المجموعة (Collection)

هي تخزين عناصر متعددة معًا وتخزينها باسم واحد يسمى متغير.

يوجد أنواع من المجموعة (Collection)

1- القائمة (List):

هي عبارة عن مجموعة عناصر مخزنة معًا في متغير واحد.

يتم كتابتها كالتالي:

```
variable_name = [element,element,..]
```

مثال:

```
colorList = [
        "blue",
        "red",
        "green",
        "black"
]
```

2- القاموس (Dictionary):

يتم استخدام القاموس (Dictionary) لتخزين قيمة (Value) معينة في مفتاح (Key) معين.

يتم كتابة القاموس (Dictionary) كالتالي:

```
variable_name = {
    key : value
}
```

مثال:

```
phoneNumbers = {
    "ali" : "01157082417",
    "ahmed" : "01157082333",
    "alaa" : "01157082444",
    "mohamed" : "01157082555",
}
```

كيفية العمل مع المجموعات؟

القائمة (List) تتكون من Index بداية من الصفر، أي أن أول عنصر في المجموعة (Collection) يكون الـ Index = 0 والذي بعده يكون الـ Index = 1 وهكذا.

لطباعة القائمة (List) نقوم بالتالي:

```
print(list_name[index])
```

مثال:

```
colorList = [
    "blue",
    "red",
    "green",
    "black"
]
print(colorList[0]) #blue
print(colorList[1]) #red
print(colorList[2]) #green
print(colorList[3]) #black
```

—----------

القاموس (Dictionary) يتكون من key: value.

لطباعة القاموس (Dictionary) نقوم بالتالي:

```
print(Dictionary _name[key])
```

مثال:

```
phoneNumbers = {
    "ali" : "01157082417",
    "ahmed" : "01157082333",
    "alaa" : "01157082444",
    "mohamed" : "01157082555",
}
print(phoneNumbers["ali"]) #01157082417
print(phoneNumbers["ahmed"]) #01157082333
print(phoneNumbers["alaa"]) #01157082444
print(phoneNumbers["mohamed"]) #01157082555
```

Iterations

التكرار (Iteration)

هو تكرار نفس الإجراء عدة مرات حتى يصل إلى نقطة نهاية محددة.

الحلقة (Loop)

رمز يتكرر، ينتقل من بداية العملية إلى نهايتها، ثم البدء من جديد.

لعمل الحلقة (Loop) يجب معرفة ثلاثة أشياء:

- المعلومات التي سنعمل عليها.
- ماذا سيحدث مع كل تكرار؟
- متى ستتوقف؟

يجب أن نكون على علم بمتى سوف تتوقف الحلقة (Loop) جيدًا؟ لأنه إذا لم نكتب شرط توقف الحلقة (Loop) سوف يحدث ما يسمى Infinite loop.

For loop

نستخدمها لتكرار عملية معينة عدد مرات معين.

على سبيل المثال إذا كان معنا القائمة - List التالية

```
colorList = [
        "blue",
        "red",
        "green",
        "black"
]
```

إذا اردنا طباعة ما بداخل القائمة - List بدون استخدام الـ For loop سوف نكتب الآتي:

```
print(colorList[0]) #blue
print(colorList[1]) #red
print(colorList[2]) #green
print(colorList[3]) #black
```

أما إذا أردنا طباعة ما بداخل القائمة - List باستخدام الـ For loop سوف نكتب الآتي:

```
for color in colorList:
```

```
print(color)
```

while loop

يتم تكرار تنفيذ الكود الذي بداخل while loop إذا كان الشرط صحيحًا، وفي كل مرة يتم التأكد إذا كان الشرط صحيحًا أم لا، إلى أن يكون الشرط خطأ وفي هذا الوقت تتوقف الـ while loop.

يتم كتابة الـ while loop كالتالي:

```
while condition - الشرط :
    #Code
```

مثال:

```
counter = 5
while counter != 100:
    print(counter)
    counter +=5
```

في هذا المثال سوف يتم طباعة الرقم 5 وكل مرة سوف يزداد الرقم ب 5 فسوف يتم طباعة 5 و 10 و15 و20 و25 ...إلخ، إلى أن يصل إلى الـ 100، وبعد ال 100 سوف تزداد القيمة لكي تصبح 105 لكن لن يتم طباعة ال 105 لأنه لن يتم تنفيذ الكود الذي بداخل ال while loop لأن الشرط خطأ لأن 105 لا تساوي 100 فسوف تتوقف ال while loop.

Using external code

Module

هو ملف بايثون يحتوي على تعليمات برمجية، مثل: المتغيرات أو الوظائف.

من الممكن أن تكتب Module أو تستخدم Module جاهزة.

Libraries

استخدام وحدات متعددة معًا بحيث يتم توزيعها واستخدامها في مجموعة.

من الممكن أن تكتب Libraries أو تستخدم Libraries جاهزة.

Python Libraries:

- PyTorch
- TensorFlow
- Pandas

Framework

عندما لا يتم استخدام مجموعة من التعليمات البرمجية معًا فحسب، بل يتم استخدامها بطريقة معينة.

Python Frameworks:

- Django
- Dash
- Cubic Web
- CherryPy

إنشاء Module

إذا كان هناك جزء من الكود سوف يتم استخدامه في أكثر من ملف، فعلى سبيل المثال:

إذا كانت دالة - Function تتكرر في أكثر من ملف، نقوم بإنشاء Module لها كالآتي:

- نقوم بإنشاء Folder وبداخله ننشأ file نضع به الدالة - Function
- وعندما نريد استعمال هذه الدالة - Function نكتب في الملف المراد استعمال الدالة – Function فيه.

```
from folder_name import file_name
```

لاستخدام Module جاهزة نقوم باستدعائها كالتالي:

```
import Module_name
```

مثال:

```
import math

print(math.floor(10.6))#10
```

ال floor تقوم بإزالة الكسر(أي التقريب إلى أصغر عدد صحيح).

```
print(math.ceil(10.6))#11
```

ال ceil تقوم بإزالة الكسر وتجعل الرقم الصحيح يزيد بواحد (التقريب إلى أكبر عدد صحيح).

Working with strings

جمع السلاسل (Concatenation)

هي دمج سلاسل (Strings) متعددة في سلسلة (String) واحدة.

مثال:

```
name = "Ali"

age = "22"

country = "Egypt"

result = "Welcom " + name + ". Your age is " + age +  ". Your country is "

+ country

print(result )

#Welcom Ali. Your age is 22. Your country is Egypt
```

دوال السلاسل (String functions)

نكتب اسم الـ String وبعد ذلك. اسم الدالة (Function)

String_name.Function_name()

إذا كان معنا String نستطيع عمل الآتي باستخدام الدوال (Functions)

- لجعل أول حرف في الـ String كابيتال - Capital نستخدم ()capitalize
- لجعل الـ String صغير- small نستخدم ()lower
- لمعرفة موقع كلمة معينة في الـ String نستخدم ()find
- لاستبدال كلمة بكلمة أخرى في الـ String نستخدم ()replace
- لكي تقسم الـ String إلى مجموعة أجزاء في قائمة - List نستخدم ()split

---

Planning a program

Style Guide

- مرجع للطرق التي يمكن أن تستعملها لكتابة الكود.
- أشهر Style Guide للبايثون هو PEP 8.
- من أشهر الـ Style Guide للجافا سكريبت هو airbnb.

- أهمية ال Style Guide أنها تساعد على كتابة كود بشكل افضل ومنظم أكثر وبأداء عالي.

Pseudocode

- كتابة وصف لما تحاول القيام به بلغة بسيطة.
- قبل كتابة الكود تكون هناك خطوة وهي حل المشكلة، ال Pseudocode يستخدم في هذه الخطوة، وبعد كتابة ال Pseudocode تستطيع كتابة الكود بلغة البرمجة التي تتقنها.
- يساعد على تحديد إطار لكي لا يكون هناك تشتت.
- يساعد على الوصول لحل المشكلة.

Debugging

التصحيح (Debugging)

يُشير إلى عملية تحديد وإصلاح الأخطاء في برنامج أو نظام معين. يتم التصحيح عن طريق تحليل سلوك البرنامج وتحديد الخطأ الموجود وإصلاحه لضمان عمل البرنامج بشكل صحيح.

حالة الاختبار (Test Cases)

تُشير إلى الأوامر أو النصوص التي يتم تصميمها لاختبار سيناريو معين في برنامج معين. تهدف حالات الاختبار إلى التحقق من أن البرنامج يعمل بشكل صحيح وفقًا للمتطلبات المحددة. يتم تنفيذ حالات الاختبار عن طريق إدخال بيانات معينة أو استخدام أوامر محددة للتحقق من سلوك البرنامج في مختلف السيناريوهات الممكنة.

يوجد ثلاثة أنواع من الأخطاء (Errors)

خطأ في بناء الجملة (Syntax error)

- هذا يحدث عند خرق قواعد اللغة.
- على سبيل المثال إذا كتبنا print(Hello, world) فسوف يحدث هذا الخطأ (Error) وهذا بسبب خرق قاعدة في اللغة وهي عدم وضع Hello, world بين "" .

خطأ أثناء وقت التشغيل (Runtime error)

- هذا يحدث عندما لا يستطيع الكمبيوتر تنفيذ الكود.
- علي سبيل المثال إذا كتبنا (2/0)*10 سوف يحدث هذا الخطأ - error بسبب القسمة على الصفر.

خطأ دلالي (Semantic error)

- هذا يحدث عندما تكون المخرجات - Outputs غير متوقعة.
- على سبيل المثال إذا كتبنا

name = “Ahmed”

print(“Hello, name”)

فهنا يحدث هذا الخطأ - error لأنك كنت تنتظر أن يكون الناتج Hello, Ahmed والناتج الذي ظهر غير متوقع.

Objects

البرمجة الكائنية (OOP)

الكود المنظم باستخدام البرمجة الكائنية - OOP تكون أجزاءه مقسمة إلى أجزاء صغيرة تسمى كائنات - Objects.

كل كائن - Object يكون له أغراض معينة، ودور يقوم به، كل الكائنات - Objects تتواصل مع بعض لكي يعمل البرنامج كوحدة متكاملة.

في البرمجة الكائنية - OOP كل كائن - Object له

- الخصائص - Attributes هي بيانات يحتويها الكائن - Object
- السلوك - Behavior هي التي يستطيع الكائن - Object فعلها.

Advanced topics

تخزين الكمبيوتر (Computer Storage)

يوجد أنواع للتخزين على الكمبيوتر منها:

Drive

يكون عليه البيانات - Data والبرامج - Programs

Memory

يتم تشغيل الكود فيها، وتخزين نواتج أي عمليات حسابية، أو تخزين بيانات مؤقتة.

عند اطفاء الكمبيوتر واعادة تشغيله يتم مسح ما يوجد في ال Memory، ويظل الذي بداخل الـ Drive موجود.

إذا كانت ال Memory مليئة أثناء تشغيل برنامج الكمبيوتر سوف يكون بطئ جدا.

إدارة الذاكرة (Memory Management)

كود يقرر ما يتم الاحتفاظ به في الذاكرة وما يتم التخلص منه.

هذا الكود إذا تمت كتابته خطأ سوف تتسبب في Memory Leak والتي تحدث عندما تزداد المساحة التي يستخدمها البرنامج بشكل غير مبرر.

Garbage Collection

عملية إدارة ذاكرة آلية تتعقب العناصر غير المطلوبة وتقوم بحذفها.

الخوارزمية (Algorithm)

- إجراء يستخدم لحل مشكلة أو إجراء عملية حسابية.
- أو هي خطة مكونة من خطوات منطقية لحل مشكلة أو عمل فكرة معينة.
- عند عمل تطبيق معين أو حل مشكلة معينة، نبدأ بكتابة استراتيجية معينة لحل المشكلة

  هذه الاستراتيجية تسمي الخوارزمية - Algorithm