

Informatique S3 :
Présentation du cours :
Structures de données en Python
Vers l'analyse de données
L2 MIASHS

Laurent Boyer

laurent.boyer@univ-paris1.fr

septembre 2024

3 parties

- ▶ **Structures de données en Python**

5 semaines environ

- ▶ **Un premier programme informatique**

4 semaines environ

- ▶ **Vers l'analyse de données**

3 semaines environ

Objectifs du cours (1ère partie)

- ▶ **Structures de données en Python :**
 - ▶ listes,
 - ▶ dictionnaires,
 - ▶ ensembles,
 - ▶ tuples.

Objectifs du cours (1ère partie)

- ▶ **Structures de données en Python :**
 - ▶ listes,
 - ▶ dictionnaires,
 - ▶ ensembles,
 - ▶ tuples.

- ▶ **En parallèle, des notions fondamentales :**
 - ▶ algorithmique,
 - ▶ complexité,
 - ▶ correction des programmes et tests.

Objectifs du cours (2ème partie)

► **Mini-projet :**

- Un sujet détaillé pour le début,
- avec de multiples prolongements possibles,
- 3 semaines de travail, avec rendu du code et mini-soutenance à la quatrième séance.

Objectifs du cours (2ème partie)

► **Mini-projet :**

- Un sujet détaillé pour le début,
- avec de multiples prolongements possibles,
- 3 semaines de travail, avec rendu du code et mini-soutenance à la quatrième séance.

► **Objectifs :**

- Mettre bout à bout les concepts vu jusque là.
- Travailler en groupe.
- Une étape importante dans l'appréhension de la programmation.

Objectifs du cours (3ème partie)

- ▶ **Vers l'analyse de données :**
 - ▶ Visualisation,
 - ▶ Première approche des outils d'analyse de données,
 - ▶ Introduction au cours de S5.

Objectifs du cours (3ème partie)

- ▶ **Vers l'analyse de données :**
 - ▶ Visualisation,
 - ▶ Première approche des outils d'analyse de données,
 - ▶ Introduction au cours de S5.

- ▶ **Des librairies nécessaires :**
 - ▶ Matplotlib
 - ▶ ...

Organisation du cours

Organisation du cours

- ▶ **12 cours (mercredi 8h-9h) + 12 TD (2h le vendredi)**
qui devraient fonctionner ensemble :
 - ▶ tds en rapport avec le cours,
 - ▶ et reprises de tds en cours si besoin (mail).

Organisation du cours

▶ **12 cours (mercredi 8h-9h) + 12 TD (2h le vendredi)**
qui devraient fonctionner ensemble :

- ▶ tds en rapport avec le cours,
- ▶ et reprises de tds en cours si besoin (mail).

▶ **TD :**

- ▶ 1 feuille/séance, à finir et retravailler chez vous,
- ▶ correction sur l'EPI dès le vendredi 14h.

Organisation du cours

Organisation du cours

- ▶ **Exercices sur l'EPI**
 - ▶ des exercices pour s'entraîner,
 - ▶ avec validation ou correction automatique.

Organisation du cours

▶ Exercices sur l'EPI

- ▶ des exercices pour s'entraîner,
- ▶ avec validation ou correction automatique.

▶ Évaluation

- ▶ trois notes de CC :
 - ▶ 1 note = 4 mini-interros (séances 2,3,4 et 5),
 - ▶ 1 note mini-projet,
 - ▶ 1 note CC type partiel en séance 11 ou 12.
- ▶ partiel en janvier.
- ▶ à propos des mini-interros :
 - ▶ un exo type TD + une question de cours,
 - ▶ sur papier (chatgpt...)

Petit test de rentrée

- ▶ 30mn, **chez vous**, d'ici mardi prochain,
 - ▶ pas noté,
 - ▶ mais obligatoire et individuel,
 - ▶ sur les listes et fonctions.
 - ▶ en ligne.
- ▶ Objectif = évaluer votre niveau (collectivement)
donc ça vaut le coup de le faire sérieusement (c'est à dire seul,
et en cherchant vraiment)
sans assistance type ChatGPT.

Des questions ?

?

Présentation du cours

Re-prise en main

Exemple ex1

```
1 def f(x) :
2     return x+2
3
4 def g(x,y) :
5     return y-x
6
7 def h(x) :
8     x = x+1
9     return ( x, g(x,f(x*2)) )
10
11 a = 3
12 print(a, h(a))
```

Exemple ex1

```
1 def f(x) :  
2     return x+2  
3  
4 def g(x,y) :  
5     return y-x  
6  
7 def h(x) :  
8     x = x+1  
9     return ( x, g(x,f(x*2)) )  
10  
11 a = 3  
12 print(a, h(a))
```



Exemple ex1

```
1 def f(x) :
2     return x+2
3
4 def g(x,y) :
5     return y-x
6
7 def h(x) :
8     x = x+1
9     return ( x, g(x,f(x*2)) )
10
11 a = 3
12 print(a, h(a))
```



3 (4, 6)

Exemple ex2

```
1 for i in range(3) :
2     print(i)
3 for i in range(3) :
4     for y in [4,5] :
5         print(i,y)
```

Exemple ex2

```
1 for i in range(3) :  
2     print(i)  
3 for i in range(3) :  
4     for y in [4,5] :  
5         print(i,y)
```



```
0  
1  
2
```

Exemple ex2

```
1 for i in range(3) :  
2     print(i)  
3 for i in range(3) :  
4     for y in [4,5] :  
5         print(i,y)
```



```
0  
1  
2
```

```
0 4  
0 5  
1 4  
1 5  
2 4  
2 5
```

Exemple ex3

```
1 def max1(l) :
2     m = l[0]
3     for x in l :
4         if x>m :
5             m=x
6     return m
7 def max2(l) :
8     p = 0
9     for x in range(len(l)) :
10        if l[x]>l[p] :
11            p=x
12    return p
13 x = [1,5,4,53,34,-5,12]
14 print(max1(x) == max2(x))
```

Exemple ex3

```
1 def max1(l) :
2     m = l[0]
3     for x in l :
4         if x>m :
5             m=x
6     return m
7 def max2(l) :
8     p = 0
9     for x in range(len(l)) :
10        if l[x]>l[p] :
11            p=x
12    return p
13 x = [1,5,4,53,34,-5,12]
14 print(max1(x) == max2(x))
```



Exemple ex3

```
1 def max1(l) :
2     m = l[0]
3     for x in l :
4         if x>m :
5             m=x
6     return m
7 def max2(l) :
8     p = 0
9     for x in range(len(l)) :
10        if l[x]>l[p] :
11            p=x
12    return p
13 x = [1,5,4,53,34,-5,12]
14 print(max1(x) == max2(x))
```



False

Exemple ex4

```
1 l = [x*x for x in range(5) ]
2 m = [x for x in l if x%2==0 ]
3 n = [x*y for x in m for y in [2,1]]
4 print(len(n),n[2])
5
6 o = []
7 for x in m :
8     for y in [2,1] :
9         o.append(x*y)
10 print(o == n)
```

Exemple ex4

```
1 l = [x*x for x in range(5) ]
2 m = [x for x in l if x%2==0 ]
3 n = [x*y for x in m for y in [2,1]]
4 print(len(n),n[2])
5
6 o = []
7 for x in m :
8     for y in [2,1] :
9         o.append(x*y)
10 print(o == n)
```



Exemple ex4

```
1 l = [x*x for x in range(5) ]
2 m = [x for x in l if x%2==0 ]
3 n = [x*y for x in m for y in [2,1]]
4 print(len(n),n[2])
5
6 o = []
7 for x in m :
8     for y in [2,1] :
9         o.append(x*y)
10 print(o == n)
```



6 8

Exemple ex4

```
1 l = [x*x for x in range(5) ]
2 m = [x for x in l if x%2==0 ]
3 n = [x*y for x in m for y in [2,1]]
4 print(len(n),n[2])
5
6 o = []
7 for x in m :
8     for y in [2,1] :
9         o.append(x*y)
10 print(o == n)
```



6 8

True

► à partir de là on peut aller plus loin.

Exemple ex5

```
1 l = [0,0,0]
2 m = [1,1,1]
3 m[0][2] = 5
4 print(m)
```

Exemple ex5

```
1 l = [0,0,0]
2 m = [1,1,1]
3 m[0][2] = 5
4 print(m)
```



Exemple ex5

```
1 l = [0,0,0]
2 m = [1,1,1]
3 m[0][2] = 5
4 print(m)
```



```
[[0, 0, 5], [0, 0, 5], [0, 0, 5]]
```

Exemple ex6

```
1 l = [0,0,0]
2 m = [l[:],l[:],l[:]]
3 m[0][2] = 5
4 print(m)
```

Exemple ex6

```
1 l = [0,0,0]
2 m = [l[:],l[:],l[:]]
3 m[0][2] = 5
4 print(m)
```



Exemple ex6

```
1 l = [0,0,0]
2 m = [l[:],l[:],l[:]]
3 m[0][2] = 5
4 print(m)
```



```
[[0, 0, 5], [0, 0, 0], [0, 0, 0]]
```

Exemple ex7

```
1 def f(a,b):
2     a[0] = 7
3     b = [0,1]
4     print(a,b)
5
6 x = [1, 2]
7 y = [1, 2]
8 f(x,y)
9 print(x,y)
```

Exemple ex7

```
1 def f(a,b):
2     a[0] = 7
3     b = [0,1]
4     print(a,b)
5
6 x = [1, 2]
7 y = [1, 2]
8 f(x,y)
9 print(x,y)
```



Exemple ex7

```
1 def f(a,b):
2     a[0] = 7
3     b = [0,1]
4     print(a,b)
5
6 x = [1, 2]
7 y = [1, 2]
8 f(x,y)
9 print(x,y)
```



```
[7, 2] [0, 1]
[7, 2] [1, 2]
```

Exemple ex8

```
1 def somme(l) :
2     m = []
3     for i,x in enumerate(l) :
4         m.append(i)
5         m.append(x)
6     return m
7
8 print(sum(somme([1,2,3])))
```

Exemple ex8

```
1 def somme(l) :
2     m = []
3     for i,x in enumerate(l) :
4         m.append(i)
5         m.append(x)
6     return m
7
8 print(sum(somme([1,2,3])))
```



Exemple ex8

```
1 def somme(l) :
2     m = []
3     for i,x in enumerate(l) :
4         m.append(i)
5         m.append(x)
6     return m
7
8 print(sum(somme([1,2,3])))
```



9

Exemple ex9

```
1 liste1 = [7,1,2,4,5,4,2,1,3,7]
2 m = len(liste1)
3 found = False
4 while m>0 and not found:
5     for i in range(len(liste1)-m):
6         if liste1[i:i+m] == liste1[i+m-1:i-1:-1]:
7             found = True
8             result = liste1[i:i+m]
9     m -= 1
10 print(result)
```

Exemple ex9

```
1 liste1 = [7,1,2,4,5,4,2,1,3,7]
2 m = len(liste1)
3 found = False
4 while m>0 and not found:
5     for i in range(len(liste1)-m):
6         if liste1[i:i+m] == liste1[i+m-1:i-1:-1]:
7             found = True
8             result = liste1[i:i+m]
9     m -= 1
10 print(result)
```



Exemple ex9

```
1 liste1 = [7,1,2,4,5,4,2,1,3,7]
2 m = len(liste1)
3 found = False
4 while m>0 and not found:
5     for i in range(len(liste1)-m):
6         if liste1[i:i+m] == liste1[i+m-1:i-1:-1]:
7             found = True
8             result = liste1[i:i+m]
9     m -= 1
10 print(result)
```



[1, 2, 4, 5, 4, 2, 1]

Travail chez vous

► Vous avez sur l'EPI

- Ces slides,
- Les exemples dans une archive zip,
- Les notes du cours de Fabrice Rossi (S2 il y a trois ans déjà).

Travail chez vous

► Vous avez sur l'EPI

- Ces slides,
- Les exemples dans une archive zip,
- Les notes du cours de Fabrice Rossi (S2 il y a trois ans déjà).

► **Nécessité de réviser pour le TD** : "listes et fonctions".
Reprendre les exemples du cours, les tester, les comprendre.

Travail chez vous

► Vous avez sur l'EPI

- Ces slides,
- Les exemples dans une archive zip,
- Les notes du cours de Fabrice Rossi (S2 il y a trois ans déjà).

► **Nécessité de réviser pour le TD** : "listes et fonctions".
Reprendre les exemples du cours, les tester, les comprendre.

► **Nécessité de tester chez vous les exemples du cours**
donc avoir un ordinateur sur lequel vous avez un environnement de travail Python, pour vous entraîner.
C'est indispensable. Venez discuter si quelque chose coince.

La semaine prochaine, des nouveautés !

► Structures de données :

- ensembles,
- dictionnaires.