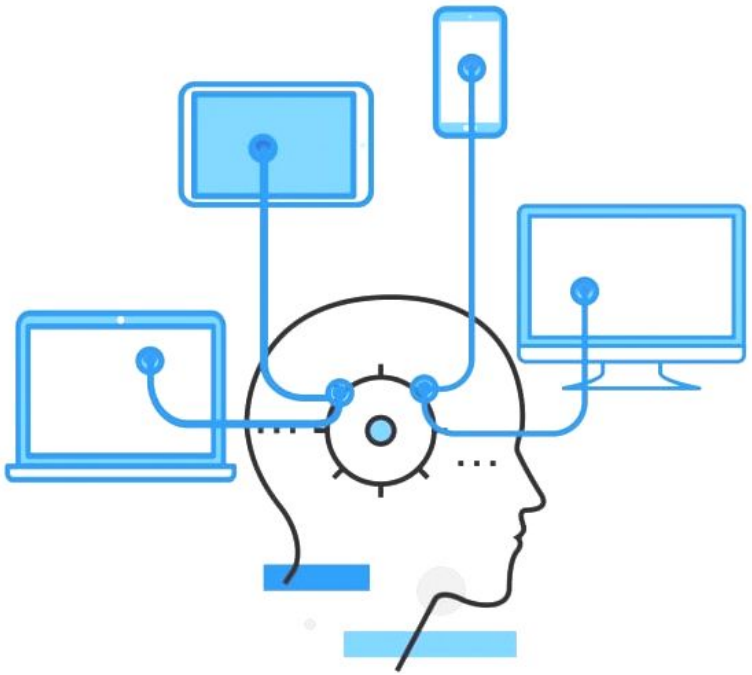


Image: pngtree

Option IA & Programmation

Rappels de syntaxes

Alexandre Mazel

Année 2022-2023

alexandre.zelma@gmail.com

Les variables de type tableau

Exemple de variable contenant un tableau:

```
a = [1,2,3]

prenoms = [“Alex”,”Elsa”,”Gaia”]

prenoms_et_age = [ [“Alex”,13], [“Gaia”,12] ]


# le premier index est le 0 !!!

print(a[0])          # => 1

print(prenoms[2])    # => “Gaia”
```

Les tableaux

```
# comment modifier un tableau


a = [1,8,3]
a[1] = 2          # a vaut [1,2,3]
a[3] = 4          # => out of range (hors du tableau)
len(a)            # retourne 3
a.append(4)       # a = [1,2,3,4]
len(a)            # retourne 4
```

Les tableaux à n dimensions

Exemple de variable contenant un tableau hétérogènes:

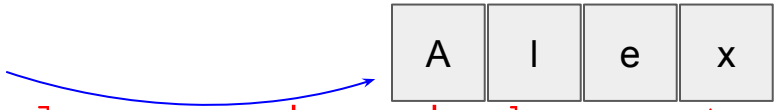
```
a = [ [1,2,3], “Alex”, 18, [100], [4,5,6] ]

prenoms_et_age = [ [“Alex”,13], [“Gaia”,12] ]

bbb = [[[10,20]]]

# le premier index est toujours le 0 !!!

len(a)      # => 5              a[2]                    => 18
a[0]        # => [1,2,3]        prenoms_et_age[0][1] => 13
len(a[0])   # => 3              prenoms_et_age[1][1] => 12
a[0][0]     # => 1              bbb[0][0][1]            => 20
a[1]        # => “Alex”         len(bbb[0][0])          => 2
```

Indices de chaînes de caractères

```
a = "Alex"
```

En mémoire a est en fait un tableau homogène de caractères:

a




```
# le premier index est toujours le 0 !!!
```

```
len(a)      # => 4
a[2]        # => 'e'
a[1:3]      # => 'le'
a[1:]       # => 'lex'
a[:2]       # => 'Al'
a[-1]       # => 'x'
a[-2:]      # => 'ex'
```

```
a[:-1]              => 'Ale'
a[0] + a[1:]        => 'Alex'
a[0]*3 + a[1:]      => 'AAAlex'
a[:]                => 'Alex'
a[::2]              => 'Ae'
a[::-1]             => 'xelA'
~~a[0] = "b" !~~ impossible de changer un caractère dans une chaine.
```

Les fichiers

usage des fichiers:
ecriture
```
file = open("toto.txt", "w")
file.write( "Salut toto!")
file.close()
```

lecture
```
file = open("toto.txt", "r")
s = file.read()
file.close()
print(s)
# autre méthode pratique:
file.readline()  # lit une ligne d'un fichier
```

autour des fichiers

récupérer la liste des fichiers d'un dossier
```
files = os.listdir("c:\\")
```

savoir si un fichier existe
```
is_exist = os.path.isfile("toto.txt")
```

récupérer la taille d'un fichier
```
size = os.path.getsize("toto.txt")
```

Les fonctions

```
# définition des fonctions
def mafonction1(param1,param2 = 1):
     # le parametre2 a une valeur par défaut qui peut etre ommise lors de l'appel

     if param1 > 4:
          return 4
     return param2 # valeur de retour

def myfunc2(parametre1,parametre2):
     val1 = parametre1*parametre2
     val3 = parametre2
     val3 = parametre2*38
     return val1,val2,val3 # la fonction retourne 3 valeurs!

# appel des fonctions
y = mafonction1(3) # le deuxieme parametre n'est pas spécifié
a,b,c = myfunc2(5,6)
```

OpenCV: function sheet

```
# I/O
im = imread(filename)
imwrite(filename, image)
imshow("window_name",image)
waitKey(millisec)

# conversion
cvtColor
resize
rot90
normalize
```

```
# filter
Sobel
CannyEdge

# numpy method
im[y,x] => couleur du pixel en (x,y)

im[:, :, 1] = 0 # efface tout les pixels vert

im = im[10:,:] # efface la bande de 10 pixels supérieur

im[0:4,0:8] = im2 # colle im2 dans le coin de im (im2 doit ici faire 8x4)
```