

Python 2 dans Visual Studio Code

«Open folder...» -> dans un répertoire que vous pourrez appeler «cours2» (ou «chose2», ou ce que vous voulez).

Ouvrir un Terminal dans Visual Studio Code.

Listes (suite + recap)

Créez un nouveau document.
Donnez-lui le nom qui vous plaît.
Inscrivez
`# coding : utf-8`

Créez une **nouvelle liste**.
`listeNb = [22, 34, 40, 55, 66, 73, 82, 91]`

min() et max()

Fonctions pour afficher plus petit ou plus grand élément d'une liste.
`print(min(listeNb))`
`print(max(listeNb))`

Fonctionne aussi avec texte.

.append()

Méthode pour ajouter éléments à liste.
Créer d'abord une liste vide.
`listeNb.append(9999999)`
`print(listeNb, max(listeNb))`

Boucles

Allez chercher à nouveau le document **films.py** de la semaine dernière dans le web:
<http://bit.ly/jhroycinema>

Placez-le dans votre répertoire dans lequel vous travaillez cette semaine et renommez le fichier: **cinema.py**
Ouvrez-le et donnez à l'immense liste qu'il contient le nom de
`films`

Maintenant, créez un nouveau document et baptisez-le **boucles.py**.

Écrivez l'entête
`# encoding: utf-8`

On va utiliser un nouveau truc. On va importer dans boucles.py la liste films contenue dans le fichier cinema.py avec la commande suivante:
`from cinema import films`

Faites, pour vérifier:
`print(films)`

-> Combien de films dans cette liste?

```
print(len(films))
```

-> Quel est le 1001e film dans cette liste?

```
print(films[1000])
```

En informatique, on va très souvent faire des boucles pour traiter des masses d'information un item à la fois. Tous les langages informatiques ont des boucles.

En python, voici la syntaxe.

for variable in liste:
on fait quelque chose avec variable

L'indentation est importante. Sinon, ça ne fonctionne pas. Utilisez tabulateur, ou 2 espaces ou plus.

Donc, ici, on va écrire:

```
for film in films:  
    print(film)
```

Compteur.

Ajoutez une variable **avant la boucle** et donnez-lui la valeur de zéro.

```
n = 0
```

Dans la boucle, augmentez-en la valeur de 1.

```
n = n + 1
```

Mais les pros écrivent ça ainsi:

```
n += 1
```

-= va faire la même chose que **n = n - 1**.

Et imprimez cette variable en même temps que chaque film:

```
print(n, film)
```

.split()

Méthode pour briser des chaînes de caractères en fonction d'un ou de plusieurs caractères donnés.

Par exemple, ici, chaque film contient trois informations intéressantes.

On va briser la variable **film** en trois en fonction du point-virgule.

```
film = film.split(';')
```

La variable film va être transformée en une liste avec autant d'éléments qu'il y a d'occurrences du caractère recherché +1.

-> EXERCICE

Écrivez pour chaque film un **print()** avec deux phrases complètes disant «Le 565e film de la liste est Eldorado. Il a été réalisé par Charles Binamé en 1995. » Utilisez la méthode **.format()**.

-> EXERCICE

Extrairez le nombre de mots de cette chaîne de caractères:

« Idées, mercredi 4 mars 2015 1259 mots, p. A7 ».

Boucle dans un range().

Il est aussi possible de faire des boucles avec un intervalle.

Ici, la syntaxe sera:

```
for i in range(1000,10000):  
    print(i/5)
```

Boucle dans une boucle. <--

Le meilleur exemple est pour générer les codes des médecins.

```
for annee in range(1930,2021):  
    for code in range(1001,2000):  
        permis = str(annee)[2:] + str(code)[1:]  
        print(len(permis),permis)
```

Dictionnaires

Créez-vous un autre document -> **dict.py**

```
# coding : utf-8
```

Allez chercher le fichier **lnh.py** ici:

<http://bit.ly/jhroyhockey>

Placez-le dans le même répertoire que celui dans lequel vous travaillez pour le moment.

On pourrait copier-coller la variable **joueurs**. Qui est une liste de dictionnaires.
Mais on va faire mieux.

```
from lnh import joueurs
```

Écrivez:

```
print(joueurs)
```

Et exécutez votre script dict.py.

Un petit rappel sur la structure des dictionnaires.

Allez chercher le 2000e de ces joueurs:

```
print(joueurs[1999])
```

Souvent on ne veut pas tout. On veut discriminer, faire des choix.

Conditions

if.

La syntaxe est toujours:

if condition:

on fait quoi

Pour n'afficher que les joueurs du Québec.

ATTENTION au double-égal -> en fait le égal simple est une attribution:

```
for joueur in joueurs:
    if joueur["pays"] == "QC":
        print(joueur)
```

Si on veut juste les joueurs québécois qui ne sont pas nés à Montréal?

On va voir le « n'égale pas » -> **!=**

```
for joueur in joueurs:
    if joueur["pays"] == "QC" and joueur["Montréal"] != "Montréal":
        print(joueur)
```

On voit ainsi comment imbriquer deux **ifs** l'un dans l'autre.

Pour faire un « et » -> **and**.

Pour faire un « ou » -> **or**.

Pour trouver joueurs dont c'est l'anniversaire et qui sont actifs:

```
for joueur in joueurs:
    if joueur["jour"] == 23 and joueur["mois"] == 1 and "2019" in joueur["activité"]:
        print(joueur)
```

elif.

Façon de vérifier une autre condition si la précédente n'a pas été remplie. Utile quand on veut travailler sur des variables différentes:

```
joueursQC = 0
joueursON = 0
```

```
for joueur in joueurs:
    if joueur["pays"] == "QC":
        joueursQC += 1
    elif joueur["pays"] == "ON":
        joueursON += 1
```

```
print(joueursON, joueursQC)
```

Clés, valeurs, items des dictionnaires

Pour afficher les clés ou les valeurs d'un dictionnaire, on peut faire une boucle. On va en faire une juste dans celui de Mélanie Joly.

```
for c in élus[34]:  
    print(c)
```

Ici, par défaut, qu'est-ce qui est indiqué?

```
for c in élus[34].keys():  
    print(c)
```

Pour aller chercher toutes les valeurs, on peut utiliser méthode **.values()**:

```
for v in élus[34].values():  
    print(v)
```

Pour itérer dans tout, on utilise la méthode **.items()**:

```
for c,v in élus[34].items():  
    print(c,v)
```

-> faites une boucle qui va imprimer chacun des dictionnaires.

Je vous rappelle que la variable élus est une liste de dictionnaires.

```
for elu in élus:  
    print(elu)
```

Une autre façon d'accéder à des éléments dans un dictionnaire, c'est d'utiliser leur clé.

La clé dans un dictionnaire, c'est comme l'index dans une liste.

Donc si on veut afficher seulement la province des élus, on pourrait entrer ceci:

```
for elu in élus:  
    print(elu['Province'])
```