

## Prise en main et premières fonctions dans le langage Python

### Exercice 1 - Parabole et tangente associée en un point donné

On cherche ici à représenter une parabole et ses tangentes associées.

- Dans le langage Python, construire la fonction *parabole* qui pour tout triplet  $(a, b, c)$  donné, renvoie la parabole  $\mathcal{P}$  d'équation  $y = ax^2 + bx + c$ .  
On imposera un tracé avec 1000 points de sorte que le tracé soit suffisamment lisse.
- Dans la console, taper les instructions suivantes. Que remarquez-vous ?

```
In : Z=12,8; Z[0]; Z[1]
```



- On note alors  $T$  la tangente à  $\mathcal{P}$  au point d'abscisse  $x_0$ , d'équation  $y = Ax + B$ .
  - Construire la fonction *tangente* qui pour tout  $(a, b, c, x_0)$  donné, calcule la valeur des coefficients  $A$  et  $B$ . Si  $A = 0$ , alors le programme affichera "la tangente est horizontale", mais dans tous les cas, on renverra les valeurs de  $A$  et  $B$ .
  - En déduire le programme *representation* qui pour tout  $(a, b, c, x_0)$  trace dans une même graphique la parabole  $\mathcal{P}$ , ainsi que la tangente  $T$  au point d'abscisse  $x_0$ .  
Dans ce dernier programme, on veillera à donner un nom à chacune des courbes et on affichera la légende à l'aide de la commande `legend()`.

### Exercice 2 - Opérations sur les fonctions et conséquences géométriques

On note  $f$  la fonction  $f : x \mapsto \sqrt{|x|}$  définie sur  $\mathbb{R}$ .

- Dans le langage Python, définir la fonction  $f$  qui à tout réel  $x$  renvoie l'image de  $x$  par  $f$ .
- Construire la fonction *courbe* qui pour tout couple  $(a, b)$  donné, renvoie la courbe représentative de  $f$  sur  $[a, b]$ .
- Modifier votre programme *courbe* afin que celui-ci prenne pour argument le triplet  $(\lambda, a, b)$ ,  $\lambda \in \mathbb{R}$ , puis renvoie sur un même graphique les courbes associées aux fonctions :
  - $f_1 : x \mapsto f(x - \lambda)$
  - $f_2 : x \mapsto f(x) + \lambda$
  - $f_3 : x \mapsto f(\lambda x)$
  - $f_4 : x \mapsto \lambda f(x)$
- Tester votre programme avec  $\lambda = 2$ . Quelles conséquences géométriques pouvez-vous remarquer ?

### Exercice 3 - Résolution des équations du second degré à coefficients réels

On considère un polynôme du second degré à coefficients réels défini par :

$$\forall x \in \mathbb{R}, p(x) = ax^2 + bx + c$$

- Préciser l'expression des racines de  $p$  en fonction de  $\Delta$  le discriminant associé. On se placera d'abord dans  $\mathbb{R}$ , puis dans  $\mathbb{C}$ .
- Dans le langage Python, construire le programme *racines* qui pour tout triplet  $(a, b, c)$  donné, renvoie les racines de  $p$  dans  $\mathbb{C}$ .  
Attention, on affichera le fait qu'on a une racine double si  $\Delta = 0$ . De plus, on retiendra que les nombres complexes s'écrivent sous la forme  $a + bj$  dans le langage Python.

### Exercice 4 - Développement en série entière de l'exponentielle

On définit pour tout  $n \in \mathbb{N}$  et pour tout  $x \in \mathbb{R}$  :

$$S(n, x) = \sum_{k=0}^n \frac{x^k}{k!} = 1 + x + \frac{x^2}{2!} + \frac{x^3}{3!} + \dots + \frac{x^n}{n!}$$

- Décrire les instructions suivantes :

```
S=0
for k in range(0,n+1):
    S=S+x**k/factorial(k)
return S
```



- Dans le langage Python, et en utilisant la séquence d'instruction précédente, construire la fonction *somme* qui pour tout couple  $(n, x)$  donné, renvoie la valeur de  $S(n, x)$ .
- Construire alors la fonction *dse* qui pour tout triplet  $(n, a, b)$ , renvoie dans une même fenêtre graphique les courbes d'équation  $y = \exp(x)$  et  $y = S(n, x)$  sur  $[a, b]$ .

4. Tester le programme précédent pour quelques valeurs de  $n$ ,  $n \in \mathbb{N}$ .
5. Modifier le programme *dse* afin que celui renvoie dans une même fenêtre, les courbes d'équation  $y = \exp(x)$  et  $y = S(k, x)$  sur  $[a, b]$  pour  $k$  allant de 0 à  $n$ . Que pouvez-vous en déduire ?

### Exercice 5 - Un programme de bienvenue

On souhaite construire un programme de bienvenue.

1. Dans le langage Python, construire le programme *bienvenue* qui ne prend pas d'argument, mais qui demande les informations suivantes à l'utilisateur :
  - quel est votre nom ?
  - êtes-vous une fille ou un garçon ?
  - quel âge avez-vous ?

puis renvoie par exemple : **Bonjour monsieur XXX**. On pourra convenir qu'une fille de moins de 18 ans peut se faire appeler mademoiselle... de plus, on pourra interpeler l'utilisateur à l'aide de la séquence d'instruction `input('question ?')`.

2. Le module **time** nous permet d'avoir des commandes relative à la gestion du temps.

(a) Décrire les instructions suivantes :

```
In : L=localtime(); h=L[3]; print(h);
```

20



(b) Adapter alors votre programme de bienvenue afin que celui-ci renvoie **Bonjour** ou **Bonsoir** en fonction de l'heure affichée.

### Exercice 6 - Un générateur de numéros

Le module **random** est une librairie additionnelle qui permet par exemple de générer des **nombre pseudos-aléatoires** :

```
In : random(); randint(0,100); choice([1,2,3,4,5]);
```

0.43404755211178814

95

3



1. Dans le langage Python, construire le programme *telephone* qui ne prend pas d'argument, mais renvoie un numéro de téléphone portable au hasard de sorte que :

```
In : telephone()
```

[0,6,5,0,1,5,3,2,0,7]



On veillera à ce que le numéro soit bien une liste de 10 chiffres commençant par 06.

2. De la même façon, construire le programme *grille* qui ne prend pas d'argument, mais renvoie une grille de Loto. Attention, elle est constituée de 5 numéros distincts à choisir parmi 1 à 49, auxquels on ajoute un numéro chance entre 1 et 10.