C12 Les tableaux

Les listes de Python

- Les listes de Python sont des structures contenant zéro, une ou plusieurs valeurs.

Exemples

C12 Les tableaux

Les listes de Python

- Les listes de Python sont des structures contenant zéro, une ou plusieurs valeurs.
- Une liste se note entre crochets : **[** et **]**

Exemples

C12 Les tableaux

Les listes de Python

- Les listes de Python sont des structures contenant zéro, une ou plusieurs valeurs.
- Une liste se note entre crochets : **[** et **]**
- Les élements sont séparés par des virgules

Exemples

C12 Les tableaux

Les listes de Python

- Les listes de Python sont des structures contenant zéro, une ou plusieurs valeurs.
- Une liste se note entre crochets : **[** et **]**
- Les élements sont séparés par des virgules

Exemples

Une liste `main` qui contient les noms des cinq doigts :

C12 Les tableaux

Les listes de Python

- Les listes de Python sont des structures contenant zéro, une ou plusieurs valeurs.
- Une liste se note entre crochets : **[** et **]**
- Les élements sont séparés par des virgules

Exemples

Une liste `main` qui contient les noms des cinq doigts :

```
main = ["pouce","index","majeur","annulaire","auriculaire"]
```

C12 Les tableaux

Les listes de Python

- Les listes de Python sont des structures contenant zéro, une ou plusieurs valeurs.
- Une liste se note entre crochets : **[** et **]**
- Les élements sont séparés par des virgules

Exemples

Une liste `main` qui contient les noms des cinq doigts :

```
main = ["pouce","index","majeur","annulaire","auriculaire"]
```

Une liste `l` contenant un unique élément : 12

```
l = [12]
```

C12 Les tableaux

Indice d'un élément

- Les éléments d'une liste sont repérés par leur position dans la liste, on dit leur indice

C12 Les tableaux

Indice d'un élément

- Les éléments d'une liste sont repérés par leur position dans la liste, on dit leur indice
- Attention, la numérotation commence à zéro, l'indice du premier élément de la liste est donc zéro

C12 Les tableaux

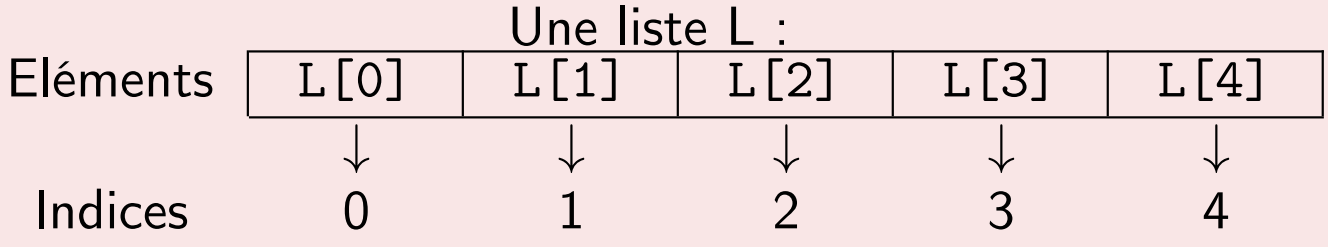
Indice d'un élément

- Les éléments d'une liste sont repérés par leur position dans la liste, on dit leur indice
- Attention, la numérotation commence à zéro, l'indice du premier élément de la liste est donc zéro
- On peut accéder à un élément en indiquant le nom de la liste puis l'indice de cet élément entre crochet

C12 Les tableaux

Indice d'un élément

- Les éléments d'une liste sont repérés par leur position dans la liste, on dit leur indice
- Attention, la numérotation commence à zéro, l'indice du premier élément de la liste est donc zéro
- On peut accéder à un élément en indiquant le nom de la liste puis l'indice de cet élément entre crochet
- L'erreur `IndexError` indique qu'on tente d'accéder à un indice qui n'existe pas.



Exemple

On considère la liste de prénoms suivants :

```
prenoms = ["Alex","Pierre","Marie","Jimmy","Elise"]
```

- Compléter le schéma suivant permettant de représenter cette liste

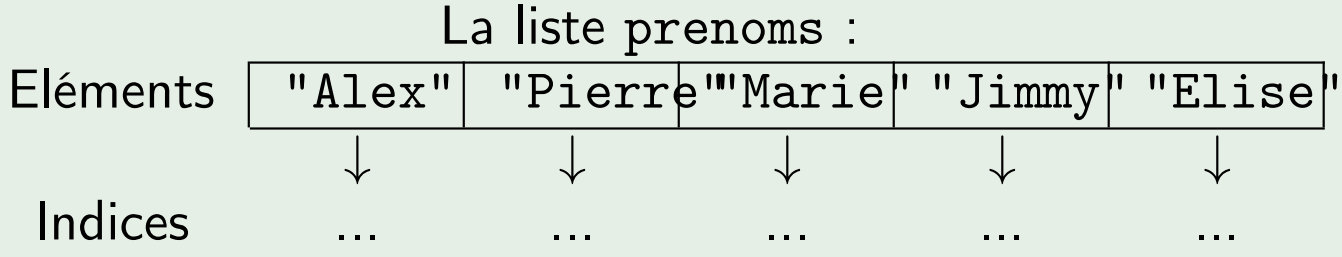
C12 Les tableaux

Exemple

On considère la liste de prénoms suivants :

```
prenoms = ["Alex","Pierre","Marie","Jimmy","Elise"]
```

- Compléter le schéma suivant permettant de représenter cette liste



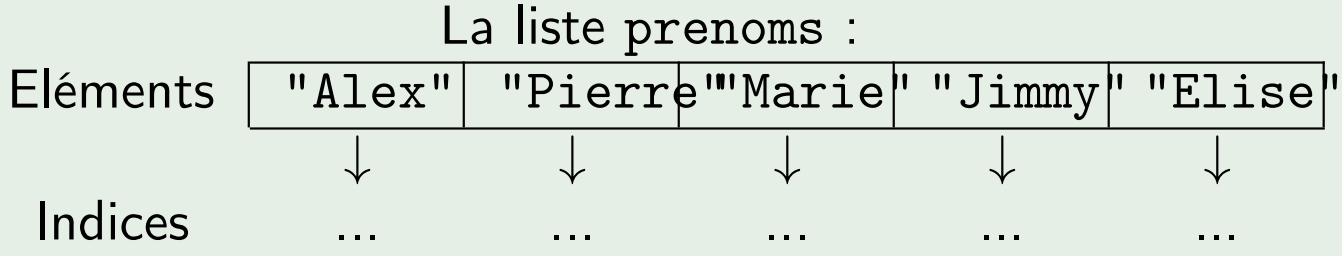
C12 Les tableaux

Exemple

On considère la liste de prénoms suivants :

```
prenoms = ["Alex","Pierre","Marie","Jimmy","Elise"]
```

- Compléter le schéma suivant permettant de représenter cette liste



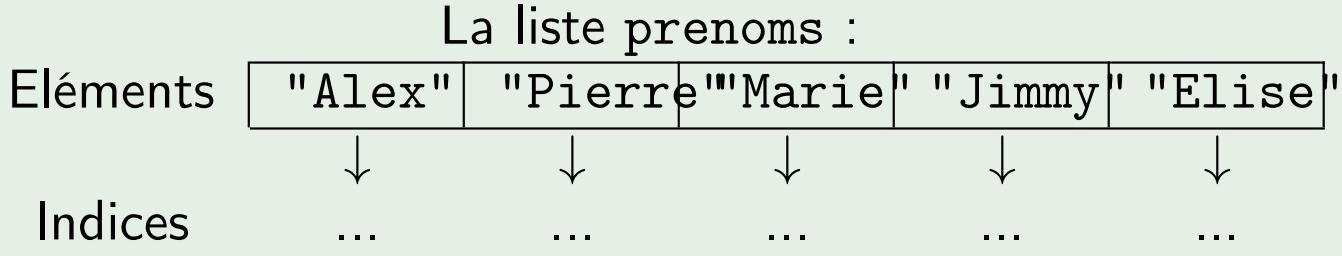
- Que contient `prenoms[2]` ?

C12 Les tableaux

Exemple

On considère la liste de prénoms suivants :
`prenoms = ["Alex","Pierre","Marie","Jimmy","Elise"]`

- Compléter le schéma suivant permettant de représenter cette liste



- Que contient `prenoms[2]` ?
- Comment accéder au premier élément de cette liste (c'est à dire "Alex") ?

C12 Les tableaux

Longueur et index négatif

- On peut accéder aux éléments d'une liste à partir de la fin en utilisant des index négatifs. L'indice -1 est le dernier élément, -2 l'avant dernier, ...

C12 Les tableaux

Longueur et index négatif

- On peut accéder aux éléments d'une liste à partir de la fin en utilisant des index négatifs. L'indice -1 est le dernier élément, -2 l'avant dernier, ...
- La fonction `len` renvoie la longueur d'une liste, c'est à dire son nombre d'éléments.

C12 Les tableaux

Longueur et index négatif

- On peut accéder aux éléments d'une liste à partir de la fin en utilisant des index négatifs. L'indice -1 est le dernier élément, -2 l'avant dernier, ...
- La fonction `len` renvoie la longueur d'une liste, c'est à dire son nombre d'éléments.

Exemples

On considère la liste `voyelles = ["a","e","i","o","u","y"]`

C12 Les tableaux

Longueur et index négatif

- On peut accéder aux éléments d'une liste à partir de la fin en utilisant des index négatifs. L'indice -1 est le dernier élément, -2 l'avant dernier, ...
- La fonction `len` renvoie la longueur d'une liste, c'est à dire son nombre d'éléments.

Exemples

On considère la liste `voyelles = ["a","e","i","o","u","y"]`

- Que renvoie `len(voyelles)` ?

C12 Les tableaux

Longueur et index négatif

- On peut accéder aux éléments d'une liste à partir de la fin en utilisant des index négatifs. L'indice -1 est le dernier élément, -2 l'avant dernier, ...
- La fonction `len` renvoie la longueur d'une liste, c'est à dire son nombre d'éléments.

Exemples

On considère la liste `voyelles = ["a","e","i","o","u","y"]`

- Que renvoie `len(voyelles)` ?
- Que va afficher `print(voyelles[-2])` ?

C12 Les tableaux

Longueur et index négatif

- On peut accéder aux éléments d'une liste à partir de la fin en utilisant des index négatifs. L'indice -1 est le dernier élément, -2 l'avant dernier, ...
- La fonction `len` renvoie la longueur d'une liste, c'est à dire son nombre d'éléments.

Exemples

On considère la liste `voyelles = ["a","e","i","o","u","y"]`

- Que renvoie `len(voyelles)` ?
- Que va afficher `print(voyelles[-2])` ?
- Que va afficher `print(voyelles[2])` ?

C12 Les tableaux

Longueur et index négatif

- On peut accéder aux éléments d'une liste à partir de la fin en utilisant des index négatifs. L'indice -1 est le dernier élément, -2 l'avant dernier, ...
- La fonction `len` renvoie la longueur d'une liste, c'est à dire son nombre d'éléments.

Exemples

On considère la liste `voyelles = ["a","e","i","o","u","y"]`

- Que renvoie `len(voyelles)` ?
- Que va afficher `print(voyelles[-2])` ?
- Que va afficher `print(voyelles[2])` ?
- Que va afficher `print(voyelles[6])` ?

C12 Les tableaux

Longueur et index négatif

- On peut accéder aux éléments d'une liste à partir de la fin en utilisant des index négatifs. L'indice -1 est le dernier élément, -2 l'avant dernier, ...
- La fonction `len` renvoie la longueur d'une liste, c'est à dire son nombre d'éléments.

Exemples

On considère la liste `voyelles = ["a","e","i","o","u","y"]`

- Que renvoie `len(voyelles)` ?
- Que va afficher `print(voyelles[-2])` ?
- Que va afficher `print(voyelles[2])` ?
- Que va afficher `print(voyelles[6])` ?
- Donner deux façons d'afficher le dernier élément de cette liste.

Opérations sur les listes

Les opérations suivantes permettent de manipuler les listes (ajout, suppression, insertion d'éléments). On fera bien attention à la syntaxe on met le nom de la liste suivi d'un point suivi de l'opération à effectuer (voir exemples)

- `remove` permet de supprimer un élément d'une liste. Par exemple : `ma_liste.remove(elt)` va enlever `elt` de `ma_liste`.

Opérations sur les listes

Les opérations suivantes permettent de manipuler les listes (ajout, suppression, insertion d'éléments). On fera bien attention à la syntaxe on met le nom de la liste suivi d'un point suivi de l'opération à effectuer (voir exemples)

- `remove` permet de supprimer un élément d'une liste. Par exemple : `ma_liste.remove(elt)` va enlever `elt` de `ma_liste`.
- `append` permet d'ajouter un élément à la fin d'une liste. Par exemple : `ma_liste.append(elt)` va ajouter `elt` à la fin de `ma_liste`.

Opérations sur les listes

Les opérations suivantes permettent de manipuler les listes (ajout, suppression, insertion d'éléments). On fera bien attention à la syntaxe on met le nom de la liste suivi d'un point suivi de l'opération à effectuer (voir exemples)

- `remove` permet de supprimer un élément d'une liste. Par exemple : `ma_liste.remove(elt)` va enlever `elt` de `ma_liste`.
- `append` permet d'ajouter un élément à la fin d'une liste. Par exemple : `ma_liste.append(elt)` va ajouter `elt` à la fin de `ma_liste`.
- `insert` permet d'insérer un élément à un indice donnée. Par exemple : `ma_liste.insert(indice,elt)` va insérer `elt` dans `ma_liste` à l'index `indice`.

C12 Les tableaux

Opérations sur les listes

Les opérations suivantes permettent de manipuler les listes (ajout, suppression, insertion d'éléments). On fera bien attention à la syntaxe on met le nom de la liste suivi d'un point suivi de l'opération à effectuer (voir exemples)

- `remove` permet de supprimer un élément d'une liste. Par exemple : `ma_liste.remove(elt)` va enlever `elt` de `ma_liste`.
- `append` permet d'ajouter un élément à la fin d'une liste. Par exemple : `ma_liste.append(elt)` va ajouter `elt` à la fin de `ma_liste`.
- `insert` permet d'insérer un élément à un indice donnée. Par exemple : `ma_liste.insert(indice,elt)` va insérer `elt` dans `ma_liste` à l'index `indice`.
- `pop` permet de récupérer un élement de la liste tout en le supprimant de la liste. Par exemple `elt=ma_liste.pop(2)` va mettre dans `elt` `ma_liste[2]` et dans le même temps supprimer cet élément de la liste.

C12 Les tableaux

Exemples

On considère la liste `ex = ['P','T','O','J']`

C12 Les tableaux

Exemples

On considère la liste `ex = ['P','T','O','J']`

C12 Les tableaux

Exemples

On considère la liste `ex = ['P','T','O','J']`

Exemples

On considère la liste `ex = ['P','T','O','J']`

- Supprimer 'J' de cette liste ?

C12 Les tableaux

Exemples

On considère la liste `ex = ['P','T','O','J']`

- Supprimer 'J' de cette liste ?
- Ajouter 'N' en fin de liste

C12 Les tableaux

Exemples

On considère la liste `ex = ['P','T','O','J']`

- Supprimer 'J' de cette liste ?
- Ajouter 'N' en fin de liste
- Insérer 'Y' en indice 1

Exemples

On considère la liste `ex = ['P','T','O','J']`

- Supprimer 'J' de cette liste ?
- Ajouter 'N' en fin de liste
- Insérer 'Y' en indice 1
- Insérer 'H' en indice 3

C12 Les tableaux

Exemples

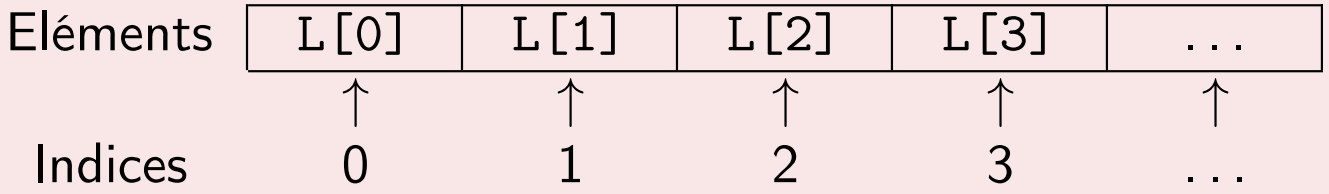
On considère la liste `ex = ['P','T','O','J']`

- Supprimer 'J' de cette liste ?
- Ajouter 'N' en fin de liste
- Insérer 'Y' en indice 1
- Insérer 'H' en indice 3
- Quel sera l'effet de l'instruction `lettre = ex.pop(3)` ?

C12 Les tableaux

Parcours d'une liste

On rappelle qu'une liste `L`, en Python peut se représenter par le schéma suivant :

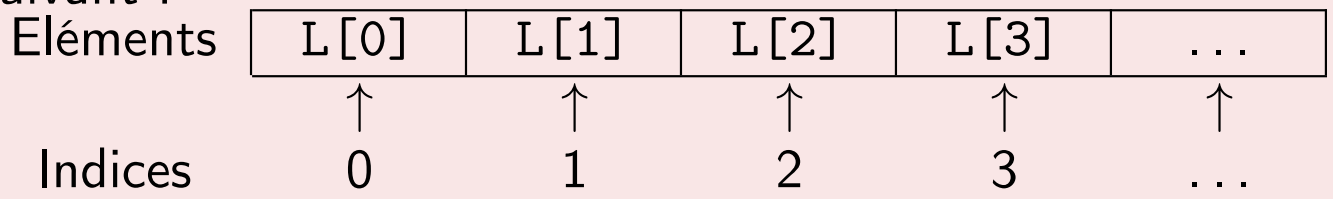


On peut parcourir cette liste :

C12 Les tableaux

Parcours d'une liste

On rappelle qu'une liste `L`, en Python peut se représenter par le schéma suivant :



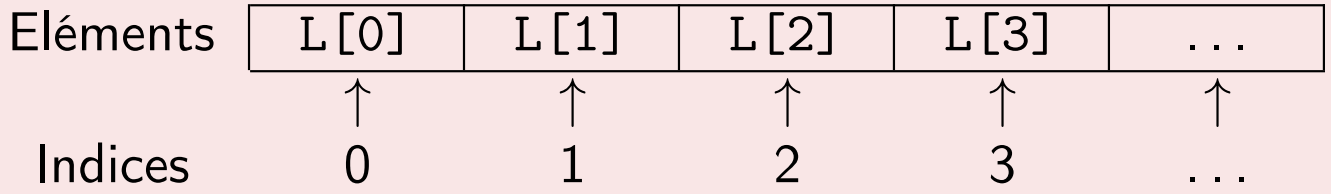
On peut parcourir cette liste :

- Par indice (on se place sur la seconde ligne du schéma ci-dessus) et on crée une variable (un entier) qui va parcourir la liste des indices : `for indice in range(len(L))`
  Il faut alors accéder aux éléments en utilisant leurs indices.

C12 Les tableaux

Parcours d'une liste

On rappelle qu'une liste `L`, en Python peut se représenter par le schéma suivant :



On peut parcourir cette liste :

- Par indice (on se place sur la seconde ligne du schéma ci-dessus) et on crée une variable (un entier) qui va parcourir la liste des indices : `for indice in range(len(L))`
  Il faut alors accéder aux éléments en utilisant leurs indices.
- Par élément (on se place sur la première ligne du schéma ci-dessus) et on crée une variable qui va parcourir directement la liste des éléments : `for element in L`
  La variable de parcours (ici `element`) contient alors directement les éléments).

Liste et chaîne de caractères

- La notation avec les crochets permettant d'accéder aux éléments d'une liste s'utilise aussi avec les chaines de caractères.

C12 Les tableaux

Liste et chaîne de caractères

- La notation avec les crochets permettant d'accéder aux éléments d'une liste s'utilise aussi avec les chaines de caractères.
  Par exemple si `mot = "Génial"` alors `mot[2]` contient la lettre

C12 Les tableaux

Liste et chaîne de caractères

- La notation avec les crochets permettant d'accéder aux éléments d'une liste s'utilise aussi avec les chaines de caractères.
  Par exemple si `mot = "Génial"` alors `mot[2]` contient la lettre `"n"`

C12 Les tableaux

Liste et chaîne de caractères

- La notation avec les crochets permettant d'accéder aux éléments d'une liste s'utilise aussi avec les chaines de caractères.
  Par exemple si `mot = "Génial"` alors `mot[2]` contient la lettre `"n"`
- Le parcours par élément peut aussi se faire sur une chaine de caractères.

Liste et chaîne de caractères

- La notation avec les crochets permettant d'accéder aux éléments d'une liste s'utilise aussi avec les chaines de caractères.
  Par exemple si `mot = "Génial"` alors `mot[2]` contient la lettre `"n"`
- Le parcours par élément peut aussi se faire sur une chaine de caractères.
  Pour afficher chaque lettre du mot "Génial", on peut donc écrire :

C12 Les tableaux

Liste et chaîne de caractères

- La notation avec les crochets permettant d'accéder aux éléments d'une liste s'utilise aussi avec les chaines de caractères.
  Par exemple si `mot = "Génial"` alors `mot[2]` contient la lettre `"n"`
- Le parcours par élément peut aussi se faire sur une chaine de caractères.
  Pour afficher chaque lettre du mot "Génial", on peut donc écrire :
  `for lettre in mot:`

C12 Les tableaux

Liste et chaîne de caractères

- La notation avec les crochets permettant d'accéder aux éléments d'une liste s'utilise aussi avec les chaines de caractères.
  Par exemple si `mot = "Génial"` alors `mot[2]` contient la lettre `"n"`
- Le parcours par élément peut aussi se faire sur une chaine de caractères.
  Pour afficher chaque lettre du mot "Génial", on peut donc écrire :

```
for lettre in mot:
    print(lettre)
```

C12 Les tableaux

Exemple

Soit la liste Python : `notes = [17,12,9,11,13,15,8]`, en effectuant des parcours de cette liste, écrire un programme permettant :

- D'écrire les élements de cette liste qui sont supérieurs à 10.

C12 Les tableaux

Exemple

Soit la liste Python : `notes = [17,12,9,11,13,15,8]`, en effectuant des parcours de cette liste, écrire un programme permettant :

- D'écrire les élements de cette liste qui sont supérieurs à 10.
- De calculer la somme des éléments de cette liste.

C12 Les tableaux

Exemple

Soit la liste Python : `notes = [17,12,9,11,13,15,8]`, en effectuant des parcours de cette liste, écrire un programme permettant :

- D'écrire les élements de cette liste qui sont supérieurs à 10.
- De calculer la somme des éléments de cette liste.
- De créer une nouvelle liste à partir de cette liste en ne conservant que les éléments inférieurs à 10

C12 Les tableaux

Exemple

Soit la liste Python : `notes = [17,12,9,11,13,15,8]`, en effectuant des parcours de cette liste, écrire un programme permettant :

- D'écrire les élements de cette liste qui sont supérieurs à 10.

```
1    notes = [17,12,9,11,13,15,8]
2    for note in notes:
3        if note>10:
4            print(note)
```

C12 Les tableaux

Exemple

Soit la liste Python : `notes = [17,12,9,11,13,15,8]`, en effectuant des parcours de cette liste, écrire un programme permettant :

- D'écrire les élements de cette liste qui sont supérieurs à 10.
- De calculer la somme des élements de cette liste

```
1    notes = [17,12,9,11,13,15,8]
2    somme_notes=0
3    for note in notes:
4        somme_notes = somme_notes + note
```

C12 Les tableaux

Exemple

Soit la liste Python : `notes = [17,12,9,11,13,15,8]`, en effectuant des parcours de cette liste, écrire un programme permettant :

- D'écrire les élements de cette liste qui sont supérieurs à 10.
- De calculer la somme des élements de cette liste
- De créer une nouvelle liste à partir de cette liste en ne conservant que les éléments inférieurs ou égaux à 10

```
1    notes = [17,12,9,11,13,15,8]
2    notes_inf_10 = []
3    for note in notes:
4        if note<=10:
5            notes_inf_10.append(note)
```

C12 Les tableaux

Création de listes

On peut créer des listes de diverses façons en Python :

C12 Les tableaux

Création de listes

On peut créer des listes de diverses façons en Python :

- Par ajout succesif d'élement on part alors d'une liste (éventuellement vide) et on ajoute chaque élément à l'aide d'instruction `append`.

C12 Les tableaux

Création de listes

On peut créer des listes de diverses façons en Python :

- Par ajout succesif d'élement on part alors d'une liste (éventuellement vide) et on ajoute chaque élément à l'aide d'instruction `append`.
- Par répétition du même élément on utilise alors le caractère `*` pour indiquer le nombre de répétitions.

C12 Les tableaux

Création de listes

On peut créer des listes de diverses façons en Python :

- Par ajout succesif d'élement on part alors d'une liste (éventuellement vide) et on ajoute chaque élément à l'aide d'instruction `append`.
- Par répétition du même élément on utilise alors le caractère `*` pour indiquer le nombre de répétitions.
  Par exemple pour créer la liste :
  `bavardages = ["bla", "bla", "bla", "bla"]`
  on peut simplement écrire :

C12 Les tableaux

Création de listes

On peut créer des listes de diverses façons en Python :

- Par ajout succesif d'élement on part alors d'une liste (éventuellement vide) et on ajoute chaque élément à l'aide d'instruction `append`.
- Par répétition du même élément on utilise alors le caractère `*` pour indiquer le nombre de répétitions.
  Par exemple pour créer la liste :
  `bavardages = ["bla", "bla", "bla", "bla"]`
  on peut simplement écrire :
  `bavardages = ["bla"]*4`

C12 Les tableaux

Création de listes

On peut créer des listes de diverses façons en Python :

- Par ajout succesif d'élement on part alors d'une liste (éventuellement vide) et on ajoute chaque élément à l'aide d'instruction `append`.
- Par répétition du même élément on utilise alors le caractère `*` pour indiquer le nombre de répétitions.
  Par exemple pour créer la liste :
  `bavardages = ["bla", "bla", "bla", "bla"]`
  on peut simplement écrire :
  `bavardages = ["bla"]*4`
- Par compréhension, c'est à dire en indiquant la définition des éléments qui composent la liste.

C12 Les tableaux

Création de listes

On peut créer des listes de diverses façons en Python :

- Par ajout succesif d'élement on part alors d'une liste (éventuellement vide) et on ajoute chaque élément à l'aide d'instruction `append`.
- Par répétition du même élément on utilise alors le caractère `*` pour indiquer le nombre de répétitions.
  Par exemple pour créer la liste :
  `bavardages = ["bla", "bla", "bla", "bla"]`
  on peut simplement écrire :
  `bavardages = ["bla"]*4`
- Par compréhension, c'est à dire en indiquant la définition des éléments qui composent la liste.
  Par exemple la liste `puissances2 = [1, 2, 4, 8, 16, 32, 64, 128]` est constitué des huits premières puissances de 2

C12 Les tableaux

Création de listes

On peut créer des listes de diverses façons en Python :

- Par ajout succesif d'élement on part alors d'une liste (éventuellement vide) et on ajoute chaque élément à l'aide d'instruction `append`.
- Par répétition du même élément on utilise alors le caractère `*` pour indiquer le nombre de répétitions.
  Par exemple pour créer la liste :
  `bavardages = ["bla", "bla", "bla", "bla"]`
  on peut simplement écrire :
  `bavardages = ["bla"]*4`
- Par compréhension, c'est à dire en indiquant la définition des éléments qui composent la liste.
  Par exemple la liste `puissances2 = [1, 2, 4, 8, 16, 32, 64, 128]` est constitué des huits premières puissances de 2
  Elle contient donc $2^0, 2^1, 2^2, \ldots 2^7$, ce qui se traduit en Python par :

C12 Les tableaux

Création de listes

On peut créer des listes de diverses façons en Python :

- Par ajout succesif d'élement on part alors d'une liste (éventuellement vide) et on ajoute chaque élément à l'aide d'instruction `append`.
- Par répétition du même élément on utilise alors le caractère `*` pour indiquer le nombre de répétitions.
  Par exemple pour créer la liste :
  `bavardages = ["bla", "bla", "bla", "bla"]`
  on peut simplement écrire :
  `bavardages = ["bla"]*4`
- Par compréhension, c'est à dire en indiquant la définition des éléments qui composent la liste.
  Par exemple la liste `puissances2 = [1, 2, 4, 8, 16, 32, 64, 128]` est constitué des huits premières puissances de 2
  Elle contient donc $2^0, 2^1, 2^2, \ldots 2^7$, ce qui se traduit en Python par :
  `puissances2 = [2**k for k in range(8)]`

Exemple

Créer les listes suivantes par le moyen qui vous semble le plus approprié :

- La liste des 20 premiers multiples de 7

Exemple

Créer les listes suivantes par le moyen qui vous semble le plus approprié :

- La liste des 20 premiers multiples de 7
- La liste constituée de 100 zéros

Exemple

Créer les listes suivantes par le moyen qui vous semble le plus approprié :

- La liste des 20 premiers multiples de 7
- La liste constituée de 100 zéros
- La liste des lettres de l'alphabet