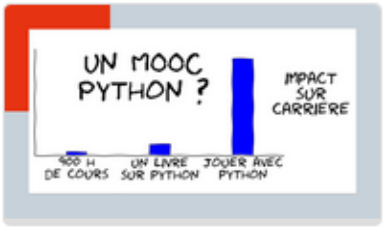
Découvrir et approfondir python

Types construits séquences, listes :

V W2-03 Les séquences.mp4
V W2-04 Les listes.mp4

Vidéos proposées par l'INRIA sur FUN MOOC

Python 3 : des fondamentaux aux concepts avancés du langage



Thierry PARMENTELAT et Arnaud Legoud

Pour bien comprendre et approfondir :

Cocher les affirmations exactes sur les séquences :

O Une séquence de n éléments contient n éléments indicés de 1 à n

O Une séquence peut contenir aucun élément

O Les séquences sont ordonnées

O Une séquence de n éléments contient n éléments indicés de 0 à n-1

O Les nombres entiers sont des séquences

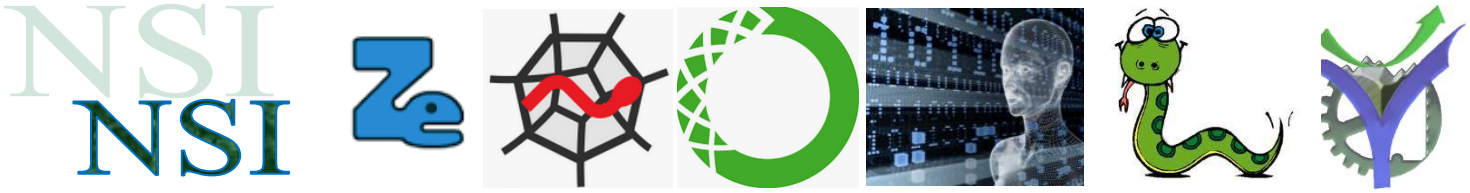
O Les séquences contiennent une relation d'ordre

Accès aux éléments individuels d'une séquence, slicing. Cocher les affirmations correctes :

Nous avons défini une chaine de caractère s qui contient la chaine ci-dessous :

L e s t y p e s c o n s t r u i t s

- Que me sélectionne s[10] :
  O ' ' **O** 'c' **O** 'o'
- La longueur de la chaine est donné par la fonction built-in :
  O s.len **O** s.len() **O** len(s)
- Pour sélectionner uniquement 'types' il faut faire :
  O s[4 :9] **O** s[4 :8] **O** s[-14 :-9] **O** s[-16 :-11]
- Que donne l'instruction s.count('s') :
  O 2 **O** 4 **O** 6



Cocher les affirmations exactes sur les listes :

O Une liste ne peut contenir que des éléments d'un même type

O Une liste est un objet mutable (modifiable)

O Une liste est une séquence

O Je peux utiliser des opérations de slicing sur les listes

O Deux listes sont définies `s1 = [1,2,3,4,5]` et `s2 = [6,7,8,9,10]` je peux les regrouper avec l'opérateur +

O J'ai une liste dénommée ma_liste, Je peux appliquer la méthode de tri ma_liste.sort() sur cette liste quelque soit les types d'éléments contenus sans provoquer d'erreurs

O Une liste peut être triée sans que l'interpréteur python en face une copie temporaire.

Opérations sur les listes :

Voilà une liste : `ma_liste = [1,'Bonjour',45.38,'le',True,'monde']`

- Que me sélectionne `>>> ma_liste[10]`

  O rien None **O** une exception index out of range **O** 'monde'

- Que me sélectionne `>>> s = ma_liste[1::2]`

  O [1,'Bonjour',45.38] **O** rien **O** ['Bonjour','le','monde']

- Que contient chaine après l'instruction `>>> chaine = " ".join(s)`

  O Bonjourlemonde **O** Bonjour le monde **O** 'Bonjour le monde'

Utilisons les listes :

Voilà une séquence d'opérations sur les listes :

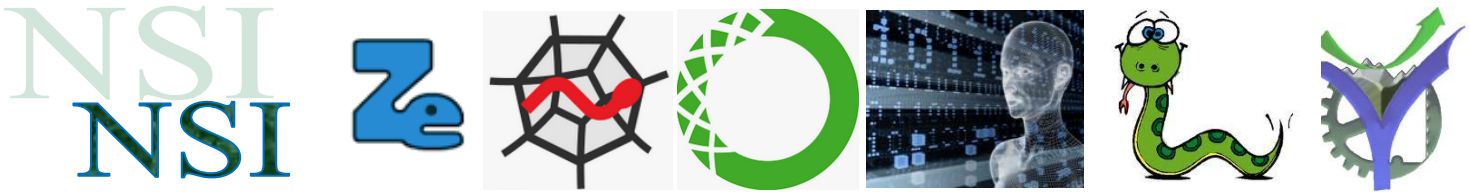
```
>>> ma_liste = [1,'Bonjour',45.38,'le',True,'monde']
>>> ma_liste[2:5]=['le','nouveau']
>>> del ma_liste[0]
```

- Que contient ma_liste après l'exécution de la séquence :

  O [1,'Bonjour',45.38,'le',True]

  O ['Bonjour','le','nouveau','monde']

  O [1,'Bonjour',45.38,'le',True,'monde','nouveau','monde']



Corrigé :

Cocher les affirmations exactes sur les séquences :

O Une séquence de n éléments contient n éléments indicés de 1 à n

O Une séquence peut contenir aucun élément

O Les séquences sont ordonnées

O Une séquence de n éléments contient n éléments indicés de 0 à n-1

O Les nombres entiers sont des séquences

O Les séquences contiennent une relation d'ordre

Accès aux éléments individuels d'une séquence, slicing :

Indices positifs	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19
	L	e	s		t	y	p	e	s		c	o	n	s	t	r	u	i	t	s
Indices négatifs	-20	-19	-18	-17	-16	-15	-14	-13	-12	-11	-10	-9	-8	-7	-6	-5	-4	-3	-2	-1

Cocher les affirmations correctes :

- Que me sélectionne s[10] :

  O ' ' O 'c' O 'o'

- La longueur de la chaine est donné par la fonction built-in :

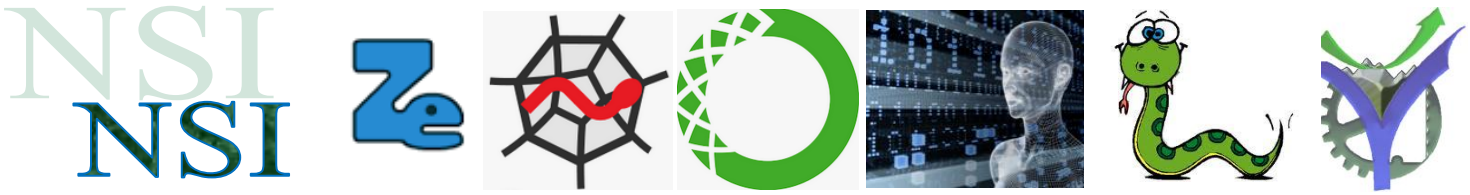
  O s.len O s.len() O len(s)

- Pour sélectionner uniquement 'types' il faut faire :

  O s[4 :9] O s[4 :8] O s[-14 :-9] O s[-16 :-11]

- Que donne l'instruction s.count('s') :

  O 2 O 4 O 6



Cocher les affirmations exactes sur les listes :

O Une liste ne peut contenir que des éléments d'un même type

O Une liste est un objet mutable (modifiable)

O Une liste est une séquence

O Je peux utiliser des opérations de slicing sur les listes

O Deux listes sont définies `s1 = [1,2,3,4,5]` et `s2 = [6,7,8,9,10]` je peux les regrouper avec l'opérateur +

O J'ai une liste dénommée ma_liste, Je peux appliquer la méthode de tri ma_liste.sort() sur cette liste quelque soit les types d'éléments contenus sans provoquer d'erreurs

O Une liste peut être triée sans que l'interpréteur python en face une copie temporaire.

Utilisons les listes :

Voilà une séquence d'opérations sur les listes :

```
>>> ma_liste = [1,'Bonjour',45.38,'le',True,'monde']

>>> ma_liste[2:5]=['le','nouveau']

>>> del ma_liste[0]
```

- Que contient ma_liste après l'exécution de la séquence :
  - **O** [1,'Bonjour',45.38,'le',True]
  - **O** ['Bonjour','le','nouveau','monde']
  - **O** [1,'Bonjour',45.38,'le',True,'monde','nouveau','monde']

```
>>> ma_liste
[1, 'Bonjour', 45.38, 'le', True, 'monde']

>>> s = ma_liste[1::2]

>>> s
['Bonjour', 'le', 'monde']

>>> chaine = " ".join(s)

>>> chaine
'Bonjour le monde'

>>>
```

```
>>> ma_liste = [1,'Bonjour',45.38,'le',True,'monde']

>>> ma_liste[2:5]=['le','nouveau']

>>> del ma_liste[0]

>>> ma_liste
['Bonjour', 'le', 'nouveau', 'monde']
```