Fonctions

Les fonctions

- Pour pouvoir réutiliser plusieurs fois certains traitements qu'on fait on utilise les fonctions.
- On en déjà rencontré plusieurs : print, range, etc
- Maintenant on va voir comment on peut créer nos propres fonctions.
- Une fonction peut a ramètres.
- La syntaxe est :

```
def nom_fonction(parametre):

    # traitement
    # à
    # effectuer
    return valeur
```

- Exercice : Ecrire une fonction qui permet de calculer l'aire d'un disque pour un rayon donné en paramètre.

TP : Exercices

- **Exo 1 :** écrire une fonction qui calcule et renvoie le maximum de deux entiers.
- **Exo 1** : écrire une fonction qui permet de calculer la distance entre deux points A(xA,yA) et B (xB, yB).
- **Exo 2** :
  - écrire une fonction cube qui retourne le cube de son argument.
  - Écrire une fonction volumeSphere qui calcule le volume d'une sphère de rayon r donné en paramètre et qui utilise la fonction cube
- **Exo 4** : Initialiser les entiers a et b par les valeurs 0 et 10 respectivement.
  - Écrire une boucle affichant et incrémentant la valeur de a tant qu'elle reste inférieure à celle de b.
  - Ecrire une autre boucle décrémentant la valeur de b et affichant sa valeur si elle est impaire. Boucler tant que b n'est pas nul.
- **Exo 5** : écrire une fonction qui affiche la représentation binaire d'un nombre entier naturel.

Chaînes de caractères

Chaines de caractères

- On déclare une chaine de caractères comme ceci

  >>>str1**=** "Hello World"

- On peut aussi déclarer une chaine vide :

  >>> str2 **=** " "

Création

- Calcul de la longueur d'une chaine de caractères:

>>>print(len**(**"Hello World" **))**

- Les commandes suivantes permettent d'accéder au i ème caractère. Les analyser :

>>>print("Hello World" **[**0**])**

>>>print("Hello World" **[**5**])**

>>>print("Hello World" **[-**1**])**

Indices

>>>texte**[**debut **:** fin**]** #permet d'afficher les caractères d'un texte compris entre l'indice debut et l'indice fin-1:

- **>>>print(**"Hello World"**[**4**:**10**])**
- **>>>print(**"Hello World"**[:**4**])**
- **>>>print(**"Hello world"**[**3**:])**

Concaténation

- Pour concaténer deux textes c'est-à-dire les mettre bout à bout :

```
>>>texte1 + texte2
```

- Exemple :

```
>>>texte1="Vive"
>>>texte2=" la programmation"
>>>texte3=" en Python"
>>>print(texte1+texte2+texte3)
```

Cast d’un objet en string

- Pour transformer un objet en texte (quand cela est possible)

```
>>>str(objet)
```

- Utilité : rajouter des variables de type entier dans un texte.

```
>>> n = 1998
>>>texte = str(n)
>>>print(texte[3])
```

Répétition (facile)

- Si on veut répéter un texte k fois d'affilée :

```
>>>texte = "Je ferai attention pendant que le prof explique le cours! "
>>>print(texte*5)
```

Sous-texte dans un texte

- Pour savoir si un texte contient un sous texte :

  >>> sous_texte **in** texte
- Renvoie **True** si la chaine de caractères sous_texte est contenue dans texte.
- Exemple 1:

  >>>texte **=** "Je vois la vie en rose"

  >>>print("rouge" **in** texte**)**

  >>>print("rose" **in** texte**)**

  >>>print("violet" **in** texte**)**

Exercice

- Ecrire un programme qui détermine si une lettre saisie par l'utilisateur est une consonne ou une voyelle.

Solution

```
print("saisir une lettre")
malettre = input()
if malettre in "aeiouyAEIOUY":
    print(malettre + " est une voyelle")
else:
    print(malettre +" est une consonne")
```

Que fait le programme suivant

```
texte="Trois bonbons à cinquante centimes coutent 1,50 DH"
for caractere in texte:
    if caractere in "0123456789":
        print(caractere)
```

Comparaison de chaines de caractères

texte1 **==** texte2 # renvoie True si les deux textes sont identiques

texte1 **!=** texte2 # renvoie True si les deux textes ont au moins un caractère différent.

texte1 **<** texte2 # renvoie True si texte1 vient avant texte2 dans l'ordre du dictionnaire.

texte1 **<=** texte2 # renvoie True si texte1 vient avant texte2 mais peuvent être les mêmes.

texte1 **>** texte2 # renvoie True si texte1 vient apres texte2 dans l'ordre du dictionnaire.

texte1 **>=** texte2 # renvoie True si texte1 vient après texte2 mais peuvent être les mêmes.

Remarque : En realité on compare les codes ascii

Exercices (chaines de caractères)

Exercice 1 :

Pour le nombre n donné dans la fenêtre ci-dessous, énumérez ses chiffres.

Pour l'affichage, on utilisera print et les chiffres seront affichés en allant à la ligne à chaque fois.

Exercice 2 :

Pour le texte suivant, créer un programme qui affiche le nombre de voyelles.

texte = « Un chasseur sachant chasser doit savoir chasser sans son chien »

Pour l'affichage, on utilisera print