

QUELQUES EXERCICES GRAPHIQUES PYTHON...

Le module `turtle` permet de dessiner dans une fenêtre graphique à l'aide d'un ensemble de primitives (fonctions) dédiées. Pour pouvoir utiliser ces primitives, il est nécessaire d'importer le module `turtle` en début de script :

```
# on importe le module turtle
import turtle
# suite du script Python...
...
```

Voici la liste des principales primitives disponibles :

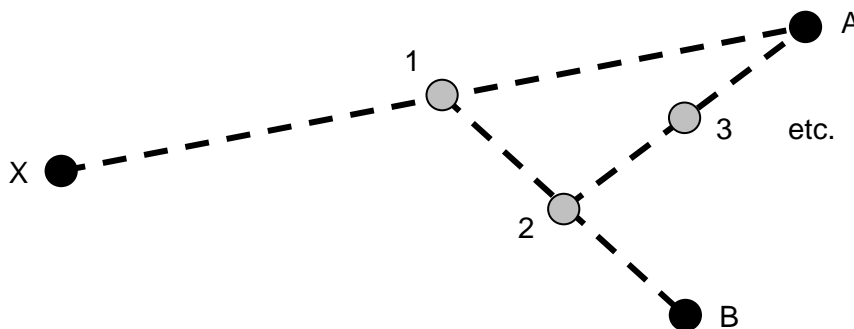
PYTHON - Utilisation du module turtle	
<code>import turtle</code>	permet d'importer les fonctionnalités du module <code>turtle</code>
<code>turtle.title(<chaîne>)</code>	donne un titre à la fenêtre <code>turtle</code> (par défaut : <code>Turtle Graphics</code>)
<code>turtle.reset()</code>	efface l'écran, recentre la tortue en (0,0)
<code>turtle.color(<chaîne>)</code>	détermine la couleur du tracé (noir par défaut). Couleurs prédéfinies : 'red', 'blue', 'green', etc.
<code>turtle.width(<épaisseur>)</code>	détermine l'épaisseur du tracé
<code>turtle.speed(<vitesse>)</code>	détermine la vitesse du tracé (valeur entière)
<code>turtle.forward(<distance>)</code>	avance d'une distance donnée
<code>turtle.backward(<distance>)</code>	recule d'une distance donnée
<code>turtle.left(<angle>)</code>	tourne à gauche d'un angle donné (exprimé en degrés)
<code>turtle.right(<angle>)</code>	tourne à droite d'un angle donné (exprimé en degrés)
<code>turtle.circle(<rayon>{,<angle>})</code>	dessine un cercle de rayon donné, ou un arc de cercle de rayon et angle donnés
<code>turtle.dot(<diamètre>,<couleur>)</code>	dessine un point (cercle plein) de diamètre et couleur donnés
<code>turtle.penup()</code>	relève le crayon (pour pouvoir se déplacer sans dessiner)
<code>turtle.pendown()</code>	abaisse le crayon (pour pouvoir se déplacer en dessinant)
<code>turtle.goto(x,y)</code>	se déplace jusqu'au point de coordonnées (x,y)
<code>turtle.xcor()</code> , <code>turtle.ycor()</code>	retourne la coordonnée courante (abscisse ou ordonnée) de la tortue
<code>turtle.write(<chaîne>)</code>	écrit la chaîne de caractères

- **Dessin de fonction**

Écrire un script Python permettant de tracer la courbe de la fonction $f(x) = x^2/10$, définie sur l'intervalle $[-10,10]$. (L'utilisateur choisira une valeur de pas et la fonction sera dessinée point par point.)

- **La flèche de Xénon**

Xénon est positionné en un certain point du plan X, et deux cibles sont positionnées en des points A et B. Xénon lance une flèche vers la cible A, mais celle-ci tombe au sol à la moitié de la distance souhaitée. Xénon se déplace, ramasse sa flèche, et la lance vers la cible B. Là encore, elle tombe au sol à la moitié de la distance souhaitée. Xénon continue ainsi, en visant alternativement les cibles A et B. Écrire un script Python qui dessine la position initiale X, les points A et B, et les positions successives de Xénon (on demandera à l'utilisateur de fixer le nombre de lancers de flèche). Qu'observez-vous ?



- **Rantanplan...**

Trois Dalton sont placés sur les sommets d'un triangle équilatéral, Rantanplan étant placé au centre du triangle. Rantanplan choisit aléatoirement l'un des Dalton et se dirige vers lui. Parvenu à la moitié de sa course, il s'arrête, choisit un autre Dalton et se dirige vers lui. Parvenu à la moitié de sa course, il s'arrête, choisit un autre Dalton et ainsi de suite...

Écrire un script Python qui dessine les positions des trois Dalton ainsi que les positions successives de Rantanplan (on demandera à l'utilisateur de fixer le nombre de déplacements de Rantanplan). Qu'observez-vous ?

Note. Pour tirer « aléatoirement » un nombre entier compris entre 1 et N, on utilise le module random de la façon suivante :

```
...
import random
...
x = random.randint(1,N) # nombre entier aléatoire entre 1 et N
```

- **Dessin d'une enveloppe**

Écrire un script Python permettant de dessiner une enveloppe selon le profil ci-dessous (la hauteur et la largeur seront données par l'utilisateur). L'enveloppe sera « centrée » en (0,0).

