



Bases de la programmation impérative (BPI)

CM2 : boucles, complexité, types abstraits et structures de données

Manuel Selva



Quel est votre ressenti ?

- Comment trouvez vous le rythme des TD ?
 - trop lent, bon, trop rapide



Quel est votre ressenti ?

- Comment trouvez vous le rythme des TD ?
 - trop lent, bon, trop rapide
- Qui n'a pas d'environnement Python sur une machine personnelle ?



Quel est votre ressenti ?

- Comment trouvez vous le rythme des TD ?
 - trop lent, bon, trop rapide
- Qui n'a pas d'environnement Python sur une machine personnelle ?
- Quel est votre environnement de travail pour les TP ?
 - VM ensimag, noVnc, Linux, Windows, Windows + WSL



Quel est votre ressenti ?

- Comment trouvez vous le rythme des TD ?
 - trop lent, bon, trop rapide
- Qui n'a pas d'environnement Python sur une machine personnelle ?
- Quel est votre environnement de travail pour les TP ?
 - VM ensimag, noVnc, Linux, Windows, Windows + WSL
- Comment trouvez vous le rythme des TP ?
 - trop lent, bon, trop rapide



Sommaire du jour

Séquences et boucles

Combien ça coûte ?

Premier exemple

Problème

Explications

SDD et ADT

À retenir



Jusqu'à présent

- Des scalaires, des chaînes de caractère de petite taille et des tuples de petite taille
- Pas de conteneur de taille variable
- Pas de `list` (sans **E**, donc c'est le type Python)
- Pas de boucle



Qu'est-ce qu'une séquence Python ?

“These represent finite ordered sets indexed by non-negative numbers.

The built-in function `len()` returns the number of items of a sequence.

When the length of a sequence is n , the index set contains the numbers $0, 1, \dots, n-1$.

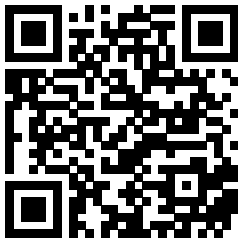
Item i of sequence `a` is selected by `a[i]`.”¹

1. <https://docs.python.org/3.6/reference/datamodel.html#the-standard-type-hierarchy>



Quelles séquences connaissez vous ?

Scannez le QR code ou entrez l'adresse à la main dans votre navigateur.



<https://bvote.ensimag.fr/#/student/selvama>



Les séquences Python

- chaînes de caractères : `s = "BPI"; s[2]`



Les séquences Python

- chaînes de caractères : `s = "BPI"; s[2]`
- tuples : `t = ("C", 1, "peu"); t[1]`



Les séquences Python

- chaînes de caractères : `s = "BPI"; s[2]`
- tuples : `t = ("C", 1, "peu"); t[1]`
- list : `l = ["intéressant", "non?"]; l[0]`



Les séquences Python

- chaînes de caractères : `s = "BPI"; s[2]`
- tuples : `t = ("C", 1, "peu"); t[1]`
- list : `l = ["intéressant", "non?"]; l[0]`
- et deux autres dont on ne parlera (peut être) pas



Comment parcourir une séquence ?

À l'ancienne : ok dans tous les langages impératifs

```
i = 0
while i < len(seq):
    elem = seq[i]
    print(elem)
    i += 1
```



Comment parcourir une séquence ?

À l'ancienne : ok dans tous les langages impératifs

```
i = 0
while i < len(seq):
    elem = seq[i]
    print(elem)
    i += 1
```

```
for i in range(len(seq)):
    elem = seq[i]
    print(elem)
```



Comment parcourir une séquence ?

À l'ancienne : ok dans tous les langages impératifs

```
i = 0
while i < len(seq):
    elem = seq[i]
    print(elem)
    i += 1
```

```
for i in range(len(seq)):
    elem = seq[i]
    print(elem)
```

De façon "Pythonic" : ok en Python

```
for elem in seq:
    print(elem)
```




Sommaire du jour

Séquences et boucles

Combien ça coûte ?

Premier exemple

Problème

Explications

SDD et ADT

À retenir



Qui dit boucle ...

... dit programme éventuellement "**complexe**"



Qui dit boucle ...

... dit programme éventuellement "**complexe**"

Combien d'instructions machines pour ce programme sans boucle ?

```
a = 17
```

```
b = a * 42
```

```
... # avec 3767 lignes sans boucle
```

```
... # ni appel de fonction
```



Qui dit boucle ...

... dit programme éventuellement "**complexe**"

Combien d'instructions machines pour ce programme sans boucle ?

```
a = 17
b = a * 42
... # avec 3767 lignes sans boucle
... # ni appel de fonction
```

Combien d'instructions machines pour ce programme avec boucle ?

```
sum = 0
for i in range(len(seq)):
    elem = seq[i]
    sum += elem
```

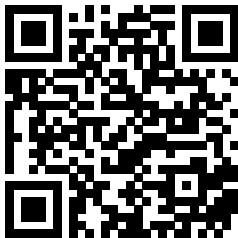


Organisation pour la suite du cours

- ~10mn pour résoudre **individuellement** un problème et voter
- ~10mn pour débattre
- ~5mn pour échanger en **petit groupe** et voter à nouveau
- ~5mn pour débattre à nouveau
- ~20mn pour comprendre la bonne réponse

Le problème à résoudre

Scannez le QR code ou entrez l'adresse à la main dans votre navigateur.



<https://bvote.ensimag.fr/#/student/selvama>



Combien d'instructions machines pour do_complex_stuff?

```
1  from random import randint    16
2  import time                   17
3
4  def is_even(integr):          18
5      return bool(integr % 2)   19
6
7  def do_complex_stuff(nb):     20
8      r = ["MIDDLE"]           21
9      for _ in range(nb):       22
10         i = randint(0, nb)    23
11         if is_even(i):        24
12             r.insert(0, i)     25
13         else:                  26
14             r.insert(len(r),i)27
15     return r                   28
```

```
16
17 if __name__ == "__main__":
18     n = input("Enter a number:\n")
19     s = time.time();
20     l = do_complex_stuff(int(n));
21     e = time.time()
22     print("1. It took me", e - s)
23     f = open("out.data", "w")
24     s2 = time.time()
25     for elem in l:
26         print(elem, file=f)
27     e2 = time.time()
28     f.close()
29     print("2. It took me", e2 - s2)
```



Combien d'instructions machines pour `do_complex_stuff` ?

Les propositions

- A. $100 * (nb * x)$ avec x qui dépend de la fonction `randint`
- B. un certain nombre qui ne dépend pas de `nb`
- C. $100 * (nb * x)$ avec x correspondant au boulot de `randint` + `is_even` + `insert` et $x < 1000$
- D. $100 * (nb * x)$ avec x qui dépend de la lune
- E. le code est incorrect, donc la question n'a pas de sens
- F. plus que toutes les autres réponses numériques
- G. il manque des données pour répondre à la question

Analysons ce programme

```
1  from random import randint    16
2  import time                    17
3
4  def is_even(integr):          18
5      return bool(integr % 2)   19
6
7  def do_complex_stuff(nb):     20
8      r = ["MIDDLE"]           21
9      for _ in range(nb):       22
10         i = randint(0, nb)    23
11         if is_even(i):        24
12             r.insert(0, i)     25
13         else:                 26
14             r.insert(len(r),i)27
15     return r                  28
```

```
16
17  if __name__ == "__main__":
18      n = input("Enter a number:\n")
19      s = time.time();
20      l = do_complex_stuff(int(n));
21      e = time.time()
22      print("1. It took me", e - s)
23      f = open("out.data", "w")
24      s2 = time.time()
25      for elem in l:
26          print(elem, file=f)
27      e2 = time.time()
28      f.close()
29      print("2. It took me", e2 - s2)
```



La bonne réponse

Les propositions

- A. $100 * (nb * x)$ avec x qui dépend de la fonction `randint`
- B. un certain nombre qui ne dépend pas de nb
- C. $100 * (nb * x)$ avec x correspondant au boulot de `randint + is_even + insert` et $x < 1000$
- D. $100 * (nb * x)$ avec x qui dépend de la lune
- E. le code est incorrect, donc la question n'a pas de sens
- F. **plus que toutes les autres réponses numériques**
- G. il manque des données pour répondre à la question



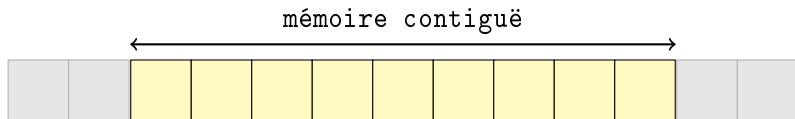
Comment améliorer les choses ?

```
1 from random import randint      19
2 import time                      20
3                                  21
4 def is_even(integr):             22
5     return bool(integr % 2)      23
6                                  24
7 def do_complex_stuff(nb):        25
8     ev = []                      26
9     odd = []                    27
10    for _ in range(nb):          28
11        r = randint(0, nb)       29
12        if is_even(r):           30
13            ev.insert(len(ev),   31
14                       r)        32
15    else:                         33
16        odd.insert(len(odd),    34
17                   r)           35
18    result = list(ev)
```

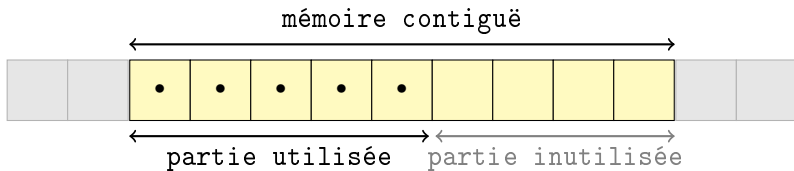
```
    result.append("MIDDLE")
    result.extend(odd)
    return result

if __name__ == "__main__":
    n = input("Enter a number:\n")
    s = time.time();
    l = do_complex_stuff(int(n));
    e = time.time()
    print("1. It took me", e - s)
    f = open("out-fast.data", "w")
    s2 = time.time()
    for elem in l:
        print(elem, file=f)
    e2 = time.time()
    f.close()
    print("2. It took me", e2 - s2)
```

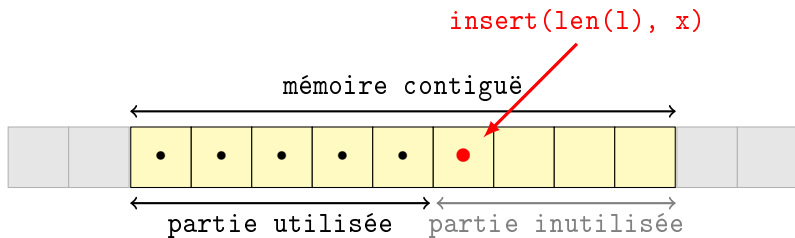
Qu'est-ce qu'une list Python ?



Qu'est-ce qu'une list Python ?

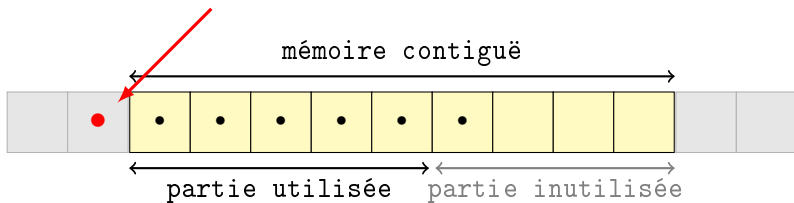


Qu'est-ce qu'une list Python ?



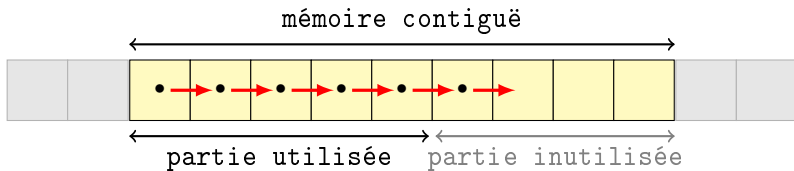
Qu'est-ce qu'une list Python ?

?? insert(0,x) ??



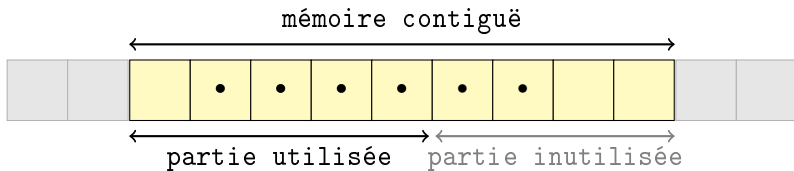
Qu'est-ce qu'une list Python ?

`insert(0, x)`



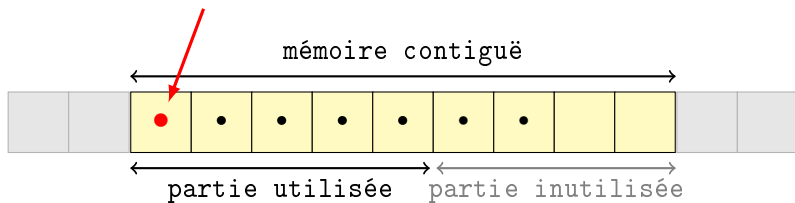
Qu'est-ce qu'une list Python ?

`insert(0, x)`



Qu'est-ce qu'une list Python ?

`insert(0, x)`





Sommaire du jour

Séquences et boucles

Combien ça coûte ?

Premier exemple

Problème

Explications

SDD et ADT

À retenir



SDD et ADT : définitions

Abstract Data Type (ADT)

Spécification d'un ensemble de données et de l'ensemble des opérations qu'on peut effectuer sur elles.



SDD et ADT : définitions

Abstract Data Type (ADT)

Spécification d'un ensemble de données et de l'ensemble des opérations qu'on peut effectuer sur elles.

Structure de données (SDD)

Mise en œuvre concrète d'un ADT.

La notion de coût d'une opération est associée uniquement à une SDD.



Revenons sur la list Python

list est une SDD

- car c'est une mise en œuvre concrète
- plutôt nommée **tableau dynamique** ou **vecteur** en dehors de la communauté Python



Revenons sur la list Python

list est une SDD

- car c'est une mise en œuvre concrète
- plutôt nommée **tableau dynamique** ou **vecteur** en dehors de la communauté Python

qui implémente les ADT

- liste (avec un e)
 - ajouts au début, au milieu et en fin
 - suppression au début, au milieu et en fin
- tableau
 - accès par index entier
 - modification de l'élément à un index donné



Attention à la terminologie

Il n'y a pas de terminologie universelle et souvent ADT et SDD sont confondus (comme en Python).



Attention à la terminologie

Il n'y a pas de terminologie universelle et souvent ADT et SDD sont confondus (comme en Python).

Il faut donc toujours s'interroger pour savoir **de quoi parle-t-on**.



Attention à la terminologie

Il n'y a pas de terminologie universelle et souvent ADT et SDD sont confondus (comme en Python).

Il faut donc toujours s'interroger pour savoir **de quoi parle-t-on**.

En BPI on se référera à :

https://bpi-etu.pages.ensimag.fr/2-iterations/adt_sdd.pdf



Sommaire du jour

Séquences et boucles

Combien ça coûte ?

Premier exemple

Problème

Explications

SDD et ADT

À retenir



À retenir

- Il faut des boucles pour parcourir des séquences
- Avec des boucles on peut écrire des programmes complexes
- ⚠ Ne pas faire n'importe quoi avec des `list` Python ⚠
- On va apprendre à maîtriser les ADT et les SDD élémentaires