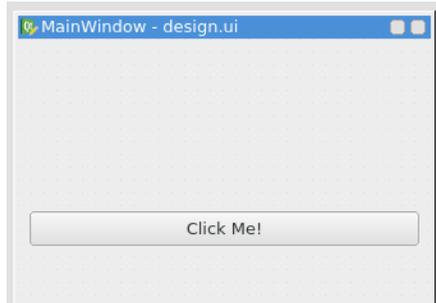
- أداة إدخال رئيسية في أي لغة برمجة
- عبارة عن مربع نص الذي يستخدم في استقبال البيانات من المستخدم
- توجد أدوات إدخال أخرى في لغات البرمجة والبايثون مثل القوائم المنسدلة وغيرها.

ثالثاً – أداة الإخراج Label



- أداة اخراج رئيسية في أي لغة برمجة
- عبارة عن مربع عرض يستخدم في عرض مخرجات البرنامج مثل: تعليمات الاستخدام ، رسائل الترحيب ، نتائج تنفيذ العمليات الحسابية والمنطقية ، عرض الصور
- لا يستطيع المستخدم التعديل عليها حتى لا يفسد المخرجات
- توجد أدوات إدخال أخرى في لغات البرمجة والبايثون مثل نوافذ الرسائل message box.

رابعاً – أداة التنفيذ Button



- أداة رئيسية في أي لغة برمجة ، ولابد من وجودها في أي مشروع برمجي
- هو الزر الذي يقوم المستخدم بالضغط عليه حتى يتم تنفيذ أوامر البرنامج

- يحتوي على الحدث البرمجي event الذي يحول المدخلات (من أدوات الإدخال مثل Entry) إلى مخرجات (تظهر في أدوات الإخراج مثل Label) مروراً بالعمليات الحسابية والمنطقية

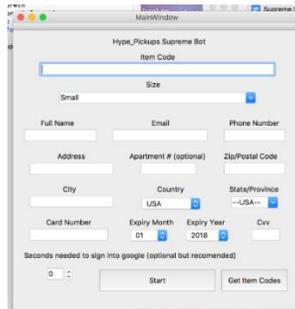
خصائص الأدوات (Properties)

لكل أداة في الواجهات الرسومية طبيعتها الخاصة ووظيفتها وبالتالي لديها خصائص مميزة عن غيرها ، إلا أن جميع الأدوات تشترك معاً في الخصائص التالية:

أولاً - الخصائص الإجبارية:

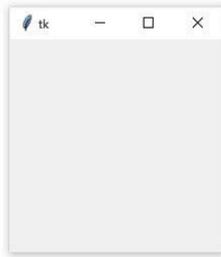
- الخصائص التي يجب ضبطها في الكود البرمجي لكل الأدوات الرسومية
 - هي اجبارية لا يتم انشاء الاداوات بدونها
 - أشهرها اسم الأداة والنافذة التابعة لها
- فيما يلي عرضها بالتفصيل:

1 - اسم الأداة Name

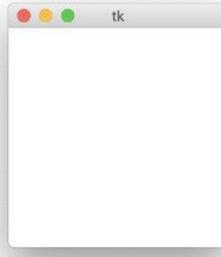


- الاسم البرمجي الذي يعطيه المبرمج للأداة
- يستخدم في العمليات البرمجية ، وفي تمييز الأداة عن الأدوات الأخرى.

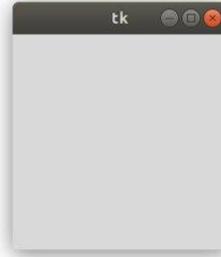
2 – النافذة التابعة frame/tk



(a) Windows



(b) macOS



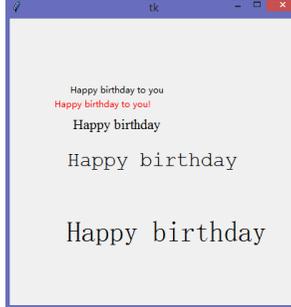
(c) Ubuntu

- النافذة التي ستظهر بها الأداة
- لابد للبرنامج أن يحتوي على نافذة frame واحدة على الأقل
- لابد أن تتبع كل أداة (widget) نافذة وحيدة .

ثانياً – الخصائص الاختيارية:

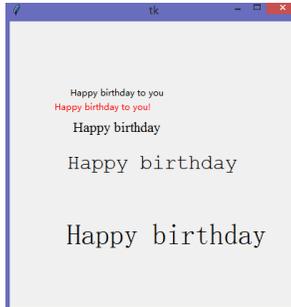
- هي خصائص تنسيقية تحسن شكل الأداة
- دم ضبطها لا يحدث أي مشكلة لأن بايثون سيضع بنفسه الخصائص الافتراضية لتلك الاداة
- أشهرها ما يتعلق بنوع وحجم الخط للنص الظاهر على الأداة ، وحجم الأداة ، وألوان الأداة
- فيما يلي عرضها بالتفصيل :

1 - نوع الخط Font



- يتيح بايثون للمبرمج إختبار نوع الخط الذي يريده على الأدوات التي تظهر بها النصوص
- يمكن للمبرمج عدم تحديد نوع الخط font ، فيقوم بايثون بإختيار نوع الخط الافتراضي لديه
- يمكن للمبرمج معرفة أسماء انواع الخط من برنامج الورد

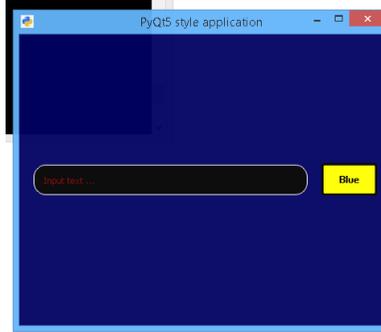
2 - حجم الخط Font



- يتيح بايثون للمبرمج إختبار حجم الخط الذي يريده على الأدوات التي تظهر بها النصوص

- يمكن للمبرمج عدم تحديد حجم محدد ، فيقوم بايثون بتحديد الحجم الافتراضي لديه

3 – لون الخلفية (bg) background



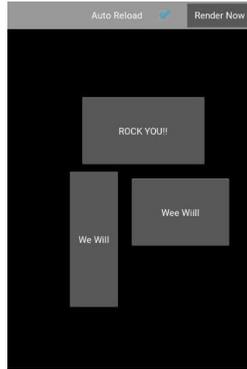
- لون خلفية الأداة ، حيث يتيح بايثون إظهار خلفية ملونة لجميع الأدوات (أحمر ، أخضر ، إلخ ...)
- يمكن للمبرمج عدم تحديد أي لون للخلفية فيقوم بايثون بوضع اللون الافتراضي للخلفية (الرمادي الفاتح للنافذة – الأبيض لمربع النص)
- يتم كتابتها باللغة الإنجليزية الطبيعية (red , green , blue)

4 – لون النص (fg) foreground



- لون النص المكتوب على الأداة ، حيث يتيح بايثون إظهار نصوص ملونة لجميع الأدوات (أحمر ، أخضر ، إلخ ...)
- يمكن للمبرمج عدم تحديد أي لون للخلفية فيقوم بايثون بوضع اللون الافتراضي للنص وهو اللون الأسود
- يتم كتابتها باللغة الإنجليزية الطبيعية (red , green , blue)

5 - حجم الأداة / geometry / width / size



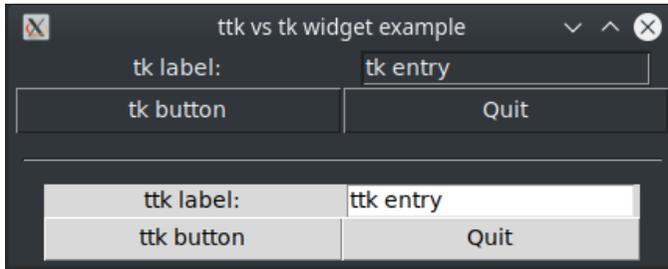
- يتيح بايثون للمبرمج تحديد الحجم الذي يريده للأدوات
- يمكن للمبرمج عدم تحديد أي حجم ، لتظهر الأدوات بحجمها الافتراضي

ثالثاً - الخصائص التنظيمية:

- هي خصائص تنظيمية يجب أن يضعها المبرمج في حساباته عند إنشاء الأداة

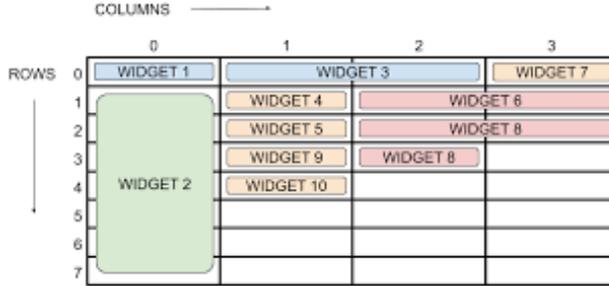
- في المشاريع البسيطة (التي تحتوي نافذة وعدة أدوات قليلة) يمكن ترك تلك الخصائص للوضع الافتراضي في بايثون
 - أما في المشاريع المتوسطة والكبيرة فلا بد للمبرمج أن يقوم بضبطها ليحصل على أفضل أداء للمشروع البرمجي
 - أشهرها النص الظاهر على الأداة ، ومكان الأداة على النافذة بشكل دقيق
- فيما يلي عرضها بالتفصيل:

1 - النص الظاهر على الأداة Title / Text



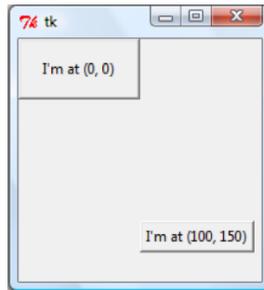
- هو النص الذي يظهر على الأداة للمستخدم
- هو اختياري في أدوات مثل : عنوان النافذة `frame` ، تعليمات الاستخدام على الاداة `label` ، النص المكتوب على مربع النص
- هو اجباري في أدوات مثل الزر `button`
- بدون هذا النص لن يستطيع المستخدم استخدام الأدوات بشكل صحيح

2 - مكان الأداة في النافذة / خاصية grid



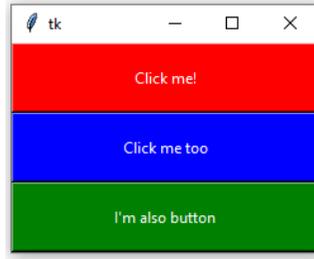
- هي طريقة من طرق تنظيم الأدوات الرسومية في النافذة (grid ، place ، pack ، place)
- فيها يقوم بايثون بتقسيم النافذة frame إلى شبكة تتكون من صفوف وأعمدة
- يتم ترقيم الصفوف من صفر من أعلى ويزيد كما اتجهنا للأسفل
- يتم ترقيم الأعمدة من اليسار بالرقم صفر ويزيد كلما اتجهنا لليمين
- بالتالي يقوم المبرمج بتحديد مكان الأداة عن طريق تحديد رقم الصف ورقم العمود الذي يريد أن تظهر الأداة فيه

3 - مكان الأداة في النافذة / خاصية place



- هي طريقة من طرق تنظيم الأدوات الرسومية في النافذة (grid ، pack ، place)
- فيها يقوم بايثون بتقسيم النافذة إلى احدثيين ، هما المحور X والمحور y
- يتم بدء ترقيم محور y من الأعلى صفر ويزيد كلما اتجهنا للأسفل
- يتم بدء ترقيم محور X بصفر من اليسار ، ويزيد كلما إتجهنا لليمين ،
- بالتالي يقوم المبرمج بتحديد مكان الأداة عن طريق تحديد قيمه في محوري X و y

4 - مكان الأداة في النافذة / خاصية pack



- هي طريقة من طرق تنظيم الأدوات الرسومية في النافذة (grid ، pack ، place)
- فيها يقوم بايثون بوضع الأدوات مرتبة تلقائيا (كل أداة في صف ، ويبدأ من الأعلى للأسفل) وتكون في منتصف الشاشة
- يتم تحديد ترتيب الأدوات في الصفوف وفق ترتيبها في الكود البرمجي
- من الطرق الشهيرة في نظام pack ضبط أماكن الأدوات (top ، left ، right ، bottm)

النوافذ البرمجية / Frame

كود استيراد مكتبة tkinter

- عند البدء في استخدام مكتبة tkinter لابد من استدعائها أولاً تمهيداً للتعامل مع أكوادها وأدواتها الرسومية
- يكتب الكود كما هو بنفس الطريقة وفي جميع البرامج وبدون تغيير
- يتم كتابته مرة واحدة فقط (أول سطر في البرنامج) ولا يتم كتابته مرة
- هو كود إجباري لابد من وجوده عند الرغبة في التعامل مع المكتبة

قاعدة كتابة كود استيراد مكتبة tkinter

```
from tkinter import *
```

كود انشاء نافذة

كود إجباري - يقوم بانشاء النافذة وتسميتها ، على النحو التالي:

قاعدة كتابة كود انشاء نافذة

```
Form_name = Tk( )
```

مثال على كود انشاء نافذة

<code>x = Tk()</code>	انشاء نافذة - يتم التعامل معها بالاسم <code>x</code>
<code>sell_form = Tk()</code>	انشاء نافذة - يتم التعامل معها بالاسم <code>sell_form</code>

كود نهاية العمليات وغلق النافذة

- كود اجباري
- يجعل البايثون يفهم أن هذه هي نهاية النافذة ، وأن جميع الأكواد الخاصة بالبرنامج ستكون بين كود الاستيراد import وكود النهاية mainloop
- يكتب الكود كما هو بنفس الطريقة وفي جميع البرامج وبدون تغيير
- يتم كتابته مرة واحدة فقط (آخر سطر في البرنامج) بعد أن يتم كتابة جميع أكواد الأدوات والعمليات الأخرى

قاعدة كتابة كود حدود نافذة	
Form_name = mainloop ()	
مثال على كود حدود نافذة	
x = mainloop ()	انهاء وغلق اكواد نافذة x
sell_form = mainloop ()	انهاء وغلق اكواد نافذة sell_form

كود إضافة نص في شريط العنوان

- كود اختياري - يكتب في سطر منفصل
- يتيح إضافة نص يظهر في شريط عنوان النافذة

قاعدة كتابة كود نص يظهر في شريط عنوان نافذة	
Form_name.title = (" text ")	
مثال على كود نص يظهر في شريط عنوان نافذة	
go.title (" welcome ")	إضافة النص (welcome) للنافذة (go)

كود تحديد حجم نافذة ومكان ظهورها في الشاشة

- كود اختياري - يكتب في سطر منفصل
- يتيح للمستخدم ضبط حجم محدد للنافذة ، مع العلم أن بايثون يقوم بضبط النوافذ حسب حجم شاشة الجهاز سواء موبايل او كمبيوتر
- يتيح نفس الكود تحديد مكان ظهورها على الشاشة في احداثيات X و Y ، بالإضافة إلى حجم النافذة

قاعدة كتابة كود نص تحديد حجم نافذة	
Form_name.geometry = (" length x width ")	
مثال على كود نص يظهر في شريط عنوان نافذة	
main. geometry (" 800 x 600 ")	ضبط حجم النافذة (main) لتصبح (800 x 600)
قاعدة كتابة كود نص يحدد حجم + احداثيات ظهور نافذة	
Form_name.geometry = (" length x width ")	
مثال على كود نص يحدد احداثيات ظهور نافذة	
main. geometry ("800x 600+150+150")	ضبط حجم النافذة (main) لتصبح (800 x 600) وتظهر في احداثي X يساوي 150 ، ولا يساوي 150

كود تحديد لون نافذة

- كود اختياري - يكتب في سطر منفصل
- يتيح للمستخدم تلوين النافذة بلون غير الافتراضي (الرمادي الفاتح)

قاعدة كتابة كود نص تلوين نافذة	
Form_name.configure (background = color)	
مثال على كود نص تلوين نافذة	
sys.configure (background = black)	ضبط لون النافذة (sys) لتصبح (باللون الأسود (black

أنشطة تعليمية على النوافذ البرمجية Frame

رقم النشاط	عنوان النشاط	وصف النشاط	توقيت التنفيذ	طريقة التنفيذ
1	أكتب الكود البرمجي	اكتب الكود الخاص بإنشاء نافذة بالإسم month	خلال المحاضرة	تعليم تعاوني مجموعات صغيرة
2	إستنتج نتيجة تنفيذ كود برمجي	استنتج نتيجة تنفيذ الكود التالي from tkinter import *		
3	اكتب برنامج بايثون	اكتب الكود الذي يجعل لون النافذة go باللون الأخضر		
4	إستخرج الخطأ البرمجي	استخرج الخطأ البرمجي في الكود التالي day . Tk		

تكليفات برمجية

قم بتنفيذ التكليفات التالية، ثم تواصل مع أستاذ المادة لتسليمها ومناقشتها في المحاضرات التالية:

م	التكليف
1	اكتب الكود الخاص بإنشاء نافذة بالإسم car
2	<code>from tkinter import *</code>
3	اكتب الكود الذي يجعل حجم النافذة <code>say = 100 x 200</code> ومكان ظهورها للمستخدم في الابعاد 100 ، 300
4	<code>s = mainloop</code>

مصادر إثنائية من بنك المعرفة المصري



Egyptian Knowledge Bank
بنك المعرفة المصري

من خلال المصادر الالكترونية التالية والموجودة ببنك المعرفة المصري والمكتبة الرقمية؛ يمكنك الاستزادة حول المحتوى العلمي للفصل:

م	عنوان المصدر	الرابط
1	كتاب عن البرمجة المرئية	https://0811nxsk9-1106-y-https-ekbedu-ethraadl-com.mplbci.ekb.eg/Record/62513
2	كتاب عن البرمجة مفتوحة المصدر	https://0811nxsk9-1106-y-https-ekbedu-ethraadl-com.mplbci.ekb.eg/Record/51416

قاموس المصطلحات الواردة بالفصل

م	المصطلح بالعربية	المصطلح بالإنجليزية	الوصف
1	الواجهات الرسومية	Graphical User Interface	هي عناصر رسومية تمكن المستخدم من التفاعل مع البرنامج من خلال عناصر تحكم مرئية ، مثل النوافذ (Forms) والأزرار (Buttons)
2	واجهات كونسول	Command line interface	هي شاشات (غالبا ذات خلفية سوداء) تعتمد على النص في التعامل مع البرنامج ، حيث يتم كتابة الأوامر كاملة باستخدام لوحة المفاتيح وبدون استخدام الماوس أو اللمس (في تطبيقات الموبايل)
3	المكتبات البرمجية	Library	ملف يتم تحميله ويدمج بلغة البرمجة ، به أكواد إذا استخدمها المبرمج سيتمكن من بناء نوعية محددة من التطبيقات
4	تي كي انتر	Tkinter	مكتبة الواجهات الرسومية في بايثون

اختبر نفسك

اختر الإجابة الصحيحة من بين البدائل الموجودة:

1		م
تحتوي عناصر رسومية تمكن المستخدم من التفاعل مع البرنامج		العبارة
GUI	أ	البدائل
CLI	ب	
Android	ج	
IOS	د	
2		م
واجهات لغات برمجة يتم فيها كتابة الاكواد بشكل تسلسلي		العبارة
GUI	أ	البدائل
DOS	ب	
Project	ج	
CLI	د	
3		م
تتميز واجهات GUI بأنها		العبارة
تعتمد على كتابة الكود المتوالي	أ	البدائل
ذات خلفية سوداء	ب	
تعتمد على الأدوات الرسومية	ج	
برمجة متقدمة	د	

4		م
ملحق بلغة البرمجة يمكنها من انتاج مشاريع برمجية		العبارة
ملف تنفيذي	أ	البدائل
مكتبة برمجية	ب	
ملف مفسر	ج	
مكتبة تطبيقات	د	
5		م
مكتبة الواجهات الرسومية في بايثون		العبارة
Tkinter	أ	البدائل
GUI	ب	
PyDroid	ج	
Windows	د	
6		م
عنصر رسومي يحتوي على جميع العناصر الرسومية الأخرى في المشروع البرمجي		العبارة
Entry	أ	البدائل
Button	ب	
Label	ج	
Frame	د	

7		م
أي من التالي من الخصائص الإجبارية للأدوات الرسومية البرمجية		العبارة
background	أ	البدائل
Name	ب	
size	ج	
Title	د	
8		م
أي من التالي مثال على كود استيراد مكتبة tkinter		العبارة
from tkinter import *	أ	البدائل
z= mainloop()	ب	
a. size("800x 600")	ج	
c. configure (background = black)	د	
9		م
أي من التالي مثال على كود انشاء نافذة		العبارة
x = Tk()	أ	البدائل
gh = Entry(states1)	ب	
alc.place (x = 20 , y = 20)	ج	
z.grid (column = 3 , row = 3)	د	

10		م
أي من التالي مثال على كود نهاية العمليات وغلق النافذة		العبارة
a. grid("800x 600")	أ	البدائل
from tkinter import*	ب	
s= mainloop()	ج	
d.grid (column = 1 , row = 2)	د	
11		م
أي من التالي مثال على كود إضافة نص في شريط عنوان نافذة		العبارة
f.title = (" insert ")	أ	البدائل
fees. configure (background = white)	ب	
b.grid (column = 4 , row = 2)	ج	
userq.place (x = 99 , y = 45)	د	
12		م
أي من التالي مثال على كود تحديد حجم نافذة الشاشة		العبارة
cv.place (x = 50 , y = 100)	أ	البدائل
a. grid("600 2")	ب	
g.grid (column = 5 , row = 1)	ج	
k. geometry("100 X 200")	د	

13		م
أي من التالي مثال على كود تحديد لون نافذة		العبارة
all22. configure (background = blue)	أ	البدائل
from tkinter import*	ب	
as.title = (" login ")	ج	
h = Entry(tree)	د	

مفتاح الإجابة

م	العبرة	الإجابة الصحيحة
1	تحتوي عناصر رسومية تمكن المستخدم من التفاعل مع البرنامج	GUI
2	واجهات لغات برمجة يتم فيها كتابة الاكواد بشكل تسلسلي	CLI
3	تتميز واجهات GUI بأنها	تعتمد على الأدوات الرسومية
4	ملحق بلغة البرمجة يمكنها من انتاج مشاريع برمجية	مكتبة برمجية
5	مكتبة الواجهات الرسومية في بايثون	Tkinter
6	عنصر رسومي يحتوي على جميع العناصر الرسومية الأخرى في المشروع البرمجي	Frame
7	أي من التالي من الخصائص الإجبارية للأدوات الرسومية البرمجية	Name
8	أي من التالي مثال على كود استيراد مكتبة tkinter	from tkinter import*
9	أي من التالي مثال على كود انشاء نافذة	x = Tk()

الإجابة الصحيحة	العبرة	م
s= mainloop()	أي من التالي مثال على كود نهاية العمليات وغلق النافذة	10
f.title = (" insert ")	أي من التالي مثال على كود إضافة نص في شريط عنوان نافذة	11
k. geometry("100 X 200")	أي من التالي مثال على كود تحديد حجم نافذة الشاشة	12
all22. configure (background = blue)	أي من التالي مثال على كود تحديد لون نافذة	13

الفصل الثاني

أدوات الإدخال والإخراج البرمجي

اداة الإدخال البرمجي / مربع النص

كود إنشاء مربع نص

كود تحديد عرض مربع نص بدل العرض الافتراضي

كود تحديد لون الخلفية ولون النص

كود تحديد مكان مربع نص بطويقة الشبكة

كود تحديد مكان مربع نص بطويقة الاحداثيات

كود تحديد مكان مربع عرض بطويقة الحزم

أداة الإخراج البرمجي / مربع العرض

كود انشاء مربع عرض

كود تحديد عرض مربع العرض بدل العرض الافتراضي

كود تحديد لون الخلفية ولون النص

كود تحديد مكان مربع عرض بطويقة الشبكة

كود تحديد مكان مربع عرض بطويقة الاحداثيات

كود تحديد مكان مربع عرض بطويقة الحزم

مقدمة الفصل

أولاً - الهدف العام:

إكساب طلاب تكنولوجيا التعليم الرقمي المفاهيم والمهارات الأساسية الخاصة بالعمليات على أدوات الإدخال والإخراج للواجهات الرسومية البرمجية.

ثانياً - الأهداف التعليمية:

بعد الانتهاء من هذا الفصل/الدرس، يتوقع من الطالب أن يكون قادراً على أن:

1. يكتب كود إنشاء مربع نص
2. يكتب كود تحديد عرض مربع نص بدل العرض الافتراضي
3. يكتب كود تحديد لون الخلفية ولون النص
4. يكتب كود تحديد مكان مربع نص بطريقة الشبكة
5. يكتب كود تحديد مكان مربع نص بطريقة الاحداثيات
6. يكتب كود تحديد مكان مربع نص بطريقة الحزم
7. يكتب كود انشاء مربع عرض
8. يكتب كود تحديد عرض مربع عرض بدل العرض الافتراضي
9. يكتب كود تحديد لون الخلفية ولون النص
10. يكتب كود تحديد مكان مربع عرض بطريقة الشبكة
11. يكتب كود تحديد مكان مربع عرض بطريقة الاحداثيات
12. يكتب كود تحديد مكان مربع عرض بطريقة الحزم

اداة الإدخال / مربع النص Entry

كود إنشاء مربع نص

- كود إجباري
- يقوم بإنشاء مربع النص وتسميته
- يكتب في سطر كود منفصل
- يمكن ان يكتب مع أكواد أخرى مثل (تحديد عرض المربع ، تحديد لون الخلفية ولون النص - سيتم شرحهم فيما بعد)
- إذا لم يتم كتابته مع تلك الأكواد سيقوم بإيثون بضبط العرض والألوان حسب الاعدادات التلقائية لمربعات النص

قاعدة كتابة كود انشاء مربع نص	
entry_name = Entry (frame_name)	
مثال على كود انشاء مربع نص	
x = Entry(mains)	انشاء مربع نص - يتم التعامل معه بالاسم x يتبع النافذة mains
user_name = Entry(sign_frame)	انشاء مربع نص - يتم التعامل معه بالاسم user_name يتبع النافذة sign_frame

كود تحديد عرض مربع نص بدل العرض الافتراضي

- كود تنظيمي
- يقوم بضبط عرض مربع النص وفق قيمة يدخلها المبرمج ، في حالة الرغبة في تصغير أو تكبير الحجم الافتراضي لمربع النص
- يفضل أن يتم كتابته مع كود انشاء مربع العرض في سطر واحد

قاعدة كتابة كود (إنشاء + عرض) مربع نص	
<code>entry_name = Entry (frame_name , width = value)</code>	
مثال على كود (إنشاء + عرض) مربع نص	
<code>x = Entry(mains , width = 25)</code>	انشاء أداة x ، تتبع النافذة mains ، عرضها 25
<code>user_name = Entry(sign_frame , width = 10)</code>	انشاء أداة user_name ، تتبع النافذة sign_frame ، عرضها 10

كود تحديد لون الخلفية ولون النص

- كود تنظيمي
- يقوم بضبط لون مربع النص ولون النص نفسه
- يفضل أن يتم كتابته مع كود انشاء مربع العرض في سطر واحد
- يمكن ان يتم كتابة الثلاث اكواد (الانشاء + العرض + الألوان) في سطر واحد

قاعدة كتابة كود (إنشاء + عرض + ألوان) مربع نص	
entry_name = Entry (frame_name , width = value , bg = " color " , fg = " color ")	
مثال على كود (إنشاء + عرض + ألوان) مربع نص	
x = Entry (mains , width = 25 , bg = " black " , fg = "white")	إنشاء أداة x ، تتبع النافذة mains ، عرضها 25 ، لون الأداة أسود ، لون النص الظاهر عليها أبيض
user_name = Entry(sign_frame , width = , bg = " green " , fg = "red")	إنشاء أداة user_name ، تتبع النافذة sign_frame ، لون الأداة أخضر ، لون النص الظاهر عليها أحمر

كود تحديد مكان مربع نص بطريقة الشبكة

- كود تنظيمي
- يقوم بضبط مكان ظهور الأداة ، بناء على صفوف مرقمة من 0 وتزيد كلما هبط للأسفل ، وأعمدة مرقمة من 0 وتزيد كلما اتجهت لليسا

قاعدة كتابة كود ضبط مكان مربع نص بطريقة الشبكة Grid	
entry_name.grid (column = value , row = value)	
مثال على كود ضبط مكان مربع نص بطريقة الشبكة Grid	
x.grid (column = 0 , row = 0)	ضبط مكان أداة x في شبكة النافذة ، الصف الأول من الأعلى ، العمود الأول من اليسار
ff.grid (column = 2 , row = 3)	ضبط مكان أداة ff في شبكة النافذة ، الصف الثالث من الأعلى ، العمود الرابع من اليسار

كود تحديد مكان مربع نص بطريقة الأحداثيات

- كود تنظيمي
- يقوم بضبط مكان ظهور الأداة ، بناء على إحداثيات x مرقمة من 0 وتزيد كلما اتجهت لليسار ، y مرقمة من صفر وتزيد كلما اتجهت للأسفل

قاعدة كتابة كود ضبط مكان مربع نص بطريقة الأحداثيات Place	
entry_name.place (x = value , y = value)	
مثال على كود ضبط مكان مربع نص بطريقة الأحداثيات Place	
x.place (x = 100 , y = 150)	ضبط مكان أداة x في أحداثيات النافذة ، موقعها على النقطة x يساوي 100 ، موقعها على النقطة y يساوي 150
ff.place (x = 200 , y = 300)	ضبط مكان أداة ff في أحداثيات النافذة ، موقعها على النقطة x يساوي 200 ، موقعها على النقطة y يساوي 150

كود تحديد مكان مربع نص بطريقة الحزم

- كود تنظيمي
- يقوم بضبط مكان ظهور الأداة ، بناء على تحديد مكان واضح (top ، left ، right ، botton)

قاعدة كتابة كود ضبط مكان مربع نص بطريقة الحزم Pack	
entry_name.pack (side = ' direction ')	
مثال على كود ضبط مكان مربع نص بطريقة الحزم Pack	
x.pack (side = ' top ')	ضبط مكان أداة x في احداثيات النافذة ، موقعها الجانب العلوي
ff.pack (side = ' right ')	ضبط مكان أداة ff في احداثيات النافذة ، موقعها الجانب الايمن

أداة الإخراج / مربع العرض Label

كود إنشاء مربع عرض

- كود إجباري
- يقوم بإنشاء مربع العرض وتسميته
- يكتب في سطر كود منفصل
- يمكن ان يكتب مع أكواد أخرى مثل (تحديد عرض المربع ، تحديد لون الخلفية ولون النص - سيتم شرحهم فيما بعد)
- إذا لم يتم كتابته مع تلك الأكواد سيقوم بإيثون بضبط العرض والألوان حسب الاعدادات التلقائية لمربعات النص

قاعدة كتابة كود انشاء مربع عرض	
label_name = Label (frame_name)	
مثال على كود انشاء مربع عرض	
f = Label (models)	انشاء مربع عرض - يتم التعامل معه بالاسم f يتبع النافذة models
result1 = Label (sign2)	انشاء مربع عرض - يتم التعامل معه بالاسم result1 يتبع النافذة sign2

كود تحديد عرض مربع بدل العرض الافتراضي

- كود تنظيمي
- يقوم بضبط عرض المربع وفق قيمة يدخلها المبرمج ، في حالة الرغبة في تصغير أو تكبير الحجم الافتراضي لمربع العرض
- يفضل أن يتم كتابته مع كود انشاء مربع العرض في سطر واحد

قاعدة كتابة كود (إنشاء + عرض) مربع العرض	
<code>label_name = Label (frame_name , width = value)</code>	
مثال على كود (إنشاء + عرض) مربع العرض	
<code>f = Label (models , width = 20)</code>	انشاء أداة f ، تتبع النافذة models ، عرضها 20
<code>result1 = Label (sign2 , width = 15)</code>	انشاء أداة result1 ، تتبع النافذة sign2 ، عرضها 15

كود تحديد لون الخلفية ولون النص

- كود تنظيمي
- يقوم بضبط لون مربع النص ولون النص نفسه
- يفضل أن يتم كتابته مع كود انشاء المربع في سطر واحد
- يمكن ان يتم كتابة الثلاث اكواد (الانشاء + العرض + الألوان) في سطر واحد

قاعدة كتابة كود (إنشاء + عرض + ألوان) مربع عرض	
<code>label_name = label (frame_name , width = value , bg = " color " , fg = " color ")</code>	

مثال على كود (إنشاء + عرض + ألوان) مربع عرض	
<code>f = Label (models , width = 20 , bg = " black " , fg = "white")</code>	إنشاء أداة <code>f</code> ، تتبع النافذة <code>models</code> ، عرضها <code>20</code> ، لون الأداة <code>أسود</code> ، لون النص الظاهر عليها <code>أبيض</code>
<code>result1 = Label (sign_frame , width = , bg = " green " , fg = "red")</code>	إنشاء أداة <code>result1</code> ، تتبع النافذة <code>sign2</code> ، لون الأداة <code>أخضر</code> ، لون النص الظاهر عليها <code>أحمر</code>

كود تحديد مكان مربع عرض بطريقة الشبكة

- كود تنظيمي
- يقوم بضبط مكان ظهور الأداة ، بناء على صفوف مرقمة من 0 وتزيد كلما هبط للأسفل ، وأعمدة مرقمة من 0 وتزيد كلما اتجهت لليساار

قاعدة كتابة كود ضبط مكان مربع عرض بطريقة الشبكة Grid	
<code>label_name.grid (column = value , row = value)</code>	
مثال على كود ضبط مكان مربع عرض بطريقة الشبكة Grid	
<code>x.grid (column = 0 , row = 0)</code>	ضبط مكان أداة <code>x</code> في شبكة النافذة ، الصف <code>الأول</code> من الأعلى ، العمود <code>الأول</code> من اليسار
<code>ff.grid (column = 2 , row = 3)</code>	ضبط مكان أداة <code>ff</code> في شبكة النافذة ، الصف <code>الثالث</code> من الأعلى ، العمود <code>الرابع</code> من اليسار

كود تحديد مكان مربع عرض بطريقة الأحداثيات

- كود تنظيمي
- يقوم بضبط مكان ظهور الأداة ، بناء على إحداثيات x مرقمة من 0 وتزيد كلما اتجهت لليسار ، y مرقمة من صفر وتزيد كلما اتجهت للأسفل

قاعدة كتابة كود ضبط مكان مربع عرض بطريقة الأحداثيات Place	
label_name.place (x = value , y = value)	
مثال على كود ضبط مكان مربع عرض بطريقة الأحداثيات Place	
x.place (x = 100 , y = 150)	ضبط مكان أداة x في أحداثيات النافذة ، موقعها على النقطة x يساوي 100 ، موقعها على النقطة y يساوي 150
ff.place (x = 200 , y = 300)	ضبط مكان أداة ff في أحداثيات النافذة ، موقعها على النقطة x يساوي 200 ، موقعها على النقطة y يساوي 150

كود تحديد مكان مربع عرض بطريقة الحزم

- كود تنظيمي
- يقوم بضبط مكان ظهور الأداة ، بناء على تحديد مكان واضح (top ، left ، right ، botton)

قاعدة كتابة كود ضبط مكان مربع عرض بطريقة الحزم Pack	
label_name.pack (side = ' direction ')	
مثال على كود ضبط مكان مربع عرض بطريقة الحزم Pack	
x.pack (side = ' top ')	ضبط مكان أداة x في احداثيات النافذة ، موقعها الجانب العلوي
ff.pack (side = ' right ')	ضبط مكان أداة ff في احداثيات النافذة ، موقعها الجانب الايمن

أنشطة تعليمية على أدوات الإدخال والإخراج

أولاً - أداة الإدخال البرمجي Entry

رقم النشاط	عنوان النشاط	وصف النشاط	توقيت التنفيذ	طريقة التنفيذ
1	أكتب الكود البرمجي	اكتب الكود الخاص بإنشاء مربع نص بالإسم value1 والتابع للنافذة syst		
2	إستنتج نتيجة تنفيذ كود برمجي	استنتج نتيجة تنفيذ الكود التالي f_g = Entry (syst , width =13 , bg = "orange" , fg = "white")	خلال المحاضرة	تعليم تعاوني مجموعات صغيرة
3	اكتب برنامج بايثون	اكتب الكود الخاص بإنشاء مربع نص بالإسم value1 والتابع للنافذة syst		

رقم النشاط	عنوان النشاط	وصف النشاط	توقيت التنفيذ	طريقة التنفيذ
		بالعرض 10		
4	إستخرج الخطأ البرمجي	استخرج الخطأ البرمجي في الكود التالي value_1 = textbox()		

ثانياً – أداة الإخراج البرمجي Label

رقم النشاط	عنوان النشاط	وصف النشاط	توقيت التنفيذ	طريقة التنفيذ
1	أكتب الكود البرمجي	اكتب الكود الخاص بتحديد مكان (go_lbl) label بالخاصية grid		
2	إستنتج نتيجة تنفيذ كود برمجي	استنتج نتيجة تنفيذ الكود التالي Lbl1 = Label (syst , text =“Access Denied ” , bg = “orange ” , fg = “white”)	خلال المحاضرة	تعليم تعاوني مجموعات صغيرة
3	اكتب برنامج بايثون	اكتب الكود الخاص بإنشاء label بالإسم txt1 والتابع m_input للنافذة والنص الظاهر عليه (This : is result		
4	إستخرج الخطأ البرمجي	استخرج الخطأ البرمجي في الكود التالي result_1 = Label()		

تكليفات برمجية

قم بتنفيذ التكليفات التالية، ثم تواصل مع أستاذ المادة لتسليمها ومناقشتها في المحاضرات التالية:

أولاً- أداة الإدخال **Entry**:

م	التكليف
1	اكتب الكود الخاص بإنشاء مربع نص بالإسم <code>txt1</code> والتابع للنافذة <code>m_input</code>
2	استنتج نتيجة تنفيذ الكود التالي <code>ghan = Entry (tries , width =44 , bg = "green" , fg = "blue")</code>
3	اكتب الكود الخاص بإنشاء مربع نص بالإسم <code>usern</code> والتابع للنافذة <code>reg</code> بالعرض 30
4	استخرج الخطأ البرمجي في الكود التالي <code>go_txt = grid (1 x 1)</code>

ثانياً - أداة الإخراج Label:

م	التكليف
1	اكتب الكود الخاص بإنشاء label بالإسم usernm والتابع للنافذة reg والنص الظاهر عليه (Insert Your User Name)
2	استنتج نتيجة تنفيذ الكود التالي klo = Label (cv , text ="Access Allowed " , bg = "black " , fg = "grey")
3	اكتب الكود الخاص بإنشاء label بالإسم jol والتابع للنافذة strom والنص الظاهر (Well done) ، بلون نص رمادي غامق ولون خلفية أسود
4	استخرج الخطأ البرمجي في الكود التالي num_6 = grid (x = 200 , row = 300)

مصادر إثنائية من بنك المعرفة المصري



Egyptian Knowledge Bank
بنك المعرفة المصري

من خلال المصادر الإلكترونية التالية والموجودة ببنك المعرفة المصري والمكتبة الرقمية؛ يمكنك الاستزادة حول المحتوى العلمي للفصل:

م	عنوان المصدر	الرابط
1	دراسة عن تنمية مهارات برمجة باستخدام الوسائط المتعددة	https://0810gxsre-1106-y-https-search-mandumah-com.mplbci.ekb.eg/Record/719974
2	دراسة عن التأثيرات التربوية لتدريس البرمجة الرسومية	https://0810gxsjr-1106-y-https-search-mandumah-com.mplbci.ekb.eg/Record/1151752

قاموس المصطلحات الواردة بالفصل

م	المصطلح بالعربية	المصطلح بالإنجليزية	الوصف
1	مربعات النص	Entry	أداة إدخال رئيسية في البرمجة ، تتيح للمستخدم كتابة قيم عددية ونصية وادخالها بالبرنامج
2	مربعات العرض	Label	أداة اخرج رئيسية في البرمجة ، تقوم بإظهار النتائج للمستخدم
3	زر	Button	أداة رئيسية في تنفيذ الاحداث البرمجية ، عند الضغط عليها يقوم البرنامج بتنفيذ أحد الأوامر
4	نافذة	Frame	أداة رئيسية تضم بداخلها جميع أدوات الإدخال والإخراج والاحداث البرمجية

اختبر نفسك

اختر الإجابة الصحيحة من بين البدائل الموجودة:

1		م
كود انشاء مربع نص هو		العبارة
f = Entry(d)	أ	البدائل
u.grid (column = 1 , row = 1)	ب	
f2 = Label (b1)	ج	
c = Button (aj , text = "ok")	د	
2		م
أي من التالي مثال على كود تحديد عرض مربع نص بدل العرض الافتراضي		العبارة
c= mainloop()	أ	البدائل
r = Tk()	ب	
Z44.title = (" cool ")	ج	
u = Entry(h , width = 10)	د	

3		م
أي من التالي مثال على كود تحديد لون الخلفية ولون النص في مربع نص		العبارة
s. geometry("50 X 50")	أ	البدائل
F5. configure (background = green)	ب	
h = Entry(afd , width = 12 , bg = " black " , fg = "white")	ج	
s = Entry(save)	د	
4		م
أي من التالي مثال على كود تحديد مكان مربع نص بطريقة الشبكة		العبارة
from tkinter import*	أ	البدائل
j.grid (column 2 , row = 2)	ب	
tt. place (x = 11 , y = 66)	ج	
v= mainloop()	د	
5		م
أي من التالي مثال على كود تحديد مكان مربع نص بطريقة الاحداثيات		العبارة
t. geometry("60 X 200")	أ	البدائل
aas = Label (frame11)	ب	
btn1 = Button (l12 , text = "inside")	ج	
d. place (x = 150 , y = 15)	د	

6		م
أي من التالي مثال على كود تحديد مكان مربع نص بطريقة الحزم		العبارة
c. pack (side = ' right ')	أ	البدائل
F5. configure (background = red)	ب	
f = Button (h , text = "click" , bg = " black " , fg = "white")	ج	
J = Tk()	د	
7		م
أي من التالي مثال على كود انشاء مربع عرض		العبارة
h= mainloop()	أ	البدائل
d1 = Label (frame3)	ب	
from tkinter import*	ج	
bb. place (x = 20 , y = 20)	د	
8		م
أي من التالي مثال كود تحديد عرض مربع عرض بدل الافتراضي		العبارة
y.grid (column = 0 , row = 1)	أ	البدائل
q = Label (models , width = 20)	ب	
A1.title = (" products ")	ج	
c = Button (g , text = "ok" , bg = " blue " , fg = "black")	د	

9		م
أي من التالي مثال على كود تحديد لون الخلفية ولون النص في مربع عرض		العبارة
u= label(uk , width = 60 , bg = “ green “ , fg = “red”)	أ	البدائل
h. geometry(“10 X 20”)	ب	
b = label (like)	ج	
b7 = Button (flo2 , text = “no”)	د	
10		م
أي من التالي مثال على كود تحديد مكان مربع عرض بطريقة الشبكة		العبارة
l = Tk()	أ	البدائل
o = label (blank)	ب	
rr. configure (background = orange)	ج	
e.grid (column = 0 , row = 0)	د	
11		م
أي من التالي كود تحديد مكان مربع عرض بطريقة الاحداثيات		العبارة
x.grid (column = 12 , row =11)	أ	البدائل
from tkinter import*	ب	
c. place(k = 100 , y = 150)	ج	
f = Tk()	د	

12		م
أي من التالي مثال على كود تحديد مكان مربع عرض بطريقة الحزم		العبارة
a. pack (side = ' left)	أ	البدائل
a. geometry("50 X 250")	ب	
j= mainloop()	ج	
yy.title = (" insert ")	د	

مفتاح الإجابة

الإجابة الصحيحة	العبرة	م
f = Entry(d)	كود انشاء مربع نص هو	1
u = Entry(h , width = 10)	أي من التالي مثال على كود تحديد عرض مربع نص بدل العرض الافتراضي	2
h = Entry(afd , width = 12 , bg = " black " , fg = "white")	أي من التالي مثال على كود تحديد لون الخلفية ولون النص في مربع نص	3
j.grid (column 2 , row = 2)	أي من التالي مثال على كود تحديد مكان مربع نص بطريقة الشبكة	4
d. place (x = 150 , y = 15)	أي من التالي مثال على كود تحديد مكان مربع نص بطريقة الاحداثيات	5
c. pack (side = ' right ')	أي من التالي مثال على كود تحديد مكان مربع نص بطريقة الحزم	6
d1 = Label (frame3)	أي من التالي مثال على كود انشاء مربع عرض	7
q = Label (models , width = 20)	أي من التالي مثال على كود تحديد عرض مربع عرض بدل العرض الافتراضي	8
u= label(uk , width = 60 , bg = " green " , fg = "red")	أي من التالي مثال على كود تحديد لون الخلفية ولون النص في مربع عرض	9
e.grid (column = 0 , row = 0)	أي من التالي مثال على كود تحديد مكان مربع عرض بطريقة الشبكة	10

الإجابة الصحيحة	العبرة	م
c. place(k = 100 , y = 150)	أي من التالي مثال على كود تحديد مكان عرض مربع عرض بطريقة الاحداثيات	11
a. pack (side = ' left)	أي من التالي مثال على كود تحديد مكان مربع عرض بطريقة الحزم	12

الفصل الثالث

الأحداث البرمجية

أداة الزر

كود إنشاء زر وكتابة نص عليه

كود تحديد لون الخلفية ولون النص

كود تحديد مكان الزر بطريقة الشبكة

كود تحديد مكان الزر بطريقة الاحداثيات

كود تحديد مكان الزر بطريقة الحزم

الاحداث البرمجية Events

مفهوم الحدث البرمجي

مفهوم حدث الضغط على زر

خطوات تنفيذ حدث برمجي

كود تنفيذ حدث برمجي

مشروع برمجية

خطوات انشاء مشروع برمجي متكامل

كود مشروع برمجي متكامل

مقدمة الفصل

أولاً - الهدف العام:

إكساب طلاب تكنولوجيا التعليم الرقمي المفاهيم والمهارات الأساسية الخاصة بالعمليات على الازرار والاحداث البرمجية.

ثانياً - الأهداف التعليمية:

بعد الانتهاء من هذا الفصل /الدرس، يتوقع من الطالب أن يكون قادراً على أن:

1. يكتب كود إنشاء زر وكتابة نص عليه
2. يكتب كود تحديد لون الخلفية ولون النص
3. يكتب كود تحديد مكان الزر بطريقة الشبكة
4. يكتب كود تحديد مكان الزر بطريقة الاحداثيات
5. يكتب كود تحديد مكان الزر بطريقة الحزمة
6. يذكر مفهوم الحدث البرمجي
7. يذكر مفهوم حدث الضغط على زر
8. يشرح خطوات تنفيذ حدث برمجي
9. يكتب كود تنفيذ حدث برمجي
10. يوضح خطوات انشاء مشروع برمجي متكامل
11. يكتب كود مشروع برمجي متكامل

أداة الزر Button

كود إنشاء زر والنص الظاهر عليه

- كود إجباري
- يقوم بإنشاء زر وتسميته + وضع النص الظاهر عليه في نفس الوقت
- يكتب في سطر كود منفصل
- يمكن ان يكتب مع أكواد أخرى مثل تحديد لون الزر ولون النص الظاهر عليه
- إذا لم يتم كتابته مع تلك الأكواد سيقوم بايثون بضبط ألوان الزر حسب الاعدادات التلقائية لمربعات النص

قاعدة كتابة كود زر (إنشاء + النص الظاهر)	
<code>button_name = Button (frame_name , text=" sentence ")</code>	
مثال على كود زر (إنشاء + النص الظاهر)	
<code>x = Button (mains , text = "click")</code>	انشاء مربع نص - يتم التعامل معه بالاسم <code>x</code> يتبع النافذة <code>mains</code> - النص الظاهر كلمة <code>click</code>
<code>btn1 = Button(sign_frame , text = " Cancel ")</code>	انشاء مربع نص - يتم التعامل معه بالاسم <code>btn1</code> - يتبع النافذة <code>sign_frame</code> - النص الظاهر كلمة <code>Cancel</code>

كود تحديد لون الخلفية ولون النص

- كود تنظيمي
- يقوم بضبط لون الزر ولون النص الظاهر عليه
- يفضل أن يتم كتابته مع كود الانشاء في سطر واحد
- يمكن ان يتم كتابة الثلاث اكواد (الانشاء + النص الظاهر + الألوان) في سطر واحد

قاعدة كتابة كود زر (إنشاء + النص الظاهر + ألوان)	
<code>button_name = Button (frame_name , text=" sentence " , bg = " color " , fg = " color ")</code>	
مثال على كود زر (إنشاء + النص الظاهر + ألوان)	
<code>x = Button (mains , text = "click" , bg = " black " , fg = "white")</code>	انشاء أداة <code>x</code> ، تتبع النافذة <code>mains</code> ، النص الظاهر <code>click</code> ، لون الأداة أسود ، لون النص الظاهر عليها أبيض
<code>btn1 = Button(sign_frame , text = " Cancel " , bg = " green " , fg = "red")</code>	انشاء أداة <code>btn1</code> ، تتبع النافذة <code>sign_frame</code> ، النص الظاهر كلمة <code>Cancel</code> ، لون الأداة اخضر ، لون النص الظاهر عليها احمر

كود تحديد مكان زر بطريقة الشبكة

- كود تنظيمي
- يقوم بضبط مكان ظهور الأداة ، بناء على صفوف مرقمة من 0 وتزيد كلما هبط للأسفل ، وأعمدة مرقمة من 0 وتزيد كلما اتجهت لليسا

قاعدة كتابة كود ضبط مكان زر بطريقة الشبكة Grid	
<code>button_name.grid (column = value , row = value)</code>	
مثال على كود ضبط مكان زر بطريقة الشبكة Grid	
<code>x.grid (column = 0 , row = 0)</code>	ضبط مكان أداة x في شبكة النافذة ، الصف الأول من الأعلى ، العمود الأول من اليسار
<code>ff.grid (column = 2 , row = 3)</code>	ضبط مكان أداة ff في شبكة النافذة ، الصف الثالث من الأعلى ، العمود الرابع من اليسار

كود تحديد مكان مربع زر بطريقة الاحداثيات

- كود تنظيمي
- يقوم بضبط مكان ظهور الأداة ، بناء على إحداثيات **x** مرقمة من 0 وتزيد كلما اتجهت لليسا ، **y** مرقمة من صفر وتزيد كلما اتجهت للأسفل

قاعدة كتابة كود ضبط مكان زر بطريقة الإحداثيات Place	
<code>button_name.place (x = value , y = value)</code>	

مثال على كود ضبط مكان زر بطريقة الإحداثيات Place	
x. place (x = 100 , y = 150)	ضبط مكان أداة x في احداثيات النافذة ، موقعها على النقطة x يساوي 100 ، موقعها على النقطة y يساوي 150
ff. place (x = 200 , y = 300)	ضبط مكان أداة ff في احداثيات النافذة ، موقعها على النقطة x يساوي 200 ، موقعها على النقطة y يساوي 150

كود تحديد مكان مربع زر بطريقة الحزم

- كود تنظيمي
- يقوم بضبط مكان ظهور الأداة ، بناء على تحديد مكان واضح (top ، left ، right ، botton)

قاعدة كتابة كود ضبط مكان زر بطريقة الحزم Pack	
button_name.pack (side = ' direction ')	
مثال على كود ضبط مكان زر بطريقة الحزم Pack	
x. pack (side = ' top ')	ضبط مكان أداة x في احداثيات النافذة ، موقعها الجانب العلوي
ff. pack (side = ' right ')	ضبط مكان أداة ff في احداثيات النافذة ، موقعها الجانب الايمن

الاحداث البرمجية

مفهوم الحدث البرمجي

- هو قيام البرنامج بإنتظار (رد فعل / حدث) يقوم به المستخدم حتى يقوم البرنامج بتنفيذ أمر محدد
- فقد يحتوي البرنامج على عدة أزرار أوامر : الأول sign in للدخول ، والثاني sign up للتسجيل ، والثالث sign out للخروج النهائي من البرنامج ، ويقوم البرنامج بإنتظار (الحدث) الذي سيقوم به المستخدم وهو (الزر الذي سيضغط عليه تحديداً) ، وعندما يختار المستخدم أحد الأزرار الثلاثة : سيقوم البرنامج بتنفيذ الأوامر الخاصة به

مفهوم حدث الضغط على زر

- من أشهر الأحداث البرمجية هو حدث قيام المستخدم بالضغط على زر ، ولكن توجد أحداث أخرى مثل قيام المستخدم بإختيار أمر من قائمة البرنامج ، أو كتابة قيمة ما ثم الضغط على زر enter مثلا
- إلا أنه في بداية تعلم لغة البرمجة يتم التركيز على حدث الضغط على زر بشكل أكثر، وبعد أن يتمرس المبرمج في برمجة الأدوات المختلفة للإدخال والإخراج البرمجي يمكنه إختيار طرق أخرى لتنفيذ الأحداث

خطوات تنفيذ حدث برمجي

- **الخطوة الأولى :** انشاء دالة - في منطقة الدوال في المشروع - تحتوي على كافة المتغيرات والعمليات المطلوب تنفيذها في الحدث
- **الخطوة الثانية :** ربط الدالة بأداة مثل الزر - عن طريق كتابة اسم الدالة داخل كود انشاء الزر من البداية

كود تنفيذ حدث برمجي

والذي يتم فيه انشاء الدالة + ربطها بزر

الخطوة الأولى - كود انشاء الدالة	
<code>def function_name () :</code>	كود انشاء الدالة
<code>Variable_name = entry_name .get()</code>	كود انشاء متغير يستقبل المدخل من مربع النص
Code	العمليات الحسابية والمنطقية
<code>Label_name.configure (text = (value , variable))</code>	كود اخراج النتائج إلى مربعات العرض ، سواء كانت قيم أو متغيرات
مثال على كود انشاء الدالة	
<code>def calc () :</code>	كود انشاء دالة بالاسم calc
<code>x = float (entry1 .get())</code>	كود انشاء المتغيرين x و y من المربعين entry1 و entry2
<code>y = float (entry2 .get())</code>	
<code>z = x + y</code>	العملية الحسابية (جمع عددين)
<code>lblresult.configure (text = (z))</code>	كود اخراج نتيجة الجمع إلى مربع العرض

الخطوة الثانية - كود الزر (انشاء + النص الظاهر) مع ربطه بالدالة	
<code>button_name = Button (frame_name , text= "sentence" , command = function_name)</code>	
لا يجب نسيان ان تحديد كود مكان الزر إجباري	
<code>button_name . place (x = value , y = value)</code>	
مثال على كود الزر (انشاء + النص الظاهر) مع ربطه بالدالة	
<code>btn1 = Button (pro , text= "click " , command = repeat)</code>	
انشاء أداة زر بالاسم <code>btn1</code> - يتبع النافذة <code>pro</code> - النص الظاهر <code>click</code> - يحمل الدالة <code>repeat</code>	
<code>btn1 . place (x = 100 , y = 100)</code>	كود مكان الزر

خطوات انشاء مشروع متكامل

لانشاء مشروع متكامل في بايثون ، يجب أن يتم تنفيذ جميع الخطوات التالي:

م	الخطوة	الكود
1	إستيراد مكتبة tkinter	<code>from tkinter import *</code>
2	إنشاء نافذة frame واحدة على الأقل	<code>frame_name = Tk()</code>
3	انشاء أدوات الادخال الرسومية ، مثل entry ، وتحديد أماكنها	<code>entry_name = Entry(frame_name)</code>
		<code>entry_name.grid (column = value, row = value)</code>
4	انشاء أدوات الاخراج الرسومية ، مثل label ، وتحديد أماكنها	<code>label_name = label(frame_name)</code>
		<code>label_name.grid (column = value, row = value)</code>
5	انشاء الدوال الخاصة بالعمليات الحسابية والمنطقية	<code>def function_name () :</code>
		<code>Variable_name = entry_name .get()</code>
		Code
6	انشاء أدوات الاحداث ، مثل button ، وتحديد أماكنها والدوال المرتبطة بها	<code>btn1= Button (frame_name, text= "sentence " , command = command_name)</code>
		<code>button_name . grid (column = value, row = value)</code>
7	غلق النافذة والبرنامج	<code>frame_name .mainloop()</code>

مثال لمشروع برمجي يقوم فيه المستخدم بإدخال نصف القطر في مربع النص
ثم يضغط على الزر ليقوم البرنامج بإخراج مساحة الدائرة في مربع العرض

م	الخطوة	الكود
1	إستيراد مكتبة tkinter	<code>from tkinter import *</code>
2	إنشاء نافذة بالاسم main	<code>main = Tk()</code>
3	انشاء أداة لادخال نصف القطر بالاسم entry1 ، وتحديد مكانها	<code>entry1 = Entry(main)</code>
		<code>entry1.grid (column = 0 , row = 0)</code>
4	انشاء أداة لاجراج مساحة الدائرة بالاسم lbl1 ، وتحديد مكانها	<code>lbl1 = label (main)</code>
		<code>lbl1.grid (column = 0, row = 1)</code>
5	انشاء دالة بالاسم calc ، تستقبل نصف القطر من أداة entry1 ، ثم تحسب مساحة الدائرة ، ثم تخرج النتيجة في أداة lbl1	<code>def calc () :</code>
		<code>a = entry1 .get()</code>
		<code>Area = 3.14 * a * a</code>
		<code>Lbl1.configure (text = (area))</code>
6	انشاء أدوات الاحداث ، مثل button ، وتحديد أماكنها والدوال المرتبطة بها ، وتحديد مكانها	<code>btn1= Button (main, text= "go " , command = calc)</code>
		<code>btn1.grid (column = 0, row = 2)</code>
7	غلق النافذة والبرنامج	<code>main.mainloop()</code>

نشاط تعليمي على إنشاء المشاريع البرمجية

البيان	الوصف
رقم النشاط	1
عنوان النشاط	مشروع برنامج كلمة السر
وصف النشاط	اكتب مشروع برمجي يقوم فيه المستخدم بإدخال كلمة السر ، فإذا كانت صحيحة تظهر للمستخدم رسالة ترحيب ، وإذا لم تكن صحيحة تظهر للمستخدم رسالة بوجود خطأ ، يتم استخدام أدوات الادخال والإخراج الرسومية في تنفيذ المشروع
توقيت التنفيذ	خلال المحاضرة
طريقة التنفيذ	تعليم تعاوني - مجموعات صغيرة

تكليفات برمجية

قم بتنفيذ التكليف التالي ثم تواصل مع أستاذ المادة لتسليمه ومناقشته في المحاضرات التالية:

م	التكليف
1	اكتب مشروع برمجي يقوم فيه المستخدم بإدخال درجته ، وبناء على تلك الدرجة يتم طباعة نتيجته اذا كان ناجح أم راسب، يتم استخدام أدوات الادخال والإخراج الرسومية في تنفيذ المشروع

مصادر إثنائية من بنك المعرفة المصري



Egyptian Knowledge Bank
بنك المعرفة المصري

من خلال المصادر الالكترونية التالية والموجودة ببنك المعرفة المصري والمكتبة الرقمية؛ يمكنك الاستزادة حول المحتوى العلمي للفصل:

م	عنوان المصدر	الرابط
1	كتاب عن برمجة الانترنت	https://0811nxsk9-1106-y-https-ekbedu-ethraadl-com.mplbci.ekb.eg/Record/62540
2	كتاب عن برمجة الشبكات	https://0811nxsk9-1106-y-https-ekbedu-ethraadl-com.mplbci.ekb.eg/Record/62535

قاموس المصطلحات الواردة بالفصل

م	المصطلح بالعربية	المصطلح بالإنجليزية	الوصف
1	الحدث البرمجي	Event	هو قيام البرنامج بإنتظار (رد فعل / حدث) يقوم به المستخدم حتى يقوم البرنامج بتنفيذ أمر محدد
2	حدث الضغط على زر	Press	من أشهر الأحداث البرمجية هو حدث قيام المستخدم بالضغط على زر ، ولكن توجد أحداث أخرى مثل قيام المستخدم بإختيار أمر من قائمة البرنامج ، أو كتابة قيمة ما ثم الضغط على زر enter مثلا

اختبر نفسك

اختر الإجابة الصحيحة من بين البدائل الموجودة:

1		م
أي من التالي مثال على كود انشاء زر وكتابة نص علي		العبارة
t. configure (background = grey)	أ	البدائل
a = Tk()	ب	
class a:	ج	
a = Button(p , text = “ save “)	د	
2		م
أي من التالي مثال على كود تحديد لون الخلفية ولون النص		العبارة
hv= mainloop()	أ	البدائل
def __init__	ب	
z= Button (b, text=“ sign ” , bg = “ red “ , fg = “ white “)	ج	
m = float (input (“insert student marks “))	د	

3		م
أي من التالي مثال على كود تحديد مكان الزر بطريقة الشبكة		العبارة
t.grid (column = 1 , row = 1)	أ	البدائل
s. geometry("15 X 25")	ب	
from tkinter import*	ج	
s. pack (side = ' right)	د	
4		م
أي من التالي مثال على كود تحديد مكان الزر بطريقة الاحداثيات		العبارة
sfb = label (jkl)	أ	البدائل
z. place (x = 55 , y = 55)	ب	
a = a object	ج	
print (x)	د	
5		م
أي من التالي مثال على كود تحديد مكان الزر بطريقة الحزمة		العبارة
x. pack (side = ' top)	أ	البدائل
er= mainloop()	ب	
a12 = label (ff)	ج	
hh = Tk()	د	

6		م
قيام البرنامج بانتظار رد فعل من المستخدم		العبارة
الحدث البرمجي	أ	البدائل
الادخال البرمجي	ب	
الإخراج البرمجي	ج	
الكائن البرمجي	د	
7		م
حدث ناتج عن الضغط على زر دخول البرنامج		العبارة
إضافة تكرار برمجي	أ	البدائل
حدث الضغط على زر	ب	
إضافة مربع نص	ج	
إضافة نافذة برمجية	د	
8		م
الخطوة الأولى في انشاء حدث برمجي		العبارة
انشاء مدخلات	أ	البدائل
انشاء عمليات	ب	
انشاء دالة	ج	
ربط الدالة بزر	د	

9		م
أي من التالي مثال على كود انشاء متغيرات وربطها بمربع نص		العبارة
Input	أ	البدائل
Insert	ب	
Entry	ج	
Get	د	
10		م
اول كود في انشاء مشروع برمجي		العبارة
from tkinter import*	أ	البدائل
frame_name = Tk ()	ب	
def function_name: ()	ج	
frame_name .mainloop()	د	
11		م
أي من التالي مثال على كود ربط زر بدالة برمجية		العبارة
y. geometry("11 X 21")	أ	البدائل
a= Button (m, text= "ok" , command = power2)	ب	
from tkinter import*	ج	
d. configure (background = black)	د	

مفتاح الإجابة

الإجابة الصحيحة	العبرة	م
a = Button(p , text = " save ")	أي من التالي مثال على كود انشاء زر وكتابة نص علي	1
z= Button (b, text=" sign ", bg = " red " , fg = " white ")	أي من التالي مثال على كود تحديد لون الخلفية ولون النص	2
t.grid (column = 1 , row = 1)	أي من التالي مثال على كود تحديد مكان الزر بطريقة الشبكة	3
z. place (x = 55 , y = 55)	أي من التالي مثال على كود تحديد مكان الزر بطريقة الاحداثيات	4
x. pack (side = ' top)	أي من التالي مثال على كود تحديد مكان الزر بطريقة الحزمة	5
الحدث البرمجي	قيام البرنامج بانتظار رد فعل من المستخدم	6
حدث الضغط على زر	حدث ناتج عن الضغط على زر دخول البرنامج	7
انشاء دالة	الخطوة الأولى في انشاء حدث برمجي	8

الإجابة الصحيحة	العبارة	م
Get	أي من التالي مثال على كود انشاء متغيرات وربطها بمربع نص	9
* from tkinter import	اول كود في انشاء مشروع برمجي	10
a= Button (m, text= "ok" , command = power2)	أي من التالي مثال على كود ربط زر بدالة برمجية	11

قائمة المراجع والمصادر

قائمة المصادر والمراجع

(جميعها من بنك المعرفة المصري ماعدا بعض مواقع الانترنت)

أولاً : المراجع العربية :

إيمان شعبان أحمد إبراهيم (2020) . أثر مستوى التغذية الراجعة الموجزة والتفصيلية في بيئة التعلم المصغر عبر الويب النقال على تنمية مهارات برمجة مواقع الإنترنت التعليمية لدى طلاب معلمي الحاسب الآلي ، كلية التربية جامعة سوهاج ، (73) ، 69-137.

إيهاب حسيب حبيب & إبراهيم محمد رشوان & صالح أحمد شاکر (2020). أثر اختلاف نمط الدعم ببيئة تعلم تكيفية فى تنمية الجانب الأدائى لمهارات البرمجة لدى طلاب المرحلة الثانوية ، كلية التربية جامعة كفر الشيخ ، 20(1) ، 243-273.

حسنا عبدالعاطي الطباخ & آية طلعت أحمد إسماعيل (2019). التفاعل بين نمط محفزات الألعاب الرقمية (تكيفي/تشاركي) ونوع التغذية الراجعة (فورية/مؤجلة) وأثره على تنمية مهارات البرمجة والانخراط لدى طلاب تكنولوجيا التعليم ، رابطة التربويين العرب ، (108) ، 60-132.

سعاد محمد عمر (2016). برنامج مقترح قائم على النظم الخبيرة لتنمية مهارات البرمجة الشيئية لدى تلاميذ الصف الثالث الإعدادي. رسالة ماجستير، كلية التربية، جامعة الفيوم.

- سيد نوح سيد عبدالجواد & عبدالرحمن كامل عبدالرحمن محمود & هاني محمد عبده الشيخ (2019). أثر نمط التغذية الراجعة المقدمة من خلال برنامج قائم على الذكاء الاصطناعي في تنمية مهارات البرمجة لدى طلاب الصف الثالث من الحلقة الثانية من التعليم الأساسي ، كلية التربية جامعة الفيوم ، 12(4)، 179-219.
- عبدالرحمن بن علي العثمان & فيصل بن عبدالعزيز المواش (2020). أثر تدريس البرمجة باستخدام سكراتش (Scratch) على الدافعية الذاتية نحو تعلم البرمجة لطلاب المرحلة الابتدائية بالرياض ، مجلة الدراسات التربوية والنفسية جامعة السلطان قابوس، 14(1)، 54-70.
- علي بن سويعد علي القرني (2020). أثر استخدام التعلم المصغر Microlearning على تنمية مهارات البرمجة والدافعية للتعلم لدى طلاب الصف الأول الثانوي. كلية التربية جامعة أسيوط ، 36(2)، 465-492.
- ماريان ميلاد (2017) . فاعلية نمط التعلم التشاركي القائم على مراسي التعلم الإلكترونية في تدريس لغة البرمجة سكراتش لتنمية بعض المهارات الأدائية والتفكير التكنولوجي بالمرحلة الإعدادية ، كلية التربية جامعة أسيوط ، 33(9) ، 263-309.
- محمد رفعت البسيوني (2012). تطوير بيئة تعلم الكترونية في ضوء نظريات التعلم البنائية لتنمية مهارات البرمجة الكائنية لدى طلاب معلمي الحاسب. مجلة كلية التربية -جامعة المنصورة.

- مسلم أحمد يوسف المالكي & إسلام جابر أحمد علام (2019). أثر اختلاف بعض متغيرات تصميم الكتاب الإلكتروني في تنمية مهارة برمجة الحاسب الآلي لطلاب الصف الأول الثانوي، كلية التربية جامعة أسيوط ، 35(4) ، 223 – 250.
- زينب حسن حامد السلامي (2016) . " نمطا الدعم التعليمي بإستخدام الواقع المعزز في بيئة تعلم مدمج وأثرها على تنمية التحصيل وبعض مهارات البرمجة والانخراط في التعلم لدى طلاب كلية التربية النوعية مرتفعي ومنخفضي الدافعية للإنجاز " ، الجمعية المصرية لتكنولوجيا التعليم ، (1) 26 ، 114-3 .

ثانياً- المراجع الأجنبية:

- Wahid, S. Z. (2018). The Effectiveness of Gamification in Improving Student Performance for Programming Lesson. **Kolokium Pembentangan Kertas Penyelidikan Dan Inovasi TVET Peringkat Politeknik dan Kolej Komuniti Zon Utara**, 1-7.
- Gokce, S. & other(2017): An Analysis of Mathematics Education Students Skills in the Process of Programming and Their Practices of Integrating It into Their Teaching, International Education Studies: Published by Canadian Center of Science and Education.
- Soares, A.(2014): Reflection on Teaching App Inventor For Non-Beginner Programmers: Issues, Challenges and Opportunities, Information Systems Educaion Journal.

ثالثاً - مواقع الإنترنت

الموقع الرسمي لبنك المعرفة المصري

<https://www.ekb.eg/>

الموقع الرسمي لبايثون

<https://www.python.org/downloads/>

الموقع الرسمي لبيئة تطوير PyCharm

<https://www.jetbrains.com/pycharm/download/>

رابط تحميل بيئة PyDroid على جوجل بلاي

<https://play.google.com/store/apps/details?id=ru.iiec.pydroid3&hl=ar&gl=US>

الموقع الرسمي لميكروسوفت

<https://www.microsoft.com/ar-eg>

قائمة المحتويات

قائمة المحتويات

الصفحات	الموضوع
9-0	مقدمة المقرر والكتاب
0	الغلاف الخارجي للكتاب
1	الغلاف الداخلي للكتاب
2	لائحة مقرر مقدمة في البرمجة
3	أهداف المقرر
4	مقدمة عامة للمقرر والكتاب
28-10	الباب الأول - مفاهيم أساسية في البرمجة
28-11	الفصل الأول - مفاهيم أساسية في البرمجة
12	مقدمة الفصل - مفاهيم أساسية في البرمجة
20-13	المحتوى العلمي للمفاهيم الأساسية في البرمجة
21	أنشطة تعليمية على المفاهيم الأساسية في البرمجة
22	تكاليفات برمجية في المفاهيم الأساسية في البرمجة
23	مصادر إثنائية من بنك المعرفة المصري حول مفاهيم البرمجة
24	قاموس مصطلحات المفاهيم الأساسية في البرمجة
25	اختبر نفسك في المفاهيم الأساسية في البرمجة
28	مفتاح إجابة اختبار نفسك في المفاهيم الأساسية في البرمجة
80-29	الباب الثاني - بيئات التطوير البرمجية
59-30	الفصل الأول - بيئة تطوير بايثون في نظام ويندوز
31	مقدمة الفصل - بيئة تطوير بايثون في نظام ويندوز
49-32	المحتوى العلمي لبيئة تطوير بايثون في نظام ويندوز
50	أنشطة تعليمية على بيئة تطوير بايثون في نظام ويندوز
51	تكاليفات برمجية في بيئة تطوير بايثون في نظام ويندوز

الصفحات	الموضوع
52	مصادر اثرائية من بنك المعرفة المصري حول نظام ويندوز
53	قاموس مصطلحات بيئة تطوير بايثون في نظام ويندوز
54	اختبر نفسك في بيئة تطوير بايثون في نظام ويندوز
59	مفتاح إجابة اختبار نفسك في بيئة تطوير بايثون في نظام ويندوز
80-60	الفصل الثاني - بيئة تطوير بايثون في نظام اندرويد
61	مقدمة الفصل - بيئة تطوير بايثون في نظام اندرويد
70-62	المحتوى العلمي لبيئة تطوير بايثون في نظام اندرويد
71	أنشطة تعليمية على بيئة تطوير بايثون في نظام اندرويد
72	تكاليفات برمجية في بيئة تطوير بايثون في نظام اندرويد
73	مصادر اثرائية من بنك المعرفة المصري حول نظام اندرويد
74	قاموس مصطلحات بيئة تطوير بايثون في نظام اندرويد
75	اختبر نفسك في بيئة تطوير بايثون في نظام اندرويد
80	مفتاح إجابة اختبار نفسك في بيئة تطوير بايثون في نظام اندرويد
208-81	الباب الثالث - برمجة واجهات CLI
94-82	الفصل الأول - المتغيرات البرمجية Variables
83	مقدمة الفصل - المتغيرات البرمجية
87-84	المحتوى العلمي للمتغيرات البرمجية
88	أنشطة تعليمية على المتغيرات البرمجية
89	تكاليفات برمجية في المتغيرات البرمجية
90	مصادر اثرائية من بنك المعرفة المصري حول مبادئ البرمجة
91	قاموس مصطلحات المتغيرات البرمجية
92	اختبر نفسك في المتغيرات البرمجية
94	مفتاح إجابة اختبار نفسك في المتغيرات البرمجية
109-95	الفصل الثاني - الإدخال والإخراج البرمجي I/O

الصفحات	الموضوع
96	مقدمة الفصل - الإدخال والإخراج البرمجي
98-97	المحتوى العلمي - الإدخال البرمجي
100-99	المحتوى العلمي - الإخراج البرمجي
101	أنشطة تعليمية على الإدخال والإخراج البرمجي
103	تكاليفات برمجية في الإدخال والإخراج البرمجي
105	مصادر اثرائية من بنك المعرفة المصري حول قواعد البرمجة
106	قاموس مصطلحات الإدخال والإخراج البرمجي
107	اختبر نفسك في الإدخال والإخراج البرمجي
109	مفتاح إجابة اختبار نفسك في الإدخال والإخراج البرمجي
125-110	الفصل الثالث - العوامل البرمجية وخرائط التدفق
111	مقدمة الفصل - بيئة تطوير بايثون في نظام ويندوز
113-112	المحتوى العلمي - (العوامل الحسابية ، عوامل الاسناد)
116-114	المحتوى العلمي - خرائط التدفق
117	أنشطة تعليمية على العوامل البرمجية وخرائط التدفق
119	تكاليفات برمجية في العوامل البرمجية وخرائط التدفق
120	مصادر اثرائية من بنك المعرفة المصري حول مفاهيم البرمجة
121	قاموس مصطلحات العوامل البرمجية وخرائط التدفق
122	اختبر نفسك في العوامل البرمجية وخرائط التدفق
125	مفتاح إجابة اختبار نفسك في العوامل البرمجية وخرائط التدفق
155-126	الفصل الرابع - الشرط والتفرع البرمجي Branching
127	مقدمة الفصل - الشرط والتفرع البرمجي
132-128	المحتوى العلمي - مفاهيم في الشرط والتفرع البرمجي
133	المحتوى العلمي - التفرع الشرطي البسيط
134	المحتوى العلمي - التفرع الشرطي الثنائي

الصفحات	الموضوع
135	المحتوى العلمي - التفرع الشرطي المتعدد
137	أنشطة تعليمية على الشرط والتفرع البرمجي
143	تكاليفات برمجية في الشرط والتفرع البرمجي
146	مصادر اثرائية من بنك المعرفة المصري حول قواعد البرمجة
147	قاموس مصطلحات الشرط والتفرع البرمجي
148	اختبر نفسك في الشرط والتفرع البرمجي
154	مفتاح إجابة اختبار نفسك في الشرط والتفرع البرمجي
173-156	الفصل الخامس - الحلقات البرمجية Loop
157	مقدمة الفصل - الحلقات البرمجية
159-158	المحتوى العلمي - مقدمة في الحلقات التكرارية
160	المحتوى العلمي - الدمج بين الحلقات التكرارية وبين الشرط
161	المحتوى العلمي - الغاء احدى لغات حلقة تكرارية
163	المحتوى العلمي - الغاء جميع لغات حلقة تكرارية
165	أنشطة تعليمية على الحلقات التكرارية
166	تكاليفات برمجية في الحلقات التكرارية
167	مصادر اثرائية من بنك المعرفة المصري حول البرمجة في التعليم
168	قاموس مصطلحات الحلقات التكرارية
169	اختبر نفسك في الحلقات التكرارية
173	مفتاح إجابة اختبار نفسك في الحلقات التكرارية
189-174	الفصل السادس - الدوال البرمجية Functions
175	مقدمة الفصل - الدوال البرمجية
178-176	المحتوى العلمي - الدوال الجاهزة
180-179	المحتوى العلمي - الدوال الغير جاهزة
181	المحتوى العلمي - الدالة المختصرة Lamda

الصفحات	الموضوع
182	أنشطة تعليمية على الدوال البرمجية
183	تكاليفات برمجية في الدوال البرمجية
184	مصادر اثنائية من بنك المعرفة المصري حول برمجة الروبوت
185	قاموس مصطلحات الدوال البرمجية
186	اختبر نفسك في الدوال البرمجية
189	مفتاح إجابة اختبار نفسك في الدوال البرمجية
208-190	الفصل السادس - البرمجة الكائنية OOP
191	مقدمة الفصل - البرمجة الكائنية
192	المحتوى العلمي - البرمجة الكائنية
193	المحتوى العلمي - الفئات Class
196	المحتوى العلمي - الكائنات Object
196-195	المحتوى العلمي - انشاء الفئات والكائنات بالطريقة البسيطة
198-197	المحتوى العلمي - انشاء الفئات والكائنات بالطريقة المتقدمة
199	أنشطة تعليمية على البرمجة الكائنية
202	تكاليفات برمجية في البرمجة الكائنية
203	مصادر اثنائية من بنك المعرفة المصري حول برمجة الويب
204	قاموس مصطلحات البرمجة الكائنية
205	اختبر نفسك في البرمجة الكائنية
208	مفتاح إجابة اختبار نفسك في البرمجة الكائنية
287-209	الباب الرابع - برمجة واجهات GUI
240-209	الفصل الأول - مقدمة في الواجهات الرسومية
211	مقدمة الفصل - الواجهات الرسومية
225-212	المحتوى العلمي - الواجهات الرسومية
229-226	المحتوى العلمي - النوافذ البرمجية Frames

الصفحات	الموضوع
230	أنشطة تعليمية على الواجهات الرسومية
231	تكاليفات برمجية في الواجهات الرسومية
232	مصادر اثنائية من بنك المعرفة المصري حول البرمجة المرئية
233	قاموس مصطلحات الواجهات الرسومية
234	اختبر نفسك في الواجهات الرسومية
239	مفتاح إجابة اختبار نفسك في الواجهات الرسومية
266-241	الفصل الثاني - أدوات الادخال والإخراج البرمجي
242	مقدمة الفصل - أدوات الادخال والإخراج البرمجي
247-243	المحتوى العلمي - أداة الادخال البرمجي / مربع النص / Entry
252-248	المحتوى العلمي - أداة الاخراج البرمجي / مربع العرض / Label
253	أنشطة تعليمية على أدوات الادخال والإخراج البرمجي
256	تكاليفات برمجية في أدوات الادخال والإخراج البرمجي
258	مصادر اثنائية من بنك المعرفة المصري حول البرمجة والتعليم
259	قاموس مصطلحات أدوات الادخال والإخراج البرمجي
260	اختبر نفسك في أدوات الادخال والإخراج البرمجي
265	مفتاح إجابة اختبار نفسك في أدوات الادخال والإخراج البرمجي
287-267	الفصل الثالث - الأحداث البرمجية Events
286	مقدمة الفصل - الاحداث البرمجية
272-269	المحتوى العلمي - أداة الزر Button
275-273	المحتوى العلمي - الاحداث البرمجية
277-276	المحتوى العلمي - خطوات انشاء مشروع متكامل
278	أنشطة تعليمية على الاحداث البرمجية
279	تكاليفات برمجية في الاحداث البرمجية
280	مصادر اثنائية من بنك المعرفة المصري حول البرمجة المتقدمة

الصفحات	الموضوع
281	قاموس مصطلحات الاحداث البرمجية
282	اختبر نفسك في الاحداث البرمجية
287	مفتاح إجابة اختبار نفسك في الاحداث البرمجية
292-288	قائمة المراجع
300-293	قائمة المحتويات

الباب الثالث

برمجة واجهات CLI

الفصل الأول

المتغيرات البرمجية Variables

مفهوم المتغيرات في البرمجة

استخدامات المتغيرات في البرمجة

أنواع المتغيرات في بايثون

كود إنشاء متغير برمجي

قواعد كتابة أسماء المتغيرات البرمجية

مقدمة الفصل

أولاً - الهدف العام:

إكساب طلاب تكنولوجيا التعليم الرقمي المفاهيم والمهارات الأساسية الخاصة بالعمليات على المتغيرات البرمجية.

ثانياً - الأهداف التعليمية:

بعد الانتهاء من هذا الفصل /الدرس، يتوقع من الطالب أن يكون قادراً على أن:

1. يذكر مفهوم المتغيرات في البرمجة
2. يوضح استخدامات المتغيرات في البرمجة
3. يبين أنواع المتغيرات البرمجية
4. يكتب كود إنشاء متغير برمجي
5. يراعي قواعد كتابة أسماء المتغيرات البرمجية

مقدمة في البرمجة

مفهوم المتغيرات في البرمجة

هي أماكن في الذاكرة، يتم حجزها باستخدام تعليمات برمجية ، تسميتها بإسم رمزي ، تستخدم في تخزين القيم المطلوب التعامل معها في البرنامج.

استخدامات المتغيرات في البرمجة

يمكن توظيف المتغيرات البرمجية لتنفيذ العديد من المهام ، منها ما يلي:

- حجز مكان للمتغيرات المحتملة (المطلوبة في البرنامج) + (التي سيقوم المستخدم بتحديددها فيما بعد) ، مثل (برنامج يطلب من المستخدم كتابة الإسم ، كلمة المرور)
- حجز مكان للمتغيرات التي سيقوم المبرمج باستخدامها في تنفيذ عمليات أخرى ، مثل (برنامج يطلب من المستخدم إدخال قيمة العملة بالجنيه ، ليقوم البرنامج بتحويلها للدولار عن طريق ضرب القيمة \times سعر الصرف الحالي)

أنواع المتغيرات البرمجية

يمكن تصنيف أنواع المتغيرات في بايثون إلى ما يلي:

أولاً - المتغيرات البسيطة

1 - المتغيرات الرقمية العددية Number

تستخدم في تخزين القيم العددية ، وتنقسم بدورها إلى :

- **Int** : متغير العدد الصحيح الذي يستخدم مع الاعداد التي لا تقبل القسمة وليس لها كسور مثل عدد الطلاب .
- **Float** : متغير العدد الحقيقي العشري الذي يستخدم مع الاعداد التي قد تحتوي على علامات عشرية وكسور مثل المساحات ودرجة الحرارة.
- **Complex** : متغير العدد المركب الذي يستخدم في الحالات البرمجية الخاصة بالمسائل الحسابية والفيزيائية والهندسية المعقدة ، ويضاف للرقم حرف j ، مثل $4j$.

2 - المتغيرات النصية String

تستخدم في تخزين القيم النصية ، سواء كانت حرف أو كلمة أو جملة أو فقرة كاملة ، مثل الاسم والوظيفة والعنوان .

3 - المتغيرات المنطقية Boolean

تستخدم في وضع شروط منطقية لقيم لا توجد بها سوا حالتان ، مثل ذكر وأنثى ، ناجح وراسب ، ولها قيمتين فقط إما True أو False ، وليس لها قيم أخرى

ثانياً - المتغيرات المركبة

تتكون من مجموعة من المتغيرات البسيطة ، وهي أربع أنواع على النحو التالي:

- **Tuple** : مصفوفة من المتغيرات، لها حجم ثابت (عدد متغيرات ثابت) ، يحدد المبرمج قيمتها ولا يمكن تغييرها بعد ذلك ، مثل أيام الأسبوع ، منطقياً عددها 7 (ثابت) ولها قيم محددة (السبت ، الأحد).

- **Set** : مصفوفة من المتغيرات، ليس لها حجم ثابت (مرنة في تغيير عدد متغيراتها) ، يحدد المبرمج قيمتها ولا يمكن تغييرها بعد ذلك بشكل سهل أو مباشر ، مثل أسماء المنتجات ، هي ليس لها عدد ثابت (قد يضاف أو يحذف منتج) لكن أسماؤها ثابتة
- **List**: مصفوفة من المتغيرات ، ليس لها حجم ثابت (مرنة الحجم) ، يمكن تغيير قيمتها في أي وقت (سهلة التعديل) ، مثل أسعار المنتجات، قد يتغير عدد المنتجات ، وأيضاً يتغير أسعارها
- **Dict**: عبارة عن جدول من المتغيرات يتكون من عامودين ، الأول رقم يمثل مفتاح للمتغير (Key) ، والثاني يمثل قيمة المتغير نفسه (Value) ، ويمكن بسهولة إضافة وحذف وتعديل قيم المتغيرات فيه ، وله نطاق واسع من الاستخدامات مثل القواميس المترجمة نفسها

كود إنشاء متغير برمجي

والذي يتم فيه إنشاء متغير + وضع قيمة مبدئية بداخله في نفس الوقت

قاعدة كتابة كود انشاء متغير	
variable_name = value	
مثال على اكواد انشاء متغيرات	
name = "Ali"	متغير - اسمه name - نوعه نص - قيمته "Ali"
age = 20	متغير - اسمه age - نوعه عدد صحيح - قيمته 20
length = 175.5	متغير - اسمه length - عدد عشري - قيمته 175.5
success = true	متغير - اسمه success - منطقي - قيمته true

قواعد كتابة أسماء المتغيرات البرمجية

يفضل أن يتم تسمية المتغير باسم يدخل بشكل منطقي على القيمة المطلوب منه تخزينها ، مثل أن يتم تسمية متغير يتم فيه تخزين اسم موظف بـ `employee` ، وعمر الموظف بـ `age` ، مع الوضع في الاعتبار مايلي:

- يفضل استخدام الأحرف الصغيرة في كتابة إسم المتغير ، ولو أردت استخدام الأحرف الكبيرة فيجب أن تتذكر أن البايتون حساس لحالة الأحرف.
- قد يتكون اسم المتغير من حرف واحد أو كلمة واحدة ، وإذا أردت أن يكون إسم المتغير أكثر من كلمة : ضع بين الكلمات (_) ولا تضع مسافات أبداً
- يمنع أن يكون الحرف الأول من إسم المتغير (رقم : مثل 1) ، ويمنع أيضاً أن يكون (علامة مثل : &) ، ويمنع أن يتم البدء بمسافة ، مع إمكانية وضع أي الأرقام والعلامات بعد الحرف الأول.
- توجد كلمات محجوزة في بايثون تستخدم في التعليمات البرمجية ، مثل (IF ، For) وبالتالي يمنع استخدامها كأسماء للمتغيرات.

أنشطة تعليمية على المتغيرات البرمجية

رقم النشاط	عنوان النشاط	وصف النشاط	توقيت التنفيذ	طريقة التنفيذ
1	أكتب الكود البرمجي	اكتب الكود الخاص بتعريف متغير يمثل إسم الطالب	خلال المحاضرة	تعليم تعاوني مجموعات صغيرة
2	إستنتج نتيجة تنفيذ كود برمجي	استنتج نتيجة تنفيذ الكود التالي marks = 50		
3	اكتب برنامج بايثون	اكتب برنامج بايثون يتم فيه ادخال متغيرات لطالب في الجامعة ، وتشمل المتغيرات : الإسم ، العمر ، العنوان		
4	إستخرج الخطأ البرمجي	استخرج الخطأ البرمجي في الكود التالي age = 199		

تكليفات برمجية

قم بتنفيذ التكليفات التالية، ثم تواصل مع أستاذ المادة لتسليمها ومناقشتها في المحاضرات التالية:

م	التكليف
1	اكتب الكود الخاص بتعريف متغير يمثل سعر أحد المنتجات
2	استنتج نتيجة تنفيذ الكود التالي "adress = "13 Alhosini st"
3	اكتب برنامج بايثون يتم فيه ادخال متغيرات لأحد المنتجات، وتشمل المتغيرات: الاسم ، السعر ، الموديل
4	استخرج الخطأ البرمجي في الكود التالي name = bakr

مصادر إثنائية من بنك المعرفة المصري



Egyptian Knowledge Bank
بنك المعرفة المصري

من خلال المصادر الالكترونية التالية والموجودة ببنك المعرفة المصري والمكتبة الرقمية؛ يمكنك الاستزادة حول المحتوى العلمي للفصل:

م	عنوان المصدر	الرابط
1	مقالة عن برامج الكمبيوتر	https://08124xsk3-1106-y-https-onlinebutterfly-education.mplbci.ekb.eg/index.php/Article/details/99311/%D8%A7%D9%84%D8%A8%D8%B1%D8%A7%D9%85%D8%AC%D9%8A%D8%A7%D8%AA
2	كتاب عن البرمجة الخطية وواجهات CLi	https://08124xsk3-1106-y-https-onlinebutterfly-education.mplbci.ekb.eg/index.php/Article/details/99311/%D8%A7%D9%84%D8%A8%D8%B1%D8%A7%D9%85%D8%AC%D9%8A%D8%A7%D8%AA

قاموس المصطلحات الواردة بالفصل

م	المصطلح بالعربية	المصطلح بالإنجليزية	الوصف
1	المتغيرات	Variables	هي أماكن في الذاكرة ، يتم حجزها بإستخدام تعليمات برمجية ، تسميتها بإسم رمزي ، تستخدم في تخزين القيم المطلوب التعامل معها في البرنامج.
2	العدد الصحيح	Integer	عدد ليس به كسور ، يستخدم لقيم الكائنات التي لا تقبل القسمة مثل عدد الطلاب في مدرسة
3	العدد العشري	Flaot	عدد قابل لأن يحتوي على كسور ، هو اكثر شمولاً من العدد الصحيح ، يستخدم لقيم الكائنات التي تقبل الكسور مثل المساحات
4	العدد المركب	Complex	نوع من المتغيرات العددية يستخدم في البرامج الخاصة بالفيزيائيين ، ويتميز بوجود حرف z مع الرقم

اختبر نفسك

اختر الإجابة الصحيحة من بين البدائل الموجودة:

1		م
أي من التالي لا يعبر عن المتغيرات البرمجية		العبرة
ضبط عملية التفرع البرمجي	أ	البدائل
أماكن في الذاكرة	ب	
يستخدم في تخزين القيم العددية	ج	
تخزن مدخلات البرنامج	د	
2		م
من استخدامات المتغيرات في البرمجة		العبرة
ضبط أماكن الأدوات في النافذة	أ	البدائل
حجز مكان للدالة البرمجية	ب	
ضبط التفرع الشرطي المتعدد	ج	
حجز مكان لقيم محتملة يدخلها المبرمج	د	
3		م
لتخزين متغير يعبر عن عدد الطلاب يستخدم النوع		العبرة
Float	أ	البدائل
Integer	ب	
Tuple	ج	
List	د	

4		م
أي من التالي كود انشاء متغير منطقي		العبارة
Age = 10	أ	البدائل
Sale = 10	ب	
Male = true	ج	
"City = " Minia	د	
5		م
أي من التالي يعد طريقة خاطئة لكتابة اسم متغير برمجي		العبارة
4me	أ	البدائل
X	ب	
OO	ج	
Main	د	

مفتاح الإجابة

م	العبرة	الإجابة الصحيحة
1	أي من التالي لا يعبر عن المتغيرات البرمجية	ضبط عملية التفرع البرمجي
2	من استخدامات المتغيرات في البرمجة	حجز مكان لقيم محتملة يدخلها المبرمج
3	لتخزين متغير يعبر عن عدد الطلاب يستخدم النوع	Integer
4	أي من التالي كود انشاء متغير منطقي	Male = true
5	أي من التالي يعد طريقة خاطئة لكتابة اسم متغير برمجي	4me

الفصل الثاني

الادخال والإخراج البرمجي I / O

الادخال البرمجي Input

مفهوم الإدخال في البرمجة

استخدامات الإدخال في البرمجة

كود الإدخال البرمجي

الإخراج البرمجي Ouput

مفهوم الإخراج في البرمجة

استخدامات الإخراج في البرمجة

كود الإخراج البرمجي

مقدمة الفصل

أولاً - الهدف العام:

إكساب طلاب تكنولوجيا التعليم الرقمي المفاهيم والمهارات الأساسية الخاصة بالعمليات على الادخال والإخراج البرمجي.

ثانياً - الأهداف التعليمية:

بعد الانتهاء من هذا الفصل /الدرس، يتوقع من الطالب أن يكون قادراً على أن:

1. يذكر مفهوم الإدخال في البرمجة
2. يشرح استخدامات الإدخال في البرمجة
3. يكتب كود الإدخال البرمجي
4. يذكر مفهوم الإخراج في البرمجة
5. يوضح استخدامات الإخراج في البرمجة
6. يكتب كود الإخراج البرمجي

الادخال البرمجي

مفهوم الإدخال في البرمجة

هي أكواد برمجية تتيح للمستخدم بنفسه إدخال القيمة التي يريد ، وتخزينها داخل المتغيرات ، حيث يتوقف البرنامج وتنشط لوحة المفاتيح ، ويطلب البرنامج مباشرة من المستخدم كتابة القيمة التي يريد ، والتي بناء عليها يتم تنفيذ العمليات الحسابية والمنطقية في البرنامج

استخدامات الإدخال في البرمجة

يفيد استخدام الادخال البرمجي فيما يلي:

- التفاعل مع المستخدم من خلال تبادل البيانات بينه وبين البرنامج
- تلبية طلبات المستخدم، حيث يدخل المستخدم القيم التي يرددها وبناء عليها يتم تنفيذ المطلوب من البرنامج
- المرونة البرمجية والتي تزيد من قوة البرنامج

كود الإدخال البرمجي

والذي يتم فيه:

- انشاء متغير
- السماح للمستخدم بإدخال قيمة من عنده
- وضع رسالة للمستخدم تعرفه ما هو مطلوب إدخاله
- تخزين القيمة التي وضعها المستخدم في هذا المتغير

قاعدة كتابة كود الادخال البرمجي
<code>variable_name = input ("Message")</code>
مثال على كود الادخال لمتغير نصي (ادخال اسم شخص)
<code>name = input ("Add your name please ")</code>
انشاء متغير - اسمه name - يسمح بإدخال قيمة نصية - يظهر للمستخدم رسالة Add your name please
مثال على كود الادخال لمتغير عددي صحيح (ادخال عدد الطلاب)
<code>student_num =int (input("Insert correct password "))</code>
انشاء متغير - اسمه student_num - يسمح بإدخال قيمة عدد صحيح - يظهر للمستخدم رسالة Insert correct password
مثال على كود الادخال لمتغير عددي عشري (ادخال درجات الطالب)
<code>m = float (input ("insert student marks "))</code>
انشاء متغير - اسمه m - يسمح بإدخال قيمة عدد عشري - يظهر للمستخدم رسالة insert student marks

الإخراج البرمجي

مفهوم الإخراج في البرمجة

هي أكواد برمجية يكتبها المبرمج بهدف إخراج النتائج للمستخدم ، فعندما يطلب المستخدم مثلا معرفة درجته ؛ فإنه يتوقع أن يقوم البرنامج (بإخراج) هذه القيمة وعرضها له وإلا لن يكون البرنامج نافعا له ، بالتالي لاغني عن وجود أكواد إخراج البيانات في البرمجة بشكل عام

استخدامات الإخراج في البرمجة

يفيد استخدام الإخراج البرمجي فيما يلي:

- إظهار رسالة للمستخدم ، مثل رسالة الترحيب ورسالة تأكيد الخروج.
- إخراج قيمة متغير أو أكثر للمستخدم ، وهذا المتغير يكون نتيجة لأحد العمليات التي طلبها المستخدم من البرنامج
- الدمج بين إظهار رسالة للمستخدم ، وبين إخراج قيمة المتغيرات للمستخدم في نفس الوقت
- قد تستخدم أكواد الإخراج في إخراج نتيجة تنفيذ بعض العمليات الحسابية البسيطة أيضا

كود الإخراج البرمجي

والذي يتم فيه اظهار قيمة للمستخدم ، الموجودة بين القوسين

قاعدة كتابة كود الإخراج البرمجي
print (thing)
مثال على كود اخراج رسالة للمستخدم
print (“welcome to our program“)
جملة اخراج - تظهر للمستخدم الرسالة welcome to our program
مثال على كود اخراج قيمة داخل متغير
print (x)
جملة اخراج - تظهر للمستخدم القيمة المطبوعة داخل المتغير x
مثال على كود اخراج رسالة + قيمة متغير، وفي نفس السطر
print (“your marks is“ , a)
جملة اخراج - تظهر للمستخدم
الرسالة your marks is + القيمة المطبوعة داخل المتغير a
مثال على كود اخراج ناتج عملية حسابية بسيطة
print (5 + 6)
جملة اخراج - تظهر للمستخدم قيمة ناتج العملية الحسابية (5+6) وهي (11)

أنشطة تعليمية على الإدخال والإخراج البرمجي

أولاً - الإدخال البرمجي

رقم النشاط	عنوان النشاط	وصف النشاط	توقيت التنفيذ	طريقة التنفيذ
1	أكتب الكود البرمجي	اكتب الكود الذي يسمح للمستخدم بإدخال عنوانه	خلال المحاضرة	تعليم تعاوني مجموعات صغيرة
2	إستنتج نتيجة تنفيذ كود برمجي	استنتج نتيجة تنفيذ الكود التالي a=float (input (“Write house area: “))		
3	اكتب برنامج بايثون	اكتب برنامج بايثون يطلب فيه من المستخدم ادخال : بيانات لطالب في الجامعة ، وتشمل المتغيرات : الاسم ، العمر ، العنوان		
4	إستخرج الخطأ البرمجي	استخرج الخطأ البرمجي في الكود التالي Salary = Enter (“input max. Salary: “)		

ثانياً - الإخراج البرمجي

رقم النشاط	عنوان النشاط	وصف النشاط	توقيت التنفيذ	طريقة التنفيذ
1	أكتب الكود البرمجي	اكتب الكود الذي يطبع رسالة نجاح عملية الشحن	خلال المحاضرة	تعليم تعاوني مجموعات صغيرة
2	إستنتج نتيجة تنفيذ كود برمجي	استنتج نتيجة تنفيذ الكود التالي print("Access denied ")		
3	اكتب برنامج بايثون	اكتب برنامج بايثون يقوم بطباعة إسم الطالب مع وجود رسالة تفيد بذلك ، على أن تكون الرسالة وإسم الطالب في سطرين منفصلين		
4	إستخرج الخطأ البرمجي	استخرج الخطأ البرمجي في الكود التالي print "what is your name"		

تكاليفات برمجية

قم بتنفيذ التكاليفات التالية، ثم تواصل مع أستاذ المادة لتسليمها ومناقشتها في المحاضرات التالية:

أولاً- الإدخال البرمجي:

م	التكليف
1	اكتب الكود الذي يسمح للمستخدم بإدخال طول ضلع مربع
2	استنتج نتيجة تنفيذ الكود التالي x = float (input ("what is your mobile price ? "))
3	اكتب برنامج بايثون يطلب فيه من المستخدم ادخال : بيانات لأحد المنتجات، وتشمل المتغيرات : الإسم ، السعر ، الموديل
4	استخرج الخطأ البرمجي في الكود التالي employee_name = input (Enter name :)

ثانياً - الإخراج البرمجي:

م	التكليف
1	اكتب الكود الذي يطبع قيمة المتغير x
2	استنتج نتيجة تنفيذ الكود التالي print (edu)
3	اكتب برنامج بايثون يقوم بطباعة عنوان الطالب مع رسالة تفيد بذلك ، على أن يكونا في نفس السطر
4	استخرج الخطأ البرمجي في الكود التالي print (Enter your code)

مصادر إثنائية من بنك المعرفة المصري



Egyptian Knowledge Bank
بنك المعرفة المصري

من خلال المصادر الالكترونية التالية والموجودة ببنك المعرفة المصري والمكتبة الرقمية؛ يمكنك الاستزادة حول المحتوى العلمي للفصل:

م	عنوان المصدر	الرابط
1	كتاب عن القواعد العامة للغات البرمجة	https://0811nxsk9-1106-y-https-ekbedu-ethraadl-com.mplbci.ekb.eg/Record/62483
2	كتاب عن لغة سكراتش ودورها في التعليم	https://0810gxsjr-1106-y-https-search-mandumah-com.mplbci.ekb.eg/Record/635724

قاموس المصطلحات الواردة بالفصل

م	المصطلح بالعربية	المصطلح بالإنجليزية	الوصف
1	الادخال البرمجي	Input	هي أكواد برمجية تتيح للمستخدم بنفسه إدخال القيمة التي يريد ، وتخزينها داخل المتغيرات ، حيث يتوقف البرنامج وتنشط لوحة المفاتيح ، ويطلب البرنامج مباشرة من المستخدم كتابة القيمة التي يريد ، والتي بناء عليها يتم تنفيذ العمليات الحسابية والمنطقية في البرنامج
2	الإخراج البرمجي	Output	هي أكواد برمجية يكتبها المبرمج بهدف إخراج النتائج للمستخدم ، فعندما يطلب المستخدم مثلاً معرفة درجته ؛ فإنه يتوقع أن يقوم البرنامج (بإخراج) هذه القيمة وعرضها له وإلا لن يكون البرنامج نافعاً له ، بالتالي لاغني عن وجود أكواد إخراج البيانات في البرمجة بشكل عام

اختبر نفسك

اختر الإجابة الصحيحة من بين البدائل الموجودة:

1	م
أي من التالي لا يتبع مفهوم الادخال البرمجي	العبارة
انشاء كائن برمجي	أ
تخزين قيمة داخل متغير	ب
تعامل المستخدم مع الكود	ج
إضافة قيمة عددية	د
2	م
يمكن استخدام الادخال البرمجي في	العبارة
تكرار الكود	أ
انشاء متغير	ب
انشاء دالة جاهزة	ج
دمج الحلقات بالشرط	د
3	م
أي من التالي كود ادخال متغير نصي	العبارة
a = input float ("insert name ")	أ
a = input int ("insert name ")	ب
a = input bool ("insert name ")	ج
a = input ("insert name ")	د

4		م
أي من التالي لا يتبع مفهوم الإخراج البرمجي		العبارة
اظهار نتائج	أ	البدائل
تفاعل مع المستخدم	ب	
اخراج الكود المصدري	ج	
اظهار المخرجات	د	
5		م
يمكن استخدام الإخراج البرمجي في		العبارة
عرض رسالة للمستخدم	أ	البدائل
عرض حجم الشاشة	ب	
ربط دالة داخل كائن	ج	
ربط مربع نص بمتغير	د	
6		م
أي من التالي كود اخراج رسالة للمستخدم		العبارة
print (c)	أ	البدائل
print (a+b)	ب	
(5) print	ج	
print ("welcome ")	د	

مفتاح الإجابة

م	العبارة	الإجابة الصحيحة
1	أي من التالي لا يتبع مفهوم الادخال البرمجي	انشاء كائن برمجي
2	يمكن استخدام الادخال البرمجي في	انشاء متغير
3	أي من التالي كود ادخال متغير نصي	a = input ("insert name ")
4	أي من التالي لا يتبع مفهوم الاخراج البرمجي	اخراج الكود المصدري
5	يمكن استخدام الإخراج البرمجي في	عرض رسالة للمستخدم
6	أي من التالي كود اخراج رسالة للمستخدم	print ("welcome ")