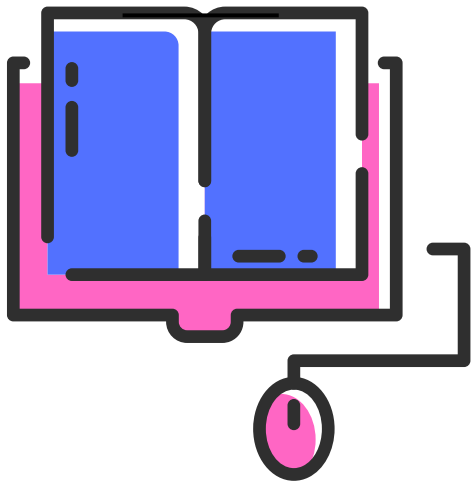
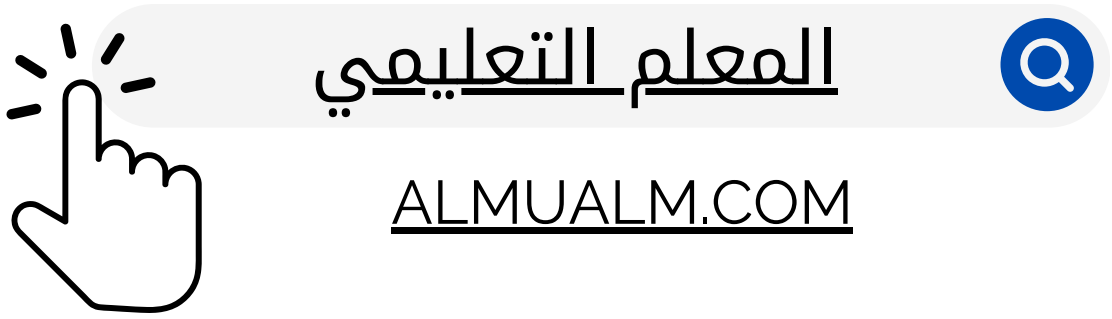
تم تحميل ورفع المادة على منصة



المعلم التعليمي

للعودة الى الموقع اكتب في بحث جوجل



ثاني متوسط
المهارات الرقمية

الفصل الدراسي الأول

ملخصات

اسم الطالب/هـ:

الشعبة:

الوحدة الأولى: إنتاج مقطع فيديو

ثاني متوسط –الفصل الأول

ملفات الوسائط:

يمكن التمييز بين ملفات الوسائط المختلفة في جهاز الحاسب بالتحقق من امتدادها، فلكل ملف اسم وامتداد مثلاً الملف (مدرستي.jpg) عبارة عن صورة اسمها "مدرستي" وامتدادها jpg والذي يظهر أن الملف عبارة عن صورة بتنسيق "JPEG".

أنواع الملفات الأكثر شيوعاً لملفات الوسائط وامتداداتها:

- ملف صورة (رسم، صورة): ".jpg" ".png" ".gif"
- ملف صورة (رسم متجه): ".svg" ".ai"
- ملف صوتي: ".mp3" ".aac"
- ملف فيديو: ".webm" ".mkv" ".avi" ".mp4"

مقاطع الفيديو:

يمكن تصوير مقاطع الفيديو باستخدام كاميرا الفيديو أو الهاتف الذكي أو الكاميرا الرقمية صغيرة الحجم أو الكاميرا الرقمية ذات العدسة الأحادية العاكسة (DSLR) أو كاميرا الويب المتصلة بالحاسب.

يشير الرمز (SD) إلى الدقة القياسية للفيديو (Standard Definition)، بينما يشير الرمز (HD) إلى الدقة العالية في الفيديو (High Definition) وتعني أن صورة الفيديو أو عدد الإطارات الملتقطة بالثانية أعلى في (HD)، وبالتالي جودة أفضل وأكثر وضوحاً وتتطلب مساحة تخزين أكبر.

يمكن ضغط ملفات الفيديو لتوفير مساحة التخزين وليسهل نقلها وتحميلها عبر الإنترنت، وهنالك أكثر من طريقة لضغط ملفات الفيديو وتصغير حجمها والتي يطلق عليها "برامج الترميز"، يتم تثبيت برامج الترميز بشكل تلقائي على الحاسب ويمكنك إضافة المزيد من برامج الترميز عند الحاجة. أمثلة على برامج الترميز: AV1 , H.264 , MPEG-4

يُظهر امتداد الفيديو نوع "الحاوية" وهي مجموعة من الملفات تظهر للمستخدم كملف واحد، وتتضمن الحاوية في معظم الأحيان برنامج ترميز الفيديو وترميز الصوت وقد تتضمن أشياء أخرى مثل الترجمة. أمثلة على الحاويات webm , mp4

أنواع ملفات الصوت:

ملفات صوتية غير مضغوطة: يتم تخزين البيانات بالشكل المطابق للتسجيل ويتمتع بأفضل جودة ممكنة ويشغل مساحات تخزين كبيرة، ومن أمثلتها تنسيق wav

ملفات صوتية مضغوطة: يفقد بعض الجودة عن الضغط ويكون حجم الملف أصغر بكثير ومن أمثلتها تنسيق mp3

من خصائص الصوت التي يتم استخدامها عند تحرير الصوت معدل العينة (Sample rate) وهو كمية التفاصيل بالصوت ومعدل البت (Bit rate) ويستخدم لتحديد درجة ضغط الصوت، كلما زادت قيم الخاصيتين زادت جودة الملف وحجمه.

ملفات الصور:

يمكن استخدام كاميرا الهاتف لالتقاط الصور أو الكاميرا الصغيرة أو الكاميرا الرقمية، الكاميرات الرقمية تنتج صوراً أكبر وأكثر وضوحاً ودقة بسبب امتلاكها عدسات أفضل ومستشعرات صور أكبر.

تتكون الصـورة الرقمية من مربعات صـغيرة متجاورة، لكل مربع من هذه المربعات لون محدد، وعند دمجها جميعاً يتم تكوين الصـورة، يسـمى كل مربع من هذه المربعات الصـغيرة "بكسـل"، لذا فإن صـورة بدقة 50 ميجا بكسـل تتكون من 50 مليون مربع صغير ملون.

بعض تنسيقات الصور الأكثر استخداما:

- JPEG بامتداد ".png" تنسيق مضغوط ويستخدم في الإنترنت
- GIF بامتداد ".gif" تدعم الشفافية والصور المتحركة
- PNG بامتداد ".png" يدعم الشفافية ويعطي جودة جيدة جداً
- TIFF بامتداد ".tif" تنسيق غير مضغوط بحجم وجودة أكبر

خطوات إنشاء فيلم أو فيديو:

أولاً: النص كتابة وصف تفصيلي تسلسلي لأحداث الفيلم يتضمن الأحداث والحوار والأبطال، وهو ما يعرف بالسيناريو.

ثانياً: جدول التصوير تقسيم كل مشهد إلى مجموعة من اللقطات وتخطيط وتدوين كيفية ظهور كل لقطة.

ثالثاً: مخطط القصة رسم المظهر الذي ستبدو عليه كل لقطة، مع كتابة التفاصيل والتعليمات على الرسم.

برنامج شوت كت (shotcut) لتحرير الفيديو:

موازنة اللون الأبيض (White Balance): ضبط الألوان لتبدو الصورة طبيعية بشكل مثالي.

مرشح البني الداكن (Sepia Tone): تأثير بني داكن هادئ على العين.

التأثيرات البصرية: تساعد على رفع جودة إخراج مقاطع الفيديو وتسهم في وضوح الرسائل المضمنة فيها.

التأثيرات الانتقالية: تستخدم لربط لقطة بأخرى.

الوحدة الثانية: مخطط المعلومات البياني

مفهوم مخطط المعلومات البياني:

هو تمثيل مرئي ملخص للمعلومات أو البيانات أو المعرفة، ويتميز بتقديم الموضــوع بطريقة مختصــرة ومركزة ويجذب الانتباه ويولّد الاهتمام، ويستخدم للأغراض التالية:

- نقل رسالة محددة بسرعة.
- تبسيط عرض كمية كبيرة من المعلومات.
- توضيح العلاقة بين البيانات وكيفية ارتباطها ببعضها البعض.
- عرض عوامل التغيير في البيانات على مدار فترة زمنية.
- تقديم مجموعة متنوعة من عناصر البيانات لإحداث تأثير بصري قوي.

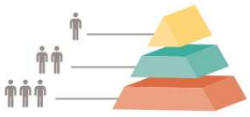
خصائص مخطط المعلومات البياني:

لإنشاء مخطط معلومات بياني هادف ومؤثر لابد من وضع الخصائص التالية في الاعتبار:

1. الكفاءة والدقة: عرض المعلومات في تسلسل سهل الفهم متضمناً الإحصائيات والحقائق الموثقة والدقيقة.
2. البساطة والوضوح: وذلك بتبسيط النصوص المعقدة لتسهيل فهم المعلومات.
3. الجاذبية والفعالية: وذلك بتقديم المعلومات بشكل جذاب للجماهير بطريقة موجزة وتصميم بسيط يجذب الانتباه.
4. التوازن: التوازن بين كمية البيانات وتصميم المخطط أمرٌ بالغ الأهمية لإيصال الرسالة الصحيحة.

أنواع مخطط المعلومات البياني واستخداماتها:

النوع	صورة	الوصف	الاستخدام
المخطط المفرد		يعرض المعلومات في سلسلة من نقاط البيانات المتصلة بواسطة خط مستقيم	يستخدم لعرض بيانات نتائج الاستطلاعات
مخطط قائم على صورة أو صورة بيانية		يعتمد على الصور ويستخدم نصوصاً وبيانات لشرح نقطة ما	مناسب لإنشاء الكتيّبات والملصقات التوضيحية
المخطط الزمني		يعرض الأحداث بترتيب زمني	مناسب لإظهار كيفية تغير شيء ما بمرور الوقت أو لشرح قصة طويلة ومعقدة
مخطط المقارنة		طريقة مرئية لمقارنة الخيارات المختلفة	يستخدم للمقارنة ولإظهار أوجه الشبه والاختلاف بين عدة عناصر
مخطط الخريطة أو الموقع		يقدم المعلومات بصرياً فيما يتعلق بالمواقع الجغرافية	يستخدم عند مقارنة الأماكن والثقافات

المخطط الهرمي		ينظم المعلومات حسب المستويات "الأهمية، الصعوبة، الدخل ... إلخ"	مناسب لمقارنة المستويات المختلفة وإظهار العلاقة بينها
مخطط السيرة الذاتية المرئية		سيرة ذاتية على شكل مخطط بياني	مناسب لإنشاء سيرة ذاتية غير تقليدية والإعلان والتسويق للشركات
المخطط الانسيابي		يلخص الخطوات المتبّعة في صنع منتج أو وصف سلسلة عملية	مثل إنشاء مخطط انسيابي لخوارزمية برنامج ما
المخطط التشريحي		يبسط المعلومات المعقدة ويشرحها بطريقة مرئية وفعالة	مناسب لعلم الأحياء والصحة والتعليم والتسويق

خطوات تصميم مخطط المعلومات البياني:

لإنشاء مخطط معلومات بياني جيّد يجب التوازن بين تقديم معلومات كافية وغير مبالغ فيها وذلك من خلال عدّة خطوات متسلسلة:

1. **اختيار موضوع** مثير للاهتمام يمكن دعمه بالمصادر الصحيحة.
2. **تحديد الجمهور المستهدف** بناءً على المستوى المعرفي المتوقع.
3. **تحديد الأهداف** ومعالجة أحد أكبر مشاكل الجمهور وحلّها.
4. **جمع البيانات** والمعلومات الموثوقة وذلك بالبحث في مصادر مختلفة وفرزها وتنظيمها.
5. **اختيار التصميم** المناسب ووضع تصور حول تقديم الموضوع بطريقة مناسبة وجذابة.
6. **المراجعة والنشر** بعد الانتهاء من تصميم مخطط المعلومات البياني ومراجعته والتأكد من خلوه من الأخطاء.

أدوات تصميم مخطط المعلومات البياني:

يستخدم تطبيق كانفا "Canva" لإنشاء مخططات المعلومات البيانية وهو تطبيق مجاني وسهل الاستخدام لتصميم الرسومات.

مميزات إضافة الأشكال والصور في مخطط المعلومات البياني:

يمكن أن تؤثر الأشكال والصور المصممة بعناية على رأي الجمهور حول موضوع ما، وتضيف الألوان الحيوية على محتوى مخطط المعلومات البياني، وتجذب الانتباه بشكل سريع وتساعد على التركيز على بيانات معينة.

مميزات تنسيق المستندات بصيغة PDF:

يعرض المستندات والرسومات بشكل صحيح بغضّ النظر عن الجهاز أو التطبيق أو نظام التشغيل أو المتصفح المستخدم.

أمثلة على برامج أخرى تستخدم لتصميم مخططات المعلومات البياني:

تطبيق إنفوجرام "Infogram" وتطبيق بيكتوشارت "Piktochart"

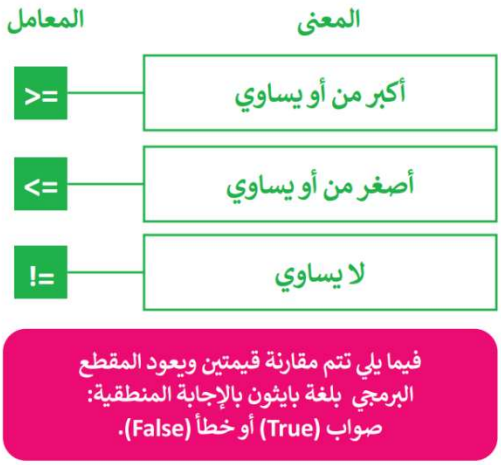
الوحدة الثالثة: البرمجة مع بايثون

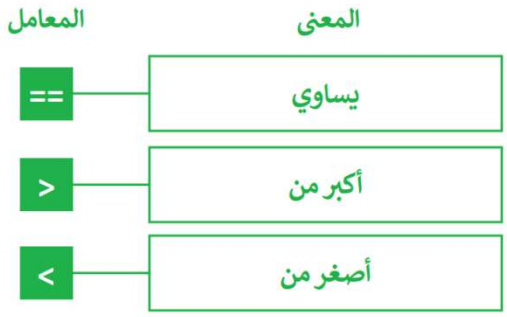
مفهوم بيئة التواصل باي تشارم (PyCharm Community):

بيئة تطوير متكاملة توفر العديد من الوحدات النمطية والحزم والأدوات لمساعدتك على البرمجة بلغة بايثون وتتوافق مع أنظمة التشغيل ويندوز ولينكس وماك أو إس.

المعاملات الشرطية:

تستخدم المعاملات الشرطية للتحكم في مسار البرنامج ولاتخاذ قرار في البرمجة، توجد عادةً كجزء من دالة "if" أو "else" حيث تقارن القيم وتعود بنتيجة صواب (True) أو خطأ (False) وبناءً على النتيجة ينفذ البرنامج العمليات المقابلة





فيما يلي تتم مقارنة قيمتين ويعود المقطع البرمجي بلغة بايثون بالإجابة المنطقية: صواب (True) أو خطأ (False).

مثال 1: المعاملات الشرطية

```
x=5
y=6
k=x<y
print(k)
```

True

```
x=5
y=6
z=x==y
print(z)
```

False

مقارنة بين معامل الإسناد (=) ومعامل المساواة (==):

معامل الاسناد (=) يغير قيمة المتغير ومعامل المساواة (==) سيقرأ قيمة المتغير ويرى ما أذا كان يساوي شيئاً ما.

المعاملات المنطقية في بايثون:

تستخدم لدمج مقارنات متعددة في المعاملات الشرطية وتستخدم لدمج شرطين أو أكثر ولفهم المعاملات المنطقية يجب العودة إلى جدول الحقيقة

المعاملات المنطقية

جدول الحقيقة:

هو جدول يسرد جميع المدخلات الممكنة للمتغير ويعرض مخرجاته وفقاً للمعامل المنطقي:

A	B	A and B	A or B	not A	not B
True	True	True	True	False	False
False	True	False	True	True	False
True	False	False	True	False	True
False	False	False	False	True	True

ترتيب المعاملات في البرمجة
()
**
* /
+ -
== > < <= >= =!
not
and
or

أمثلة على المعاملات المنطقية في بايثون:

```
x=5
y=6
k=x<10 and y<8
print(k)
```

True

```
x=True
y=False
z=x==y
print(z)
```

False

الجمل الشرطية في بايثون:

تستخدم لاتخاذ القرارات في لغة برمجة بايثون ولإنشاء برامج تستجيب لمدخلات المستخدم أثناء تنفيذها وتعطي نتائج مختلفة بناءً على المدخلات المختلفة

أنواع الجمل الشرطية:

جملة "if" الشرطية البسيطة: إذا تحقق الشرط فستنفذ العبارة التي تتبع if

جملة "if...else" الشرطية: إذا تحقق الشرط فسيتم تنفيذ العبارة التي تتبع if وإن لم يتحقق سيتم تنفيذ العبارة التي تتبع else

جملة "if...elif": تحتوي على أكثر من عبارة if شرطية، ويتحقق البرنامج من الشروط واحداً تلو الآخر فإذا تحقق أحد الشروط يتم تنفيذ ما تحت هذا الشرط ويتجاوز باقي الشروط وإذا لم يتحقق أياً من الشروط يتم تنفيذ جملة else

```
grade=12
if grade>=15:
    print("ممتاز")
elif grade>=10:
    print("جيد جداً")
else:
    print("تحتاج إلى المحاولة أكثر")
```

if الشرط:
عبارة 1
elif:
عبارة 2
else:
عبارة 3

جيد جداً

```
grade=7
if grade>=10:
    print("ناجح")
else:
    print("تحتاج إلى المحاولة")
```

if الشرط:
عبارة 1
else:
عبارة 2

تحتاج إلى المحاولة أكثر

```
grade=17
if grade>=10:
    print("ناجح")
```

if الشرط:
العبارة

ناجح

المسافة البادئة(Indentation):

يستخدمها البرنامج للإشارة إلى العبارات المعتمدة على تحقيق الشرط إذا لم تترك مسافة بادئة فستتلقى رسالة خطأ، الأسطر غير المزودة بمسافة بادئة ستنفّذ حتى إذا لم يتحقق الشرط.

مفهوم التداخل (Nesting)"

مصطلح يستخدم لوصف وضع كائن أو أكثر داخل كائن آخر، وفي برمجة الحاسب، العبارة المتداخلة عبارة موجودة داخل عبارة أخرى في المقطع البرمجي الأساسي، وتستخدم المسافة البادئة لتحديد الشرط الذي تنتمي إليه العبارة.

الجملة الشرطية "IF" المتداخلة:

عبارة عن جملة if البسيطة تكون موجودة داخل "متداخلة مع" جملة if الأخرى أو جملة if...else الشرطية، ويمكن دمج أي عدد من العبارات في أي مجموعة داخل بعضها البعض.

الحلقات (Loops):

تستخدم هذه الدالة عند الحاجة إلى تكرار مجموعة من الأوامر في النص البرمجي، وتسمح لك بتنفيذ سطر واحد أو مجموعة من المقاطع البرمجية عدة مرات، ويمكنك تحديد عدد التكرارات برقم معين أو ربطها بشرط محدد، ويدعم بايثون نوعين من الحلقات (`for`) و (`while`)

حلقة (for):

تستخدم لتكرار مجموعة من الأوامر لعدد محدد من المرات ويكون عدد التكرارات محدداً في قيم **دالة النطاق** (`range()`)، ويجب استخدام **المسافة البادئة** في الحلقات ليتعرف البرنامج على العبارات المضمنة في الحلقة والتي يتم تنفيذها في كل تكرار.

دالة النطاق (range()):

تستخدم مع الحلقة لتحديد عدد التكرارات ويسمى المتغير الذي يحسب التكرار **العداد (counter)** ويبدأ العد من 0 ويزيد بمقدار 1

```
#i يطبع قيمة
for i in range (5):  استخدم دالة النطاق لتحديد عدد مرات التنفيذ
    print(i)  يجب أن تسبق البيانات المتكررة بمسافة بادئة
```

```
0
1
2
3
4
```

القيمة 5 ليست مدرجة لأن العد يبدأ من 0 ويتوقف قبل الوصول للرقم

```
2
3
4
```

يمكنك تحديد مقدار البداية بإضافة قيمة، فمثلاً في النطاق range(2,5) يبدأ العد من 2 وينتهي عند 4

```
1
3
```

القيمة التلقائية للزيادة في دالة النطاق هي 1 ويمكن تغيير هذه القيمة بإضافة قيمة ثالثة لدالة النطاق تسمى "الخطوة"، فمثلاً في النطاق range(1,5,2) يبدأ العد من 1 وينتهي عند 4 ويزداد بمقدار 2

حلقة (while) الشرطية:

تستخدم عندما لا يكون عدد التكرارات معروفاً سابقاً، طالماً أن الحالة صحيحة "الشرط متحقق" فإن الحلقة تتكرر وتفحص بعد كل تكرار للتأكد من صحتها، وعندما تصبح الحالة خطأ، يتوقف التكرار وينتقل المقطع البرمجي إلى السطر التالي بعد الحلقة، وإذا كانت حالة الشرط خطأ من البداية لن يتم تنفيذ الحلقة وينتقل للسطر التالي بعدها، وأيضاً يجب أن تسبق الأوامر في حلقة **while** الشرطية **بمسافة بادئة**.

```
# يدخل المستخدم رقم ويستمر بالطباعة إلى أن يدخل المستخدم رقم صفر
while x! =0:
    print(x)
    x=int(input(":x أدخل قيمة"))
print("نهاية الحلقة")
```

```
أدخل قيمة x : 5
5
أدخل قيمة x : 6
6
أدخل قيمة x : 0
0
نهاية الحلقة
```

يمكن استخدام حلقة **while** الشرطية للتحقق من مدخلات المستخدم في متغير معين، مثل تحديد العدد المدخل بين 0 و 100

حلقة لا نهائية (infinite loop): إذا لم يصبح شرط حلقة **while** خطأ، فسينتهي بك الأمر بحلقة لا نهائية (infinite loop) وهي حلقة لا تنتهي أبداً، لذلك يجب عليك التأكد من تضمين أمر أو مجموعة أوامر تغير الحالة من صواب إلى خطأ لإيقاف حلقة **while**

عبارة الإيقاف (Break statement): تستخدم لإنهاء حلقة قبل أن تصبح حالتها خطأ وينتقل البرنامج إلى السطر المتواجد بعد الحلقة ويمكن استخدامها في حلقة **while** وحلقة **for**، وصيغتها **break**

الحلقات المتداخلة (Nested loops):

الحلقات المتداخلة هو وضع حلقة داخل حلقة ويسمى هذا الإجراء **بالتداخل** (Nesting) ويمكنك إدخال أي نوع من الحلقات داخل أي نوع آخر، فيمكنك إدخال حلقة **for** داخل حلقة **while** أو العكس، وتعد حلقات **for** المتداخلة الأكثر شيوعاً.

قواعد الحلقات المتداخلة:

- يجب أن تكون كل حلقة داخلية مدمجة بالكامل داخل الحلقة الخارجية ولا يمكن أن تتشابك الحلقات.
- لا يمكن استخدام نفس المتغير كعدّاد لحلقتين متداخلتين أو أكثر.
- الحلقة الداخلية يجب أن تكتمل أولاً.
- الحلقة الداخلية تنفذ جميع تكراراتها لكل تكرار من تكرارات الحلقة الخارجية.

أنماط الطباعة:

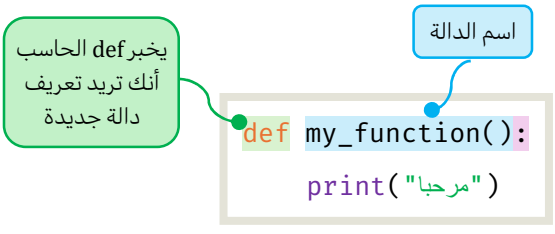
يمكن استخدام بايثون والحلقات المتداخلة لعرض الأنماط على الشاشة، ولطباعة أي نمط هناك هيكل عام عليك اتباعه وذلك بتحديد عدد الصفوف والأعمدة للنمط، تحدد الحلقة الخارجية عدد السطور المستخدمة وتحدد الحلقة الداخلية عدد الأعمدة المستخدمة.

الدوال (Functions):

الدالة هي مجموعة من الأوامر التي يتم تجميعها في مكان واحد مع إعطائها اسماً (تعريفاً) ويتم استدعائها عند الحاجة لها، ويحتوي البايثون على مجموعة متنوعة من الدوال الجاهزة مثل دالة **الطباعة** `print()` ودالة **النطاق** `range()` ودالة **الإدخال** `input()` وغيرها.

إنشاء دالة خاصة بك:

يعرف الجزء الأول من الدالة باستخدام أمر **تعريف الدالة** `def` متبوعاً **باسم الدالة**، وفي نهاية **رأس الدالة** هناك نقطتان ":" للإشارة إلى أن ما يلي عبارة عن مجموعة من المقاطع البرمجية، ويجب وضع مسافة بادئة لها، يتبع ذلك **محتوى الدالة**.

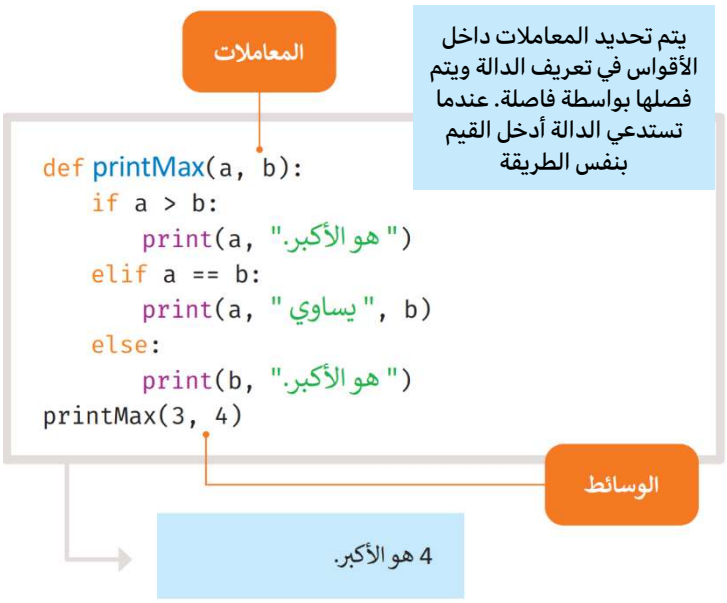


استدعاء دالة:

لاستدعاء دالة، استخدم اسم الدالة متبوعاً بأقواس ().

المعاملات والوسائط (Parameters and Arguments):

المُعامِلات (Parameters): المتغيرات التي يمكن الإعلان عنها في الدالة وتستخدم داخل الدالة ويشار إليها باسم **بالمتغيرات المحلية (Local Variables)** لأنه لا يمكن الوصول إليها إلا من خلال الدالة.



الوسائط (Arguments): المتغيرات التي تم تمريرها إلى الدالة لتنفيذها، وتتلقى المتغيرات المحلية للدالة قيم الوسائط كمدخلات ثم تتابع تنفيذ الدالة.

عبارة الإرجاع (Return statement): تستخدم لإنهاء تنفيذ استدعاء الدالة وإرجاع قيمة التعبير ويؤدي ظهورها إلى أنهاء تنفيذ أوامر الدالة وإعادة المقطع البرمجي إلى النقطة التي تلي أمر استدعائها مباشرة.

الوسائط الافتراضية (Default arguments):

تستخدم لجعل المعاملات اختيارية واستخدام قيم افتراضية إذا كان المستخدم لا يريد إعطاء قيم لهذه المعاملات، فإذا لم يتم تمرير قيمة أخرى أثناء تسمية الدالة فستأخذ وسائط الدالة قيم الإعداد الافتراضية.

المتغيرات المحلية والعامة (Local and Global Variables):

عند تحديد المتغيرات داخل تعريف الدالة فإنها لا تؤثر ولا تتأثر بالمتغيرات الأخرى التي تحمل نفس الاسم خارج الدالة، المتغيرات داخل الدوال تسمى **متغيرات محلية**، تبدأ من تعريف اسم الدالة وتنتهي عند توقف الدالة.

إذا كنت ترغب في تعيين قيمة لاسم محدد في المستوى الأعلى من البرنامج (ليس ضمن نطاق الدوال أو الفئات) فأنت بحاجة لاستخدام **المتغير العام** ويمكن تعريفه بإضافة كلمة global قبل تعريف المتغير.

العمل مع إكسل وبايثون:

باستخدام بايثون يمكن أتمتة بعض المهام المتكررة وإجراء العمليات الحسابية المعقدة بشكل أسرع من العمل مع إكسل فقط.

المكتبات في لغات البرمجة: هي حزمة مقاطع برمجية مكتوبة سابقاً تودي بعض المهام بحيث لا يحتاج المبرمج إلى كتابة المقطع البرمجي من البداية في كل مرة.

مكتبة أوبين بيكسل (openpyxl): مكتبة يمكن تنزيلها وتثبيتها ويمكن من خلالها إجراء عمليات القراءة والكتابة والحسابات الرياضية وإنشاء الرسوم والمخططات البيانية.

استيراد مكتبة أوبين بيكسل:

لاستخدام مكتبة أوبين بيكسل بعد تثبيتها تحتاج إلى استيرادها ويمكنك القيام بذلك باستخدام أمر الاستيراد (import).

```
# استيراد مكتبة أوبين بيكسل
Import openpyxl
```

العمل مع دفاتر العمل (Workbooks):

يطلق على ملف إكسل اسم **دفتر العمل (workbook)** ويحتوي ملف إكسل على أكثر من ورقة عمل واحدة ويمكن إدخال البيانات ومعالجتها في كل ورقة.

- للعمل على ملف أكسل موجود نستخدم **دالة تحميل دفتر العمل** `load_workbook()`
- لرؤية أسماء الأوراق الموجودة داخل دفتر العمل نستخدم **سمة أسماء أوراق العمل** `sheetnames`.
- لمعرفة عدد السطور نستخدم **سمة السطر الأقصى** `max_row`
- لمعرفة عدد الأعمدة نستخدم **سمة العمود الأقصى** `max_column`
- للحصول على جميع القيم لصف أو أكثر في ورقة عمل إكسل نستخدم دالة `iter_rows`
- للحصول على جميع القيم لعمود أو أكثر في ورقة عمل إكسل نستخدم دالة `iter_cols`

نهاية مقرر المهارات الرقمية للصف الثاني المتوسط
الفصل الدراسي الأول
مع تمنياتي للجميع بالتوفيق