

Transition Scratch Python en seconde

Exemple de progression et de situations

Situation 4 :

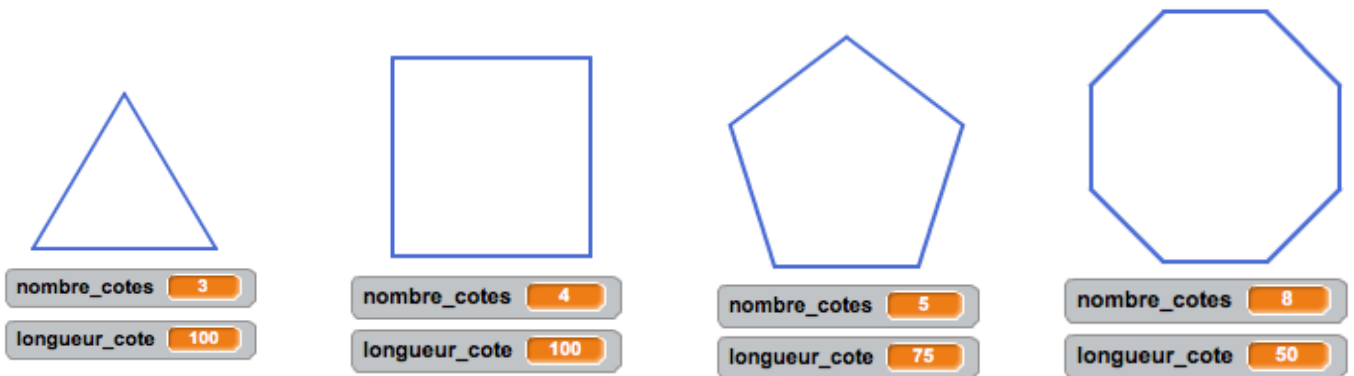
Construction d'un polygone régulier et de figures avec « turtle »

Objectifs relatifs au thème 4 Algorithmique et programmation :

- écriture de l'algorithme en pseudo code
- les boucle « pour », « tant que » avec itérateur et l'utilisation de fonctions en Python

Exercice :

Créer un script Scratch permettant de tracer un polygone régulier où le nombre de côtés et la longueur d'un côté sont choisis par l'utilisateur en entrée.



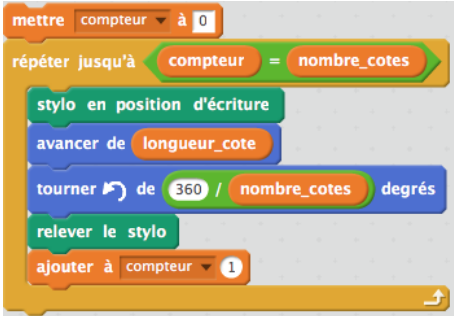
Institutionnalisation : écriture d'une boucle

Boucle « pour »

<u>Script Scratch</u>	<u>Pseudo-code</u>	<u>Code Python</u>
<pre>répéter nombre_cotes fois stylo en position d'écriture avancer de longueur_cote tourner de 360 / nombre_cotes degrés relever le stylo</pre>	<pre>pour i allant de 0 à nombre-cotes stylo en écriture avancer de longueur_cote tourner à gauche de 360/nombre_cotes stylo relevé</pre>	<pre>for i in range (nombre_cotes): down() forward(longueur_cote) left(360/nombre_cotes) up()</pre>

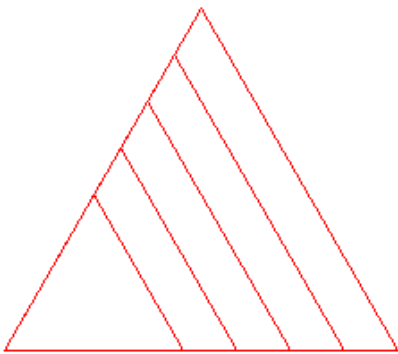
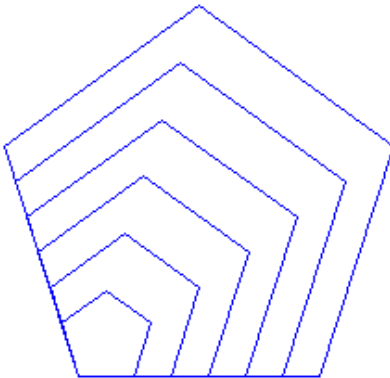
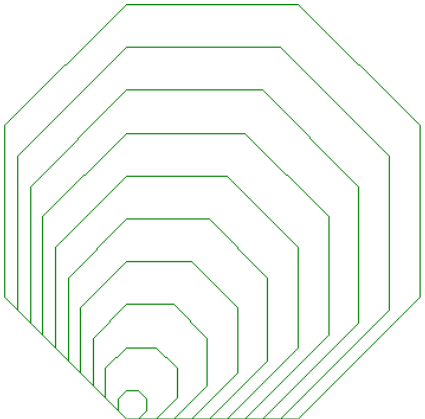
Boucle « tant que »

<u>Script Scratch</u>	<u>Pseudo-code</u>	<u>Code Python</u>
-----------------------	--------------------	--------------------

	<pre>compteur ← 0 tant que compteur ≠ nombre_cotes stylo en écriture avancer de longueur_cote tourner à gauche de 360/nombre_cotes stylo relevé compteur ← compteur + 1</pre>	<pre>compteur=0 while compteur!=nombre_cotes: down() forward(longueur_cote) left(360/nombre_cotes) up() compteur=compteur+1</pre>
--	---	---

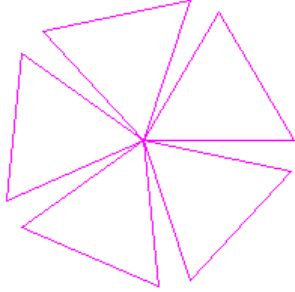
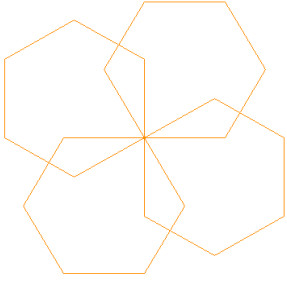
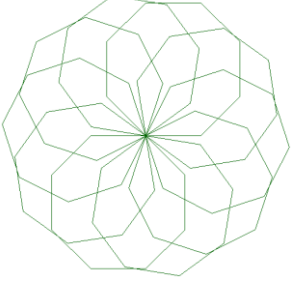
Exercice :

1. Écrire une fonction « PolygoneRegulier » en Python qui prend en paramètres les variables « nombre_cotes » et « longueur_cote » qui dessine un polygone régulier à « nombre_cotes » côtés dont les longueurs mesurent « longueur_cote ».
2. **Défi 1 :**Écrire un code Python utilisant la fonction PolygoneRegulier permettant d'obtenir les figures suivantes avec des variables correspondant au nombre de côtés, à la longueur d'un côté, au nombre de figures, au pas et à la couleur.

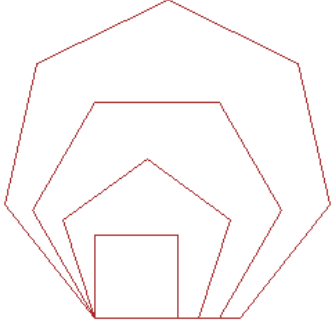
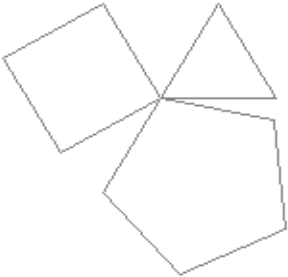
 <pre>nombre_cote =3 longueur_cote=100 nombre_figure=5 pas=30 couleur="red"</pre>	 <pre>nombre_cote =5 longueur_cote=30 nombre_figure=6 pas=20 couleur="blue"</pre>	 <pre>nombre_cote =8 longueur_cote=10 nombre_figure=10 pas=15 couleur="green"</pre>
--	--	---

Prolongements possibles :

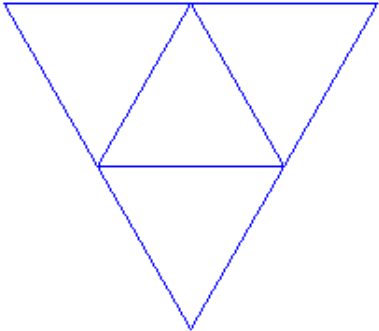
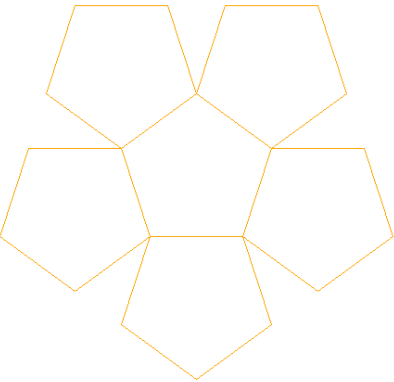
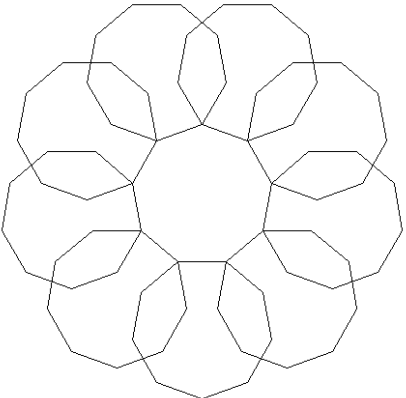
3. **Défi 2 :** Écrire un code Python utilisant la fonction PolygoneRegulier permettant d'obtenir les figures suivantes avec des variables correspondant au nombre de côtés, à la longueur d'un côté, au nombre de figures et à la couleur.

 nombre_cote =3 longueur_cote=100 nombre_figure=5 couleur="magenta"	 nombre_cote =6 longueur_cote=100 nombre_figure=4 couleur="darkorange"	 nombre_cote =8 longueur_cote=100 nombre_figure=10 couleur="darkgreen"
--	---	---

4. **Défi 1bis et 2bis :** Modifier les codes Python des défis 1 et 2 afin de pouvoir obtenir les figures suivantes.

Défi 1bis	Défi 2bis
 nombre_cote =4 longueur_cote=60 nombre_figure=4 pas=15 couleur="brown"	 nombre_cote =3 longueur_cote=60 nombre_figure=3 couleur="grey"

5. **Défi 3 :** Écrire un code Python utilisant la fonction PolygoneRegulier permettant d'obtenir les figures suivantes avec des variables correspondant au nombre de côtés, à la longueur d'un côté, à la couleur .

 nombre_cote =3 longueur_cote=100 couleur="blue"	 nombre_cote =5 longueur_cote=100 couleur="orange"	 nombre_cote =9 longueur_cote=50 couleur="black"
--	---	--

