

TP 4 - Fonctions

Exercice 1

1. Ecrire un programme principal qui fait appel à une fonction *carre()* retournant le carré d'un nombre.

```
>>> carre(4)
16
>>> carre(11.11111111)
123.45679009876542
>>> |
```

2. Compléter le script précédent afin d'afficher le carré des nombres entiers de 1 à 100.

```
>>>
0 2 = 0
1 2 = 1
2 2 = 4
3 2 = 9
4 2 = 16
5 2 = 25
6 2 = 36
7 2 = 49
8 2 = 64
9 2 = 81
10 2 = 100
11 2 = 121
12 2 = 144
13 2 = 169
```

Exercice 2

1. Ecrire un programme principal qui fait appel à une fonction retournant l'aire d'un disque de rayon R.

```
>>> airedisque(2.5)
19.634954084936208
>>>
```

Remarque : Le rayon R sera un paramètre de la fonction.

2. Modifier la fonction précédente en ajoutant un paramètre permettant de préciser l'unité de mesure de cette aire.

```
>>> airedisque(4.2, "cm")
55.41769440932395 cm 2
>>> |
```

Exercice 3

Écrire une fonction qui compte le nombre de voyelles (minuscules, sans accents) dans une chaîne de caractères passée en argument.

```
>>> nb_voyelles("Combien de voyelles y a-t-il dans ce texte?")
13
>>> |
```

Exercice 4

Écrire une fonction *egaux(ch,carac)* qui demande une chaîne à l'utilisateur puis un caractère et affiche le nombre de caractères égaux à celui-ci dans la chaîne.

Exercice 5

1. Écrire un programme qui crée une liste de mots tapée successivement au clavier jusqu'à ce que l'utilisateur tape la lettre *F*.
2. Compléter ce programme afin de tester si les mots de cette liste sont rangés dans l'ordre alphabétique.
3. Modifier le programme afin qu'il renvoie le premier nom qui n'est pas dans l'ordre alphabétique.

Exercice 6

1. Écrire la fonction *retourne(ch)* qui étant donné une chaîne de caractères renvoie la chaîne écrite « à l'envers ».
2. Écrire une fonction qui teste si une chaîne de caractères est un palindrome.
3. En déduire un programme qui affiche les années palindromiques de 1 à n.

Une année est palindromique si la chaîne de caractères correspondante est un palindrome.

On pourra utiliser la fonction *str* qui transforme un entier en la chaîne correspondante.

Exercice 7

Écrire un programme qui recherche le plus grand élément présent dans une liste de nombres entrée par l'utilisateur.

Exercice 8

1. Écrire la fonction `etoiles(n)` qui affiche n étoiles sur une ligne (sans saut de ligne à la fin).
2. Écrire la fonction `blancs(p)` qui affiche p blancs sur une ligne (sans saut de ligne à la fin).
3. Écrire la fonction `etoiles_bl(n)` qui affiche n étoiles avec un espace entre elles (sans espace au début et sans saut de ligne à la fin).
4. En déduire les fonctions :
 - `carre(n)` : qui affiche un carré de n étoiles sur n .

```
1  carre(5)
2  * * * * *
3  * * * * *
4  * * * * *
5  * * * * *
6  * * * * *
```

- `rectangle(n,p)` : qui affiche un rectangle de p étoiles sur n étoiles.
- `tri_rect(n)` : qui affiche un triangle « rectangle » d'étoiles.

```
1  tri_rect(5)
2  *
3  * *
4  * * *
5  * * * *
6  * * * * *
```

5. Écrire la fonction `triangle(n)` qui affiche un triangle d'étoiles.

```
1  triangle(5)
2  *
3  * *
4  * * *
5  * * * *
6  * * * * *
```

Exercice 9

A l'aide de la fonction `randint()` du module `random`, écrire une fonction qui retourne un mot de passe de longueur N (chiffres, lettres minuscules ou majuscules).

On donne :

chaîne =

```
>>> password(6)
stZ3PE
>>> password(50)
Q5rQcW4uTV9hYk9qcKxnRBzeheUpWfQ2u2mdtj6gkHWOAO3y8W
>>>
```

'0123456789ABCDEFGHIJKLMNOPQRSTUVWXYZabcdefghijklmnopqrstuvwxyz'