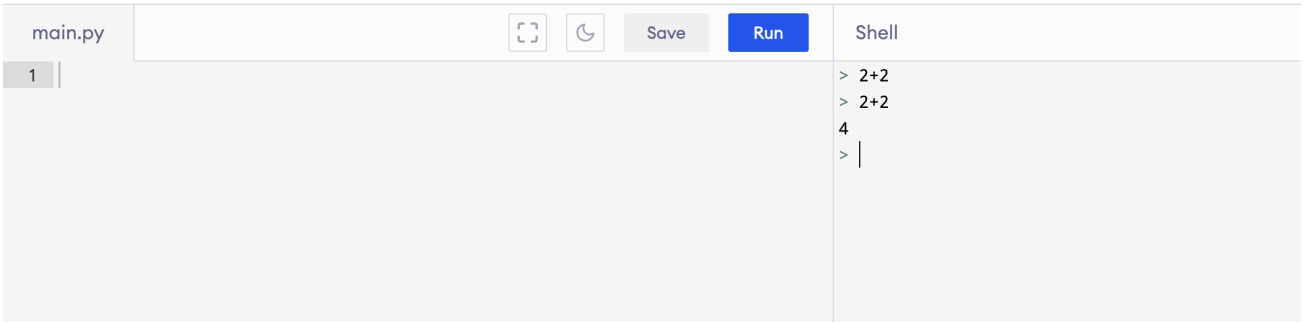
TP nº 2 (noté) : Initiation au langage Python

1/ Découverte de Python

Allez sur le site `https://www.programiz.com/python-programming/online-compiler/` , supprimez tout le contenu du fichier "main.py".

Faire un test de calcul dans la partie de droite, en écrivant 2+2 et en appuyant sur la touche "Entrée" du clavier



Le langage Python permet d'effectuer de nombreux calculs

Opérations	Symbole	Exemple	Avec Python
Addition	+	$25+33=58$	`>>> 25+33` `58`
Soustraction	-	$25-33=-8$	`>>> 25-33` `-8`
Multiplication	*	$25\times33=825$	`>>> 25*33` `825`
Puissance	**	$25^3=15\,625$	`>>> 25**3` `15625`
Division	/	$\frac{23}{3}\approx7{,}66667$	`>>> 23/3` `7.666666666666667`
Reste dans la division entière	%	$23=7\times3+2$	`>>> 23%3` `2`
Quotient dans la division entière	//	$23=7\times3+2$	`>>> 23//3` `7`

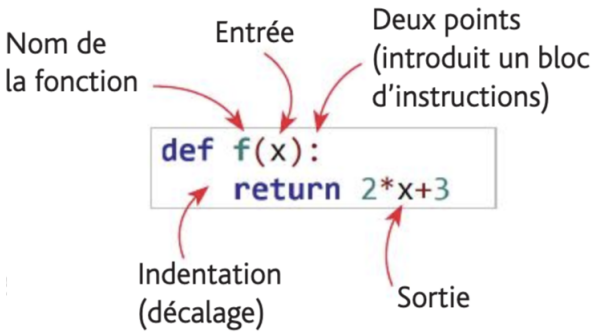
Le langage Python permet aussi de manipuler et d'affecter des variables avec des valeurs.

Consigne	Langage naturel	Langage Python	Exemple
La variable n prend la valeur $n+1$	$n \leftarrow n+1$	$n=n+1$	`>>> n=3` `>>> n=n+1` `>>> n` `4`

Consigne	Langage naturel	Instructions dans l'éditeur	Après lancement du script
La variable x prend la valeur 3. Affecter à x la valeur $x+0{,}1$ puis afficher le résultat.	$x \leftarrow 3$ $x \leftarrow x+0{,}1$ Afficher x	`x=3` `x=x+0.1` `print(x)`	`3.1`

Le langage Python permet par ailleurs de définir et de manipuler des fonctions.

★ Soit f la fonction définie sur $\mathbb{R}$ par $f(x) = 2x + 3$.



- Cette fonction peut être écrite en Python avec l'instruction **def**.
  ATTENTION A L'INDENTATION ! L'indentation est de 4 espaces ou une tabulation et n'oubliez pas les " :"
- Une fonction peut avoir **plusieurs entrées** (paramètres) ou aucune.
- Plusieurs types de sorties sont possibles :
  — Une fonction peut renvoyer un résultat (return), qui pourra être réutilisé (dans une autre fonction parexemple). Cette écriture sera privilégiée car elle permet de faire de la programmation modulaire (découper une tâche complexe en plusieurs tâches simples).
  — Une fonction peut ne pas envoyer de résultat, et, par exemple, se contenter de l'afficher (print).

Consigne	Résultat	
Écrire une fonction, *fonc*, qui connaissant deux entiers renvoie leur produit	`def fonc(x,y):` `  return x*y`	`def fonc(x,y):` `  print(x*y)`
Que renvoie l'instruction fonc(5, –3) ?	`>>> fonc(5,-3)` `-15`	`>>> fonc(5,-3)` `-15`
Que renvoie l'instruction fonc(2, 4) + fonc(–1, 3) ?	`>>> fonc(2,4)+fonc(-1,3)` `5`	Affichage d'un message d'erreur
Commenter	La fonction renvoie un résultat. **En sortie**, on obtient un nombre qui peut être utilisé.	La fonction ne renvoie pas de résultat. **En sortie**, on n'obtient qu'un affichage, il ne peut donc pas être utilisé.

- En mode **console**, pour afficher la valeur d'une variable, il suffit de taper son nom (voir les encadrés précédents).

- En mode **éditeur**, on DOIT utiliser la fonction *print(...)*
  — Pour afficher la valeur d'une variable nommée *dist*, on écrit : *print(dist)* (il faut, bien sûr, que la variable *dist* ait été créée avant ...).
  — Pour afficher un texte, on utilise des guillemets : *print("Ce programme calcule des distances.")*
  — Pour mixer les deux, on sépare avec des virgules : *print("La distance calculée est :",dist)*

2/ Exercices

Exercice n° 1. Traduisez en Python le programme suivants puis entrez le dans la console de Python et donnez la valeur finale de r.

Langage naturel :	En Python :
$m \leftarrow 2$	1 _____
$n \leftarrow m - 1$	2 _____
$p \leftarrow n + 4$	3 _____
$q \leftarrow 3 \times p$	4 _____
$r \leftarrow q - m$	5 _____
Afficher la variable r	6 _____

r vaut _____

Remarque : La langage naturel est la façon la plus simple de décrire un algorithme, malheureusement il est nécessaire de transformer en un langage pour que la la machine puisse l'interpréter ;

Exercice n° 2. Pour chaque question, tapez les instructions suivantes en mode console et dîtes ce que vous remarquez concernant les variables a et b.

(A)

```
1  a=4
2  a=1
3  print(a)
```

(B)

```
1  a=4
2  b=7
3  a=b
4  print(a)
5  print(b)
```

(C)

```
1  a = 4
2  b = 7
3  b = a
4  print(a)
5  print(b)
```

Réponses :

(A)

(B)

(C)

Deux propriétés de l'affectation

Écrasement quand on met une valeur dans une variable, toute valeur existant déjà dans la variable est effacée (écrasée). Par exemple, à la suite des deux instructions :

$a \leftarrow 2$

$a \leftarrow 3$

la variable a contient 3 et pas 5. Si on veut ajouter 3 au contenu précédent de a, on doit faire :

$a \leftarrow a + 3$

Copie l'affectation entre variables copie le contenu d'une variable sans l'effacer. Par exemple, à la suite des trois instructions :

$a \leftarrow 2$

$b \leftarrow 7$

$a \leftarrow b$

le contenu de b est copié dans a. Ainsi les deux variables a et b contiennent la valeur 7.

Remarque : En langage Python $a = b$ ne veut pas dire la même chose que $b = a$.

— $a = b$ veut dire : on met la valeur de b dans a

— $b = a$ veut dire : on met la valeur de a dans b.

Exercice n° 3. Tapez et complétez les algorithmes suivants afin qu'ils affichent ce que le programme doit retourner

(A) **Programme Python**

```
longueur = 141592
largeur  =  27182
print(                                        )
```

Le programme doit retourner :

»» L'aire du rectangle est : 3848753744

(B) **Programme Python**

```
print(                                        )
```

Le programme doit retourner :

»» 2 puissance 100 = 1267650600228229401496703205376

(C) **Programme Python**

```
prenom="Fabien"
print(                                 )
print(                                                                          )
print(                                                          )
```

Le programme doit retourner :

»» Fabien est trop trop beau
Même s'il passe beaucoup de temps sur l'ordinateur Fabien est le plus intelligent de la classe
Vive Fabien ! Vive Fabien ! Vive Fabien !

Les différents types de données

Les données informatiques peuvent se présenter sous différents aspects, entre autres :
- des **nombres entiers**, tels que : 4 ; 236 ; – 21 etc.
- des **nombres réels**, tels que : 3.14 ; – 1.732 etc.
- du **texte** , par exemple : "Bonjour tout le monde !"
- des **tableaux** ou **listes** contenant des données.
  Par exemple : jours = ["Samedi","Dimanche","Lundi","Mardi","Mercredi","Jeudi","Vendredi"]
  Il est parfois nécessaire de passer d'un type à l'autre, on appelle cela un trans-typage.

Exercice nº 4 (Types de données)**.**

(1) Tapez dans la console et notez sur la feuille les réponses de Python

```
» a=30                                  Réponses
» type(a)
» b="test !"
» type(b)
```

Remarque : La commande *type(...)* demande le type de donnée d'une variable. Pour Python, les entiers sont de type *int* (integer = entier) et les textes de type *str* (string = chaîne de caractère).

(2) Tapez dans la console et notez sur la feuilles les réponses de Python :

```
» nb1 = 10                              Réponses
» nb2 = 20
» nb1+nb2
» chn1 = "encore un "
» chn2 = "test"
» chn1+chn2
» chn2+nb1
```

Remarque : **On ne peut pas ajouter un texte et un nombre**. Si on veut obtenir le texte "test10", il faut convertir le nombre 10 en le texte "10". Ceci se fait à l'aide d'une fonction de trans-typage, ici *str(...)*.

Il existe également les commandes *int(...)* pour convertir en entier et *float(...)* pour convertir en nombre réel.

(3) Tapez dans la console et notez sur la feuilles les réponses de Python :

```
» chn2+str(nb1)                         Réponses
» str(nb1)
» float(nb1)
» int(nb1)
» int(5.25)
```

(4) Tapez et complétez, dans l'éditeur de Python, l'algorithme suivant. Les nombres 38 et 7 ne doivent pas apparaître dans les lignes cachées.

```
1 a="38"
2 b=7.0
3 print(                              )
4 print(                                  )
```

Le programme doit retourner :

»» Le produit de 38 et de 7 est 266
La division de 38 par 7 donne environ 5.428571428571429

Entrées

Que se passe-t-il lorsqu'un utilisateur veut retirer de l'argent à un distributeur bancaire ?

- le distributeur affiche une zone d'entrée de code et attend que l'utilisateur entre son code ;
- le distributeur traite ensuite cette information (vérification du code puis actions adaptées).

De même, pour qu'un utilisateur puisse entrer des données dans un algorithme existant, on utilise la commande Saisir (ou Lire ou Demander) en langage naturel.
Cette commande se traduit en langage Python en $input(\dots)$ Par exemple, la commande :

$$\text{votre\_code} = input(\text{"Quel est votre code ?"})$$

affichera le texte « Quel est votre code ? », attendra la réponse de l'utilisateur et placera la réponse dans la variable votre_code (qu'on pourra utiliser plus tard, pour vérification par exemple).

Exercice n° 5.

(1) Écrivez un programme en Python qui demande à un utilisateur son prénom et affiche le message :

Bonjour *prénom_choisi*, comment allez-vous ?

(2) Testez le programme suivant et essayez de le réparer

```
1 nombre = input("Choisi un nombre")
2 print("Le double de", nombre, "est",2*nombre)
```

La commande *input(...)* renvoie automatiquement du texte. Il sera ainsi parfois nécessaire de convertir ce texte en entier ou en réel, comme dans l'exemple ci-dessous :

larg = input("Largeur du rectangle :") larg = float(larg)

ou directement :

larg = float(input("Largeur du rectangle :"))

Exercice bonus. Un magicien demande à un spectateur :

— de penser à un nombre entier ;
— de le multiplier par 5 ;
— d'ajouter 7 au résultat ;
— de multiplier par 4 le résultat ;
— d'ajouter 6 au résultat ;
— de multiplier par 5 le résultat ;
— d'annoncer le résultat final obtenu.

1. Le spectateur pense au nombre 4, quel nombre annonce-t-il ?
2. Le magicien trouve à chaque fois le nombre choisi au départ par le spectateur ! Soit il est très fort en calcul mental, soit il a un truc de magicien... Créez un programme Python qui :
   - demande un nombre entier ;
   - effectue les opérations demandées par le magicien ;
   - affiche le résultat final (celui que le spectateur annonce)
3. A l'aide de ce programme, vérifiez la réponse à la question 1).