OBJETS CONNECTÉS

DEV WEB - DIGITAL CAMPUS

INTERVENANT

- Aymeric De Abreu
- Agence de développement d'applications mobile - AppOuest
- mail => ada@appouest.com, mettre en préfix du sujet [DC - DEV]

PROGRAMME

- Introduction au langage Python
- Développement orienté objets connectés

GIT

- Instructions des TPs : https://github.com/aymarick/dc-dev-python-20
- Rendu des TPs sur Github

PYTHON

- Bases syntaxiques du langage
- Programmation orientée objets
- Du web avec Flask

PYTHON

- Type de langage : interprété et haut niveau
- Plateformes : macOS, Linux, Windows
- Plusieurs usages : scripts, développement web, électronique etc...

PARTICULARITÉS

- Séquentiel, fonctionnel et objet
- Typage dynamique fort
- Indentation obligatoire qui délimite les structures.

ENVIRONNEMENT DE DÉVELOPPEMENT

- Utilisation de Python 3.x
- Au choix :
    - En ligne de commande dans le terminal avec un editeur de texte simple (type sublime text)
    - Avec un IDE (PyCharm / VSCode)

VARIABLES

- Crées à la volé
- Utilisation de « = » pour l'affectation
- ex :

```
name = 'John'
```

La variable name contient la chaine de caractère « John »

VARIABLES

- Modifiable
- ex :

```
name = 'John'
name = 'Jane'
```

VARIABLES

- Assignation parallèle :

```
firstname, lastname = 'John', 'Doe'
```

- Assignation multiple :

```
name_a = name_b = 'Jane'
```

- Lecture d'une entrée utilisateur (dans le terminal) :

```
firstname = input('Quel est ton prénom ?')
```

LES TYPES

- Typage dynamique fort
- Dans la plupart des cas on ne spécifie pas le type :

```
 name = 'John' #le type est une chaine de caractère
 age = 32 #le type est un nombre entier
```

- Pas de cast automatique :

```
 print(name + age) #Erreur, age n'est pas une chaine
de caractère
 print(name + str(age)) #Fonctionne ! ici on convertit
age en chaine de caractère.
```

LES TYPES

- bool : True / False (Attention à la majuscule)
- None : Valeur vide / nulle
- Types numériques :
    - int : nombre entier [-2 147 483 648 ; +2 147 483 647]
    - long : nombre entier [-infini, -2 147 483 648] et [+2 147 483 647, +infini]
    - float : nombres à virgule
    - complex : Nombres complexes

LES TYPES

- chaînes :
    - str : chaîne de caractère encodés en UTF-8
    - bytes : chaîne d'octets

LES TYPES

- structures de données :
    - Listes (list) : liste de valeurs

```
values = [3,  5,  'Test', 3.14]
```

    - Dictionnaires : tableau associatifs

```
dic = {'age': 32, 'name': 'John'}
```

LES TYPES

- Connaitre le type d'une variable :

```
print(type(name))
```

LES OPÉRATEURS MATHÉMATIQUES

- + : Addition (et concaténation pour les chaînes de caractères)
- - : Soustraction
- * : multiplication
- / : division
- ** : puissance
- % : modulo (reste de la division entière)
- // : division entière

LES OPÉRATEURS DE COMPARAISON

- == : égalité
- < : Inférieur
- > : Suppérieur
- <= : Inférieur ou égal
- >= : Supérieur ou égal
- != : Différent

LES OPÉRATEURS LOGIQUES

- and : Et logique
- or : Ou logique
- is : Affirmation
- not : Négation (inverse l'expression)
- in : Appartient

LES CONDITIONS

- if, elif et else : Si, Sinon Si, Sinon

- Délimitations des blocs par des indentations !

```
if var1 > var2 :
    #Do something
elif var1 == var2 :
    #...
else :
    #do something else
```

LES BOUCLES

```
while #Condition# :
    print('...')
```

```
for elt in liste :
    print(elt)
```

```
for key, elt in dic :
    print('clé : '+key)
    print('value : '+elt)
```

LES FONCTIONS

- bloc de code réutilisable
- possibilité de passer des paramètres
- Chaque nom de fonction doit être unique.
- Peut retourner une valeur

```
def addition(number1, number2) :
    return number1 + number2
```

LES FONCTIONS

- Possibilité de mettre des paramètres par défaut :

```
def printHello(name = 'World') :
    print('Hello, '+name)
```

LES BIBLIOTHÈQUES

- Certaines bibliothèques sont installées par défaut

```
import os
print(os.name)
```

- D'autres demandes à être installées (par le gestionnaire de librairie Python « Pip »)

```
pip install flask
```

GESTION DES ERREURS

- Possibilité d'intercepter les erreurs

```
age = 'vingt trois'
try:
    age_number = int(age) #erreur
except:
    print('La variable age n\'a pas pu
être converti en <int>')
```