

Introduction à Python (2)

1) Affichage de la valeur d'une variable à l'écran

Pour afficher le contenu (la valeur) d'une variable à l'écran, on utilise la fonction **print**.

Exemple :

```
a = 12
b = "Coucou"

print(a)
print(b)
print(b, a)
```

Le résultat affiché sera

```
12
Coucou
Coucou 12
```

On peut également combiner du texte, des données et des variables dans un même print :

Essayer le code ci-dessous.

```
heure = 10
minute = 36
print("Il est ", heure, "et ", minute, "minutes.")
```

Dans la console, le message suivant sera alors affiché :

```
Il est 10 heures et 36 minutes.
```

Il suffit de séparer les différentes variables par des virgules.

2) Demander à l'utilisateur de saisir la valeur d'une variable

Il est possible de demander à l'utilisateur de rentrer la valeur d'une variable. Pour cela, on utilise la fonction **input**.

Exemple :

```
prenom = input("Quel est votre prénom?")
```

Avec Edupython, une fenêtre s'ouvre, invitant l'utilisateur à saisir la valeur attendue.

La valeur saisie par l'utilisateur est stockée dans la variable prenom et peut donc être utilisée dans la suite du code.

Rq : avec une autre interface de développement (IDE), il faut saisir la valeur dans la console.

Par défaut la variable renvoyée par la fonction input est une chaîne de caractères. Si on souhaite saisir, un entier ou un flottant (nombre réel), il faut le convertir :

```
age = input("Quel est votre age?")
```

Le type de la variable age est str

```
age = int(input("Quel est votre age?"))
```

Le type de la variable age est int

```
prix = float(input("Quel est le prix de l'objet?"))
```

Le type de la variable prix est float

Écrire un code qui demande (en utilisant la fonction input) le prénom et l'âge de la personne et affiche (en utilisant la fonction print) le message :
Bonjour (insérer le prénom), vous avez (insérer l'âge).

3) Branchement conditionnel

En Python, on utilise des **structures de contrôle** comme les boucles (abordée au TP précédent).

Comme structure de contrôle, on peut utiliser le **branchement conditionnel** aussi appelé **conditions**, qui permettent d'exécuter des instructions en fonction des conditions.

Un test est effectué en utilisant une condition (ou **prédicat**). La condition est évaluée et renvoie vrai (True) ou faux (False).

Syntaxe :

```
if condition :  
    → instruction1  
    → instruction2  
    → ....  
else :  
    → instruction3  
    → instruction4  
    → ....  
fin du branchement conditionnel
```

Si la condition est vérifiée alors on exécute les instructions qui suivent les :

Attention à l'indentation

Sinon on exécute les instructions qui suivent les :

Les instructions qui ne sont pas indentées indiquent que le branchement conditionnel est terminé.

Exemple :

```
age = int(input("Quel est votre age?"))
if age >= 18 :
    print("Vous êtes majeur.")
else :
    print("Vous êtes mineur.")
```

Si on veut tester plusieurs conditions, on peut utiliser **elif** :

```
age = int(input("Quel est votre age?"))
if age >= 18 :
    print("Vous pouvez conduire.")
elif age >= 15:
    print("Vous pouvez commencer la conduite accompagnée.")
else :
    print("Vous n'avez pas le droit de conduire.")
```

Proposer un script qui demande la note obtenue (sur 20) puis affiche une appréciation (ATTENTION ! Il faut que cette note soit convertie au format numérique (entier ou réel) pour pouvoir faire des comparaisons :

Assez-bien à partir de 12
Bien à partir de 14
Très bien à partir de 16
Excellent à partir de 18
Parfait si la note est de 20.

Utilisons de nouveau le module Turtle comme la semaine passée – pour rappel, il faut une structure telle que :

```
import * from turtle

[Instructions.....]

done()
```

Proposer un script qui demande la valeur du nombre de côtés d'un polygone régulier (entre 3 et 5) et qui affiche :

- un triangle équilatéral de 100 pixels de côté si la valeur donnée est 3
- un carré de 100 pixels de côté si la valeur donnée 4
- un pentagone de 100 pixels de côté si la valeur donnée est 5

➔ DEFI TURTLE

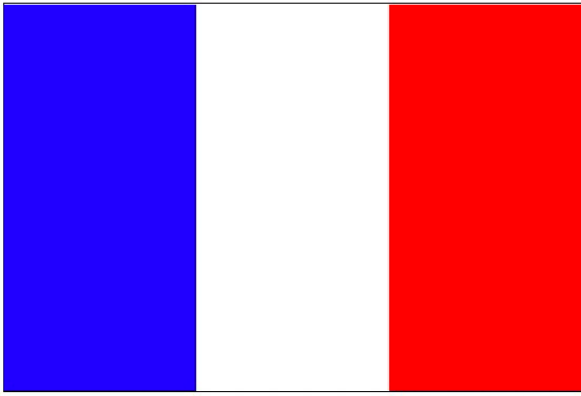
Dessinez dans Turtle les figures de la page suivante, sachant que :

- `penup()` soulève le stylo et `pendown()` le repose (donc entre les deux on peut déplacer la tortue sans dessiner).
- `pensize(10)` change l'épaisseur du trait – je vous laisse tester...
- `pencolor('blue')` donne la couleur bleue au trait qui est dessiné (et évidemment on peut remplacer « blue » par d'autres couleurs en anglais – faites des essais !)
- Pour remplir (colorier) une forme avec une couleur :
 - On commence par définir la couleur : `fillcolor('red')`
 - On dit à Turtle qu'on veut commencer à colorier : `begin_fill()`
 - On dessine ce que l'on veut.... (des carrés ou des rectangles par exemple)
 - On dit à Turtle qu'on a fini de dessiner ce qu'on veut colorier :
`end_fill()`
 - Et Turtle colorie !
- Et pour dessiner un cercle de rayon 100, on écrira : `circle(100)`
- Il est évident qu'ici on va avoir besoin de boucles, comme on en a utilisé la semaine dernière. Pour rappel, pour dessiner un carré, on avait fait :

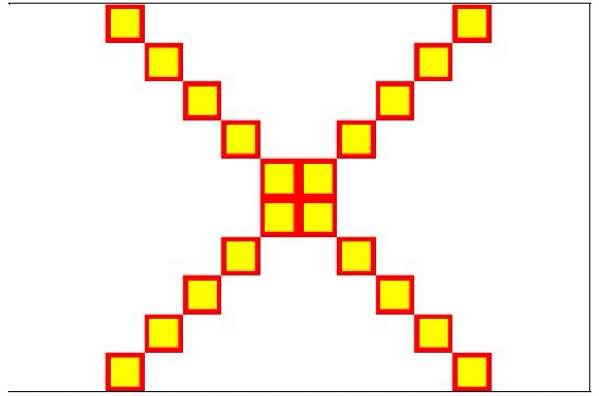
```
for i in range(4):  
    forward(100)  
    right(90)
```

Attention : n'oubliez pas les parenthèses à la fin des fonctions !

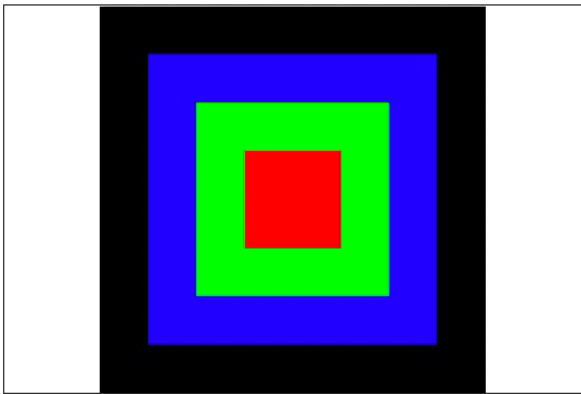
a)



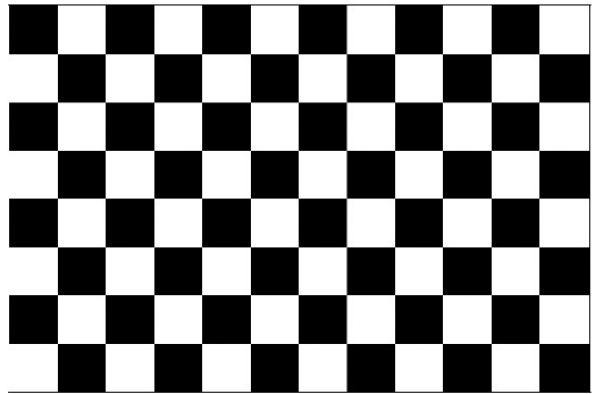
e)



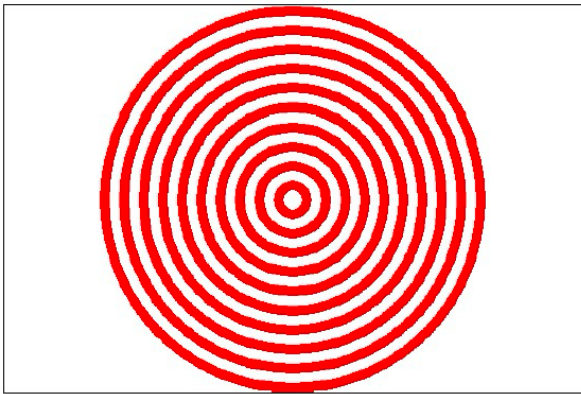
b)



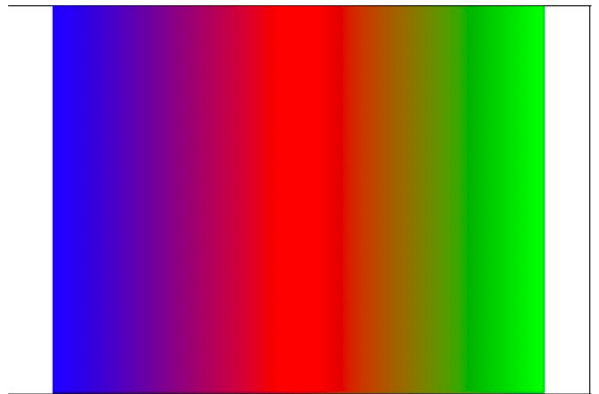
f)



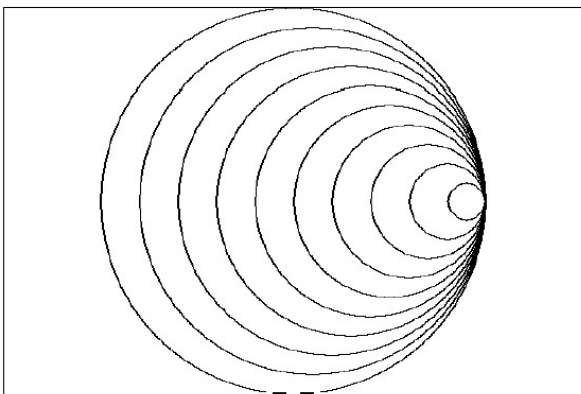
c)



g)



d)



h)

