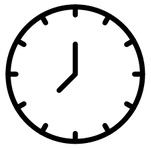
Cours B2

PYTHON 3

Par Kantin FAGNIART

CADRE DE CLASSE



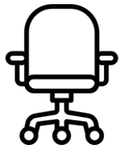
À l’heure en classe

15 minutes de pause

Respect des dates et du format des rendus sur Moodle

Attitude professionnelle (tenue, language, ...) et respect mutuel

Pas de triche et respect des consignes



Respect du matériel et de la salle de classe

Pas de nourriture ni boisson (sauf bouteille fermable)

Par Kantin FAGNIART

Installations

1 Installation Windows

2 Installation Mac

3 Installation d’un IDE

Par Kantin FAGNIART

Installation Windows

Mission :

- ✓ Cliquez sur le lien Download
- ✓ Sélectionnez la version de Python que vous souhaitez utiliser.
- ✓ On vous propose un (ou plusieurs) lien(s) vers une version Windows : sélectionnez celle qui conviendra à votre processeur.
- ✓ Enregistrez puis exécutez le fichier d'installation et suivez les étapes.
- ✓ Une fois l'installation terminée, vous pouvez vous rendre dans le menu Démarrer > Tous les programmes. Python devrait apparaître dans cette liste.
- ✓ Python est installé sous Windows.

Download Python

Par Kantin FAGNIART

Installation MAC OS X

Mission :

- ✓ Téléchargez la dernière version de Python. Ouvrez le fichier .dmg et faites un double-clic sur le paquet d'installation Python.mpkg
- ✓ Un assistant d'installation s'ouvre, laissez-vous guider
- ✓ Python est maintenant installé !

Download Python

Par Kantin FAGNIART

Installation d’un IDE

EDI en français ou IDE pour Integrated Development Environement

Mission :

- ✓ Cliquez sur le lien Download

Download Pycharm

Par Kantin FAGNIART

Les Notions de Base

Par Kantin FAGNIART

Les Notions de Base

Mot	Définition	Mot	Définition	Mot	Définition	Mot	Définition
and	Opérateur ET booléen logique	**elif**	Condition contraire	**if**	Condition	**pass**	
as		**else**	Contraire	**import**	Importation de module	**print**	***Afficher*** ***En python 2 mots cle*** ***En python 3*** **des fonctions du module builtins deviennent un mot réservé**
assert	évaluer une condition donnée	**except**	Sauf (à utiliser après "try")	**in**	Contient	**raise**	
break	Sortie de boucle	**exec**	***En python 2 mots clés*** ***En python 3*** **des fonctions du module builtins deviennent un mot réservé**	**is**	Est	**return**	Stopper la fonction courante (renvoyer sa valeur)
class	Définition de classe d'objet	**finally**		**is not**	N'est pas	**sort**	Classer par ordre alphabétique
continue		**for**	Boucle	**lambda**	Définition d'une fonction Lambda	**try**	Essayer (généralement suivi de "except" : sauf)
def	Définition de fonction	**from**	De	**not**	Négation logique	**while**	Boucle
del	Suppression de	**global**	Définition dans une fonction d'une variable globale	**or**	Opérateur de choix OU booléen logique	**yield**	S'emploie uniquement dans une fonction, et renvoie son résultat régénéré
with							
True	**Nouveau mots clé Python 3**	**False**	**Nouveau mots clé Python**	**None**	**Nouveau mots clé Python**	**nonlocal**	**Nouveau mots clé Python**

Par Kantin FAGNIART

Indentation

```
21      def divide(a, b):  Kantin FAGNIART
22          if a == 0:
23              return 0
24          elif b == 0:
25              return 0
26          else:
27              return a / b
28
```

Symboles

,	()	[]	{}	.	:	Espace
	Tuple	liste	Ensemble dictionnaire			

Les Tuple : non modifiables

```python
a = (1, 'un')

if __name__ == '__main__':
    print(a)
    print(a[0])
    print(a[1])
```

```
(1, 'un')
1
un
```

Les générateurs

```python
g = (i**2 for i in range(10))

if __name__ == '__main__':
    for i in g:
        print(i)
```

```
0
1
4
9
16
25
36
49
64
81
```

Par Kantin FAGNIART

Symboles

,	()	[]	{}	.	:	Espace
	Tuple	liste	Ensemble dictionnaire			

liste

```python
a = [1, 'un', 2, 3, 4, 0, 5]

if __name__ == '__main__':
    print(a)
    print(a[0])
    print(a[1:])
```

```
[1, 'un', 2, 3, 4, 0, 5]
1
['un', 2, 3, 4, 0, 5]
```

Les générateurs

```python
k = [i**2 for i in range(10)]

if __name__ == '__main__':
    print(k)
```

```
[0, 1, 4, 9, 16, 25, 36, 49, 64, 81]
```

Par Kantin FAGNIART

Symboles

,	()	[]	{}	.	:	Espace
	Tuple	liste	Ensemble dictionnaire			

Dictionnaire

```python
dico = {"key": 1, "key2": 2, "key3": "B2 Info"}

if __name__ == '__main__':
    print(dico["key"])
    print(dico["key3"])

```

```
1
B2 Info
```

Par Kantin FAGNIART

Opérateurs arithmétiques

+	-	*	**	/	//	%	~
			puissance	Division réel	Division entier	modulo	~x = (-x)-1

```python
a = 2 + 3
b = 2 - 3
c = 2 * 3
d = 2 ** 3
e = 5 / 2
f = 5 // 2
g = 5 % 2
h = ~5

if __name__ == '__main__':
    print(a)
    print(b)
    print(c)
    print(d)
    print(e)
    print(f)
    print(g)
    print(h)
```

```
5
-1
6
8
2.5
2
1
-6
```

Par Kantin FAGNIART

Opérateurs logiques

<	>	<=	>=	==	!=	not	or	and

```python
a = 5 < 2
b = 5 > 2
c = 5 <= 2
d = 5 >= 2
e = 2 != 2
f = 2 == 2

if __name__ == '__main__':
    print(a)
    print(b)
    print(c)
    print(d)
    print(e)
    print(f)

```

```
False
True
False
True
False
True
```

Par Kantin FAGNIART

Définition des fonctions

def <Nom Fonction > ([parametres]) :

<tabulation> instructions

```python
def Bonjour(message):  2 usages
    return "Bonjour, " + message + " comment allez vous?"

if __name__ == '__main__':
    print(Bonjour("B2 info"))

    Bonjour2 = Bonjour
    print(Bonjour2("B2 info"))
```

```
Bonjour, B2 info comment allez vous?
Bonjour, B2 info comment allez vous?
```

Par Kantin FAGNIART

Classe

```python
class Classe1:  1 usage
    attribut1="Class1"
    attribut2=24

    def methode1(self):  2 usages
        return (self.attribut1, self.attribut2)

    def methode2(self, arg1, arg2):  1 usage
        self.attribut1=arg1
        self.attribut2=arg2
        pass

c = Classe1()

if __name__ == '__main__':
    print(c.methode1())
    c.methode2( arg1: "Instance a",  arg2: 12)
    print(c.methode1())
```

```
('Class1', 24)
('Instance a', 12)
```

Par Kantin FAGNIART

Instruction conditionnelle : if (): else

```python
def negatif(val):  2 usages
    if val < 0:
        return True
    return False

if __name__ == '__main__':
    print(negatif(-10))
    print(negatif(10))
```

```
True
False
```

Par Kantin FAGNIART

Instruction conditionnelle : if (): else

```python
def negatif(val):  1 usage
    if val < 0:
        return True
    return False

def positif(val):  1 usage
    if val > 0:
        return True
    return False

def print_signe(val):  3 usages
    if negatif(val):
        print("Le chiffre %.1f est negatif" %val)
    elif positif(val):
        print("Le chiffre %.1f est positif" %val)
    else:
        print("Le chiffre est nul")
    pass

if __name__ == '__main__':
    print_signe(10)
    print_signe(-5)
    print_signe(0)
```

```
Le chiffre 10.0 est positif
Le chiffre -5.0 est negatif
Le chiffre est nul
```

Par Kantin FAGNIART

for & while

for <variables> in <liste>:
<tabulation> instructions

```
if __name__ == '__main__':
    for i in (2,4,6,8):
        print("%d est un nombre paire" %i)

```

```
2 est un nombre paire
4 est un nombre paire
6 est un nombre paire
8 est un nombre paire
```

while test :
<tabulation> instructions

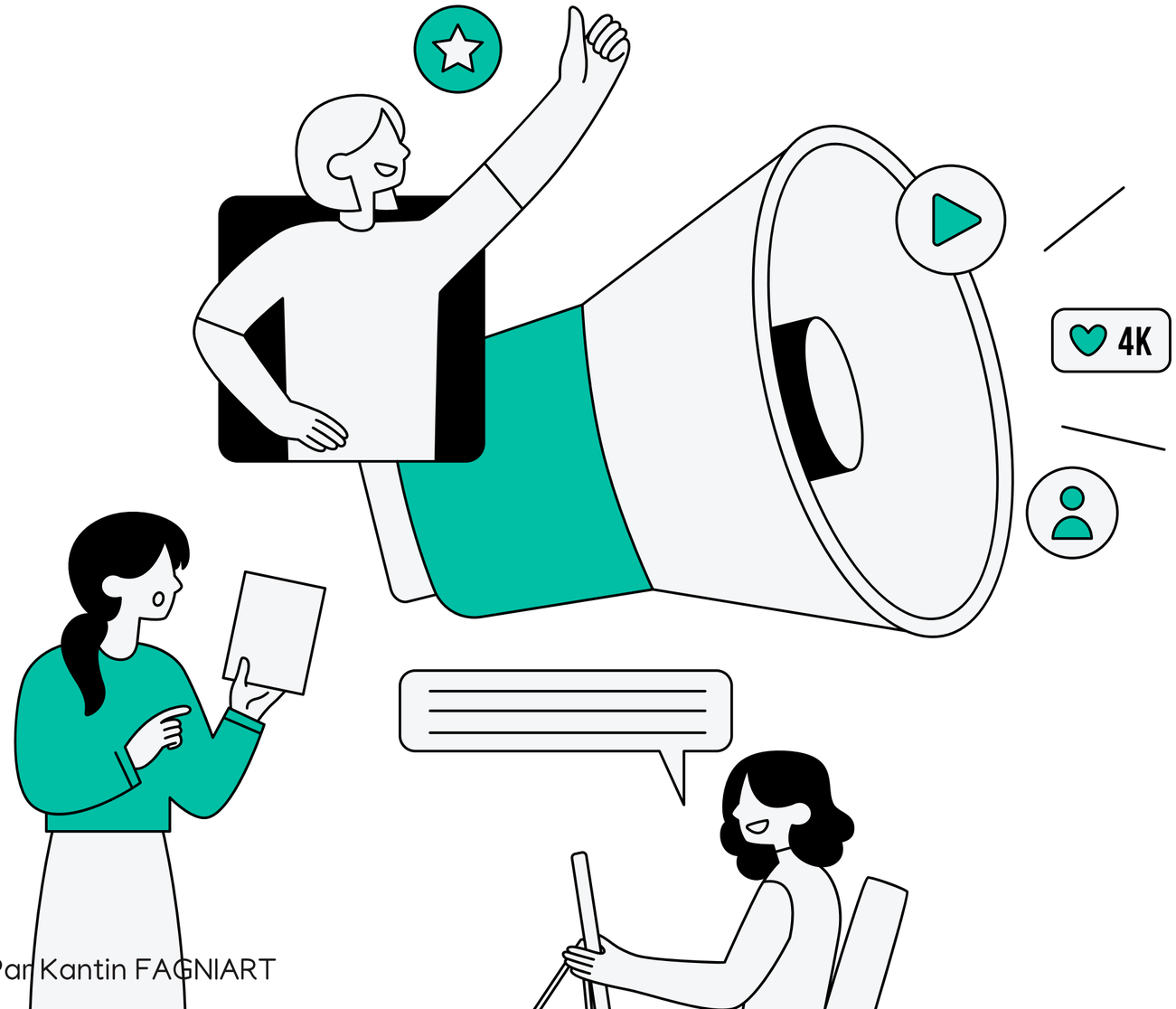
```
i = 0

if __name__ == '__main__':
    while i < 10:
        i = i+2
        print("%d est un nomre paire"%i)

```

```
2 est un nomre paire
4 est un nomre paire
6 est un nomre paire
8 est un nomre paire
10 est un nomre paire
```

Par Kantin FAGNIART



À vous de jouer !

Par Kantin FAGNIART