

Fiche 2 : Construction de figures géométriques

Exemple 1 : Réalisation d'un triangle équilatéral

Programmation par blocs (langage visuel)	Programmation en langage Python
	<pre> from turtle import* color ("black") shape ("turtle") pensize (10) for n in range (3): forward(100) right (120) </pre> <p style="text-align: center;">OU</p> <pre> from turtle import* Cote=1 while Cote<=3: forward(100) right(120) Cote=Cote+1 </pre>

L'indentation consiste en l'ajout de tabulations ou d'espaces en début de ligne de code.

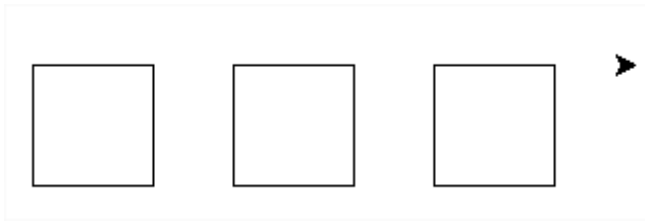
Elle permet de délimiter un bloc code et surtout que le programme s'exécute correctement.

Dans notre script Python, on voit bien le bloc « for » ou « while » délimité à l'aide des tabulations effectuées aux lignes qui suivent.

<p>Module « turtle » : En appelant ce module, Python permet de créer dessiner. Remarque : il faut pour cela importer ce module en début de programme.</p> <p>Color, pensize : Possibilité de choisir la couleur pour dessiner et l'épaisseur du trait.</p> <p>Shape : Possibilité de choisir l'apparence du curseur</p> <table border="1" style="margin-left: auto; margin-right: auto;"> <thead> <tr> <th colspan="2">Apparences du curseur</th> </tr> </thead> <tbody> <tr> <td>arrow</td> <td>flèche</td> </tr> <tr> <td>circle</td> <td>cercle</td> </tr> <tr> <td>square</td> <td>carré</td> </tr> <tr> <td>triangle</td> <td>triangle</td> </tr> <tr> <td>classic</td> <td>classique</td> </tr> <tr> <td>blank</td> <td>rien</td> </tr> </tbody> </table>	Apparences du curseur		arrow	flèche	circle	cercle	square	carré	triangle	triangle	classic	classique	blank	rien	<p>Forward, backward, right, left : Permet d'avancer, de reculer, de tourner à droite ou à gauche. Remarque : les valeurs données entre parenthèses sont en pixels pour déplacer le curseur et en degrés pour tourner.</p> <p>Fonction range() : Cette fonction crée une liste de nombres entiers. range(k) crée la liste des nombres entiers de 0 à k-1 range(p,k) crée la liste des nombres entiers de p à k-1 Exemple : range(3) crée la liste de nombres « 0, 1 et 2 » Exemple : range(2,5) crée la liste de nombres « 2, 3 et 4 »</p> <p>Boucle for : Elle permet d'indiquer le nombre de fois où une action doit être répétée. for n in range(3) : indique donc à l'ordinateur d'exécuter 3 fois le script associé à la boucle for. Attention : l'en-tête de l'instruction for doit toujours se terminer par « : ».</p>
Apparences du curseur															
arrow	flèche														
circle	cercle														
square	carré														
triangle	triangle														
classic	classique														
blank	rien														

Un peu de pratique : modifiez le programme pour obtenir l'affichage d'un carré.

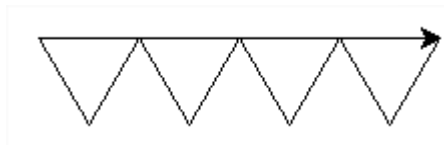
Exemple 2 : Dessiner 3 carrés espacés de 40 pixels en utilisant une fonction sans paramètre



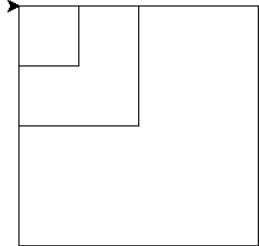
Programmation par blocs (langage visuel) avec SofusPy qui permet de créer des fonctions contrairement à Scratch ou PyBlock	Programmation en langage Python
	<pre>from turtle import * def Carre(): for Nbre_Cotes in range(4): forward(60) right(90) for Nbre_Carres in range(3): Carre() penup() forward(100) pendown()</pre>

<p>def() : Dans le programme ci-dessus a été définie la fonction « Carre » qui permet de tracer un carré de côté 60 pixels.</p>	<p>penup() et pendown() : Permet de lever ou de baisser le stylo.</p>
--	--

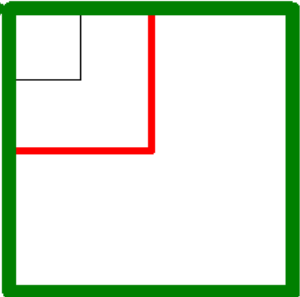
Un peu de pratique : modifiez le programme pour obtenir :




Exemple 3 : Dessiner 3 carrés imbriqués en utilisant une fonction avec un paramètre

Dessin à réaliser	Programmation en langage Python	Commentaires
	<pre>from turtle import* def Carre(Taille): for Nbre_Cotes in range(4): forward(Taille) right(90) Carre(50) Carre(100) Carre(200)</pre>	<p>Ici la fonction Carre() est définie avec un paramètre : le paramètre « Taille ».</p> <p>À la fin du programme, la fonction Carre() est appelée tour à tour avec les arguments 50, 100 puis 200. Ces arguments sont les longueurs (en pixels) des côtés des différents carrés dessinés.</p>

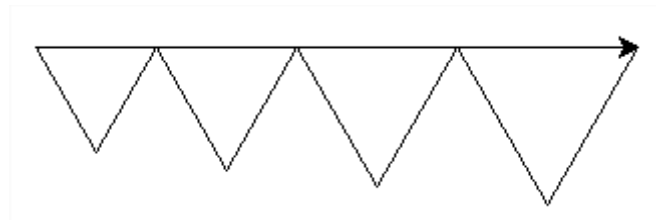
Exemple 4 : dessiner 3 carrés imbriqués en utilisant une fonction avec trois paramètres (taille du carré, épaisseur du trait et couleur du trait)

Dessin à réaliser	Programmation en langage Python	Commentaires
	<pre>from turtle import* def Carre(Taille,Epaisseur,Couleur): for Nbre_Cotes in range(4): pensize(Epaisseur) color(Couleur) forward(Taille) right(90) Carre(50,1,"black") Carre(100,5,"red") Carre(200,10,"green")</pre>	<p>Ici la fonction Carre est définie avec trois paramètres : le paramètre « Taille », le paramètre « Epaisseur » et le paramètre « Couleur ».</p>

Exemple 5 : Dessiner 3 carrés espacés de 50 pixels (avec un 1^{er} carré de côté 60 pixels et les côtés qui augmentent de 10 pour chaque nouveau carré)

Dessin à réaliser	Programmation en langage Python
	<pre data-bbox="842 277 1525 766">from turtle import* i=60 def Carre(taille): for Nbre_Cotes in range(4): forward(taille) right(90) for Nbre_Carres in range(3): Carre(i) penup() forward(50+i) pendown() i=i+10</pre>

Un peu de pratique : modifiez le programme pour obtenir :



La longueur des côtés des triangles augmente de 10 pixels à chaque fois