

A la découverte des variables

Objectifs

À la fin de cette leçon, vous pourrez :

- Décrire les fondamentaux sur l'usage des variables en Python
- Réaliser des opérations arithmétiques sur les variables.
- Analyser du code Python
- Réaliser et exécuter un script

Introduction

Une variable permet de **stocker une donnée** d'un programme sur la mémoire vive (RAM) de l'ordinateur. A cette donnée, le programmeur associe un nom auquel il pourra faire appel pour faire référence à la donnée plus tard en programmant. Cette leçon présentera d'abord les types de données de base, ensuite la manière dont python gère la déclaration et l'affectation de ces données, et enfin à faire des opérations arithmétiques sur ces données

I. Le monde des variables

Objectifs

- Citer les types de variable de base.
- Décrire le processus de déclaration et d'affectation de valeur à une variable.
- Identifier le type d'une variable sous Python.

1. Type de variable de base

Cette section introduit quatre types de base que sont :

- **Les entiers** désignent tous les éléments qui appartiennent à l'ensemble des nombres entiers naturels **N**. Le type entier se nomme *int* en python.
- **Les réels ou flottants** désignent tous les nombres à virgule dans l'ensemble des nombres réels **R** (exemple : 13.25). On utilise "." au lieu de "," pour écrire un nombre à virgule sous python et le type flottant se nomme *float* en python.
- **Les chaînes de caractère** : Ce type de variable permet de stocker une série de lettre, un mot ou une phrase. La valeur qu'une variable de type chaîne de caractère reçoit s'écrit entre **guillemets** ("ma chaîne") ou entre **apostrophes** ('ma chaîne'). Le type chaîne de caractère se nomme *str* (string) en python.
- **Les booléens** sont des variables d'ordre 2 ; c'est à dire qui ne peut prendre que 2 valeurs **True** ou **False**. Le type booléen se nomme *bool* en python

Attention : Booléen

Il faut bien noter les majuscules pour les 2 valeurs que peuvent prendre les booléens.

2. Déclaration de variable et affectation : typage statique vs typage dynamique

Définition : Affectation

Une affectation ou **assignation** consiste à attribuer une valeur à une variable déclarée. Lorsqu'elle se fait pour la première fois sur une variable, on parle **d'initialisation**. L'opérateur d'affectation est le symbole "=" (égale). une affectation ne renvoie aucune sortie sur l'écran. Il est possible de changer la valeur d'une variable en faisant une ré-affectation.

Typage statique

En **langage C** ou dans tout autre langage utilisant un **mode de traduction compilé**, la déclaration d'une variable se fait en précisant le type de cette variable. On dit que le type de la variable est défini avant l'exécution du programme (**typage statique**). Dans ce cas la valeur qui est affectée à la variable doit être du même type que la variable. **Lors d'une déclaration de variable, l'initialisation n'est pas obligatoire.**

Exemple : langage C

```
int age=20 ou float moyenne=12.50 ou char genre='M' ou float variance
```

Typage dynamique

Avec python, on déclare une variable sans préciser son type. On parle de **typage dynamique**, c'est-à-dire que le type de la variable déclarée est déterminé lors de l'exécution du programme. **Dans ce cas l'initialisation est obligatoire** ; ainsi la variable reçoit le même type que la valeur qui lui a été affectée. Pour découvrir le type d'une variable sous Python, la fonction **type(nom_de_la_variable)** est utilisée.

Exemple : sur python 2 (déclaration, affectation)

```
>>>age=25
>>>type(age)
<type 'int'> --> 25 est un entier, alors la variable age est de type int
>>>nom='Francois'
>>>type(nom)
<type 'str'> --> Francois est une chaîne de caractère, alors la variable nom est de type str
```

3. Nom de variable

Python présente certaines règles concernant le nom qu'un programmeur peut donner à une variable

Règle 1

Pour déclarer une variable qui contient plusieurs mots, il faut soit remplacer chaque espace par un **underscore** ("_") ou tiret de 8, soit supprimer l'espace et mettre la première lettre du mot suivant en majuscule

Exemple : python

mon_age OU mon_salaire_mensuel OU monAge OU monSalaireMensuel

Règle 2

Il est formellement interdit de débiter le nom d'une variable par un chiffre ou d'utiliser dans la composition du nom un symbole autre que underscore, ou encore les caractères accentués (à, é, è, ç, etc.).

Exemple : python

A ne pas faire: 2x_mon_salaire OU mon_école OU mon-salaire-x2

A faire: mon_salaire_x2 OU monAgeEn2018 OU mon_age_en_2018 OU mon_ecole OU monEcole

Règle 3

Python est **sensible à la casse**; c'est-à-dire qu'une variable déclarée avec un caractère majuscule est différente d'une variable déclarée avec un caractère minuscule

Exemple : Python

age, Age et AGE sont trois variables différentes

Conseil : Nom des variables

Il est recommandé de donner des noms de variable **significatif**. Par exemple: `profession` plutôt que `p` pour recueillir la profession d'un utilisateur ou encore `age` plutôt que `a` pour recueillir son age

4. Video

Cette vidéo résume toute l'activité d'apprentissage le monde des variables. Cliquez sur le lien pour y avoir accès <https://youtu.be/kkOEavyCgzk1>

5. Exercice

[Solution n°1 p 10]

Lequel de ces noms sont des noms de variable valide

<input type="checkbox"/>	ma-date-de-naissance
<input type="checkbox"/>	MadateDe-naissance
<input type="checkbox"/>	ma_date_de_naissance
<input type="checkbox"/>	MA MOYENNE
<input type="checkbox"/>	maMoyenne
<input type="checkbox"/>	ma Moyenne

¹<https://youtu.be/kkOEavyCgzk1>

6. Exercice

[Solution n°2 p 10]

Ligne 1 : Déclarer une variable filiere et initialisez la avec la valeur 'ISN'

Ligne 2 : Écrire l'instruction pour afficher le contenu de la variable filière avec la fonction print sous python 3

NB : n'ajoutez pas d'espace dans l'écriture du code

```
>>> [ ]
```

```
>>> [ ]
```

```
ISN
```

7. Exercice

[Solution n°3 p 10]

Compléter le code pour indiquer le type de cette variable

```
>>> age = 20.5
```

```
>>> [ ]
```

```
<type '[ ]'>
```

8. Exercice

[Solution n°4 p 11]

Parmi ces types de variable, lesquels sont parmi les types de bases de python

<input type="checkbox"/>	float
<input type="checkbox"/>	integer
<input type="checkbox"/>	char
<input type="checkbox"/>	bool
<input type="checkbox"/>	reel
<input type="checkbox"/>	str

II. Opérations arithmétique sur les variables en python

Objectifs

- Réaliser des opérations arithmétiques sur les variables
- Analyser et prédire le résultat d'opérations arithmétiques sur les variables

1. Opérateurs de calcul

Symbole	Opérateur	Description
+	Addition	
-	Soustraction	
*	Multiplication	
/	Division	Retourne le quotient de type réel ou flottant d'une division
//	Division entière	Retourne le quotient de type entier d'une division
%	Modulo	Retourne le reste d'une division entière
**	Exposant	

2. Priorité des opérateurs (1)

Définition

La priorité désigne l'ordre dans lequel les calculs sont effectués dans une expression complexe. La priorité dans une suite d'opération arithmétique suit la règle d'ordre **PEMDAS** (**P**arenthèse **p**rioritaire sur **E**xposant **p**rioritaire sur **M**ultiplication et **D**ivision **p**rioritaire sur **A**ddition et **S**oustraction) ; la multiplication et la division partagent le même ordre de priorité, de même l'addition et la soustraction partagent le même ordre de priorité.

Remarque : Important

Lorsque deux opérateurs possèdent le même ordre de priorité, l'ordre de calcul se fait de gauche à droite

3. Priorité des opérateurs (2)

Parenthèse

- $2*(3-1) = 2*(\mathbf{2}) = 4$
- $(2+1)**(4-1) = (\mathbf{3})*(\mathbf{3}) = 27$

Exposant

- $2**2+1 = \mathbf{4}+1 = 5$
- $3*1**100 = 3*\mathbf{1} = 3$

Multiplication et Division

- $2*3-1=6-1=5$
- $2/3-1=0.66666-1=-0.3333...$

Égalité de priorité

- $29*100//60$, **le calcul se fait de gauche à droite car les deux opérateurs ont le même ordre de priorité**. On effectue d'abord la multiplication $29*100=2900$ et ensuite se fait la division entière $2900//60=48$. Finalement on a $29*100//60=48$.
- Faisons maintenant le calcul de $29*(100//60)$. On effectue d'abord la division entière $100//60=1$ et ensuite la multiplication $29*1=29$. Finalement on a $29*(100//60)=29$.

On obtient deux résultats différents. **Le calcul s'est déroulé de droite à gauche car la parenthèse est prioritaire sur la multiplication.**

4. Opérations sur les réels

Il est possible d'appliquer sur les variables réelles tout type d'opération arithmétique

Exemple : Python 2

```
>>>nombre1 = 22
>>nombre2 = 3.5
>>nombre1 + 4
26
>>>nombre1 * nombre2
77.0 --> indique que le résultat de la multiplication est de type float
>>>nombre1 / 2.0
11.0
```

5. Opérations sur les chaînes de caractères

Addition (Concaténation)

Il est possible en Python d'utiliser **l'opérateur +** pour additionner plusieurs chaînes de caractère; on parle alors de **concaténation** (dérivé du verbe concaténer). Il s'agit de mettre bout à bout deux ou plusieurs chaînes de caractères de manière à en former une autre. **La concaténation n'est possible qu'entre des chaînes de caractère.**

Exemple : Concaténation Python version 2

```
>>>"world" + " wide" + " web"
world wide web
>>>filier = "DAS"
>>>print "Je suis en " + filier
Je suis en DAS
```

Multiplication

L'opérateur * est utilisé pour répéter n fois (où n est un entier) une chaîne de caractère. Il est impossible de multiplier deux chaînes de caractères.

Exemple : Python

```
>>>"w" * 3
www
>>>condiment = 'tomate'
>>>condiment * 2
tomatetomate
```

6. Exercice

[Solution n°5 p 11]

Quel est le résultat de ce code ?

```
>>>fruit="mangue"
>>>print fruit * 3
```

- ☐ fruitfruitfruit
- ☐ "fruitfruitfruit"
- ☐ manguemanguemangue
- ☐ "manguemanguemangue"

7. Exercice

[Solution n°6 p 11]

Compléter la zone de texte pour que ce code fonctionne correctement

```
>>>nombre1 = 5
>>>nombre2 = 
>>>print nombre1 + nombre2
12
```

8. Exercice

[Solution n°7 p 11]

Quel est le résultat de ce code ?

```
>>>x = 2
>>>y = 3
>>>y = 5
>>>x = 4
>>>print(x*y)
```


9. Exercice

[Solution n°8 p 11]

Écrivez le contenu de la variable *identite* ?

```
>>>nom = "YAO"  
>>>prenom = " Jean Charles"  
>>>identite = nom + prenom  
>>>print(identite)
```

III. 3. Exercice d'application

Objectifs

- Faire une synthèse des deux activités d'apprentissage précédente
- Réaliser un script Python

A l'issue des points précédents, on a appris à déclarer des variables en python et surtout à leur donner un nom significatif. Maintenant nous allons par la pratique réaliser un petit programme pour nous aider à manipuler les variables.

1. Sujet

Le but de cet exercice est d'écrire un programme qui permet de convertir un nombre donnée de seconde en heure, minute et seconde. L'apprenant doit faire une réflexion personnelle avant de regarder la solution qui sera proposée.

2. Analyse et explication

Notre programme doit recevoir comme donnée d'entrée « *secondes* » un nombre entier fourni au clavier par l'utilisateur. L'idée est de segmenter la variable *secondes* en heure, minute et seconde.

On sait que 1 minute = 60 secondes et que 1 heure = 60 minutes par conséquent 1 heure = 3600 secondes.

D'abord pour obtenir le nombre d'heure on doit effectuer une division entière de la variable *secondes* par 3600. Le quotient de cette division représente le nombre d'heure.

Ensuite pour obtenir le nombre de minute on doit récupérer le reste de la division entière de la variable *secondes* par 3600 qui est « *secondes_restantes_1* » et effectuer la division entière de la variable *secondes_restantes_1* par 60. Le quotient de cette division représente le nombre de minute.

Enfin le reste de la division entière de la variable `secondes_restantes_1` par 60 représente le nombre de seconde.

3. Correction

Vidéo

Cliquez sur le lien ci-dessous pour avoir accès à la vidéo

<https://youtu.be/314KC71uejE2>

Conclusion

Ce cours de manipulation de variable à permis de comprendre des notions de déclaration, affectation et opérations sur les variables. Les TP ont permis de se familiariser avec l'interpréteur interactif de Python et de remarquer la petite différence entre la version 2 et la version 3 de python concernant la gestion des variables. La prochaine leçon sera consacrée aux structures conditionnelles.

Solution des exercices

> Solution n°1 (exercice p. 4)

Lequel de ces noms sont des noms de variable valide

<input type="checkbox"/>	ma-date-de-naissance
<input type="checkbox"/>	MadateDe-naissance
<input checked="" type="checkbox"/>	ma_date_de_naissance
<input type="checkbox"/>	MA MOYENNE
<input checked="" type="checkbox"/>	maMoyenne
<input type="checkbox"/>	ma Moyenne

> Solution n°2 (exercice p. 5)

Ligne 1 : Déclarer une variable filiere et initialisez la avec la valeur 'ISN'

Ligne 2 : Écrire l'instruction pour afficher le contenu de la variable filière avec la fonction print sous python 3

NB : n'ajoutez pas d'espace dans l'écriture du code

```
>>>filiere='ISN'
```

```
>>>print(filiere)
```

```
ISN
```

> Solution n°3 (exercice p. 5)

Compléter le code pour indiquer le type de cette variable

```
>>>age = 20.5
```

²<https://youtu.be/314KC71uejE>

```
>>>type(age)
<type 'float'>
```

> **Solution n°4** (exercice p. 5)

Parmi ces types de variable, lesquels sont parmi les types de bases de python

<input checked="" type="checkbox"/>	float
<input type="checkbox"/>	integer
<input type="checkbox"/>	char
<input checked="" type="checkbox"/>	bool
<input type="checkbox"/>	reel
<input checked="" type="checkbox"/>	str

> **Solution n°5** (exercice p. 8)

Quel est le résultat de ce code ?

```
>>>fruit="mangue"
>>>print fruit * 3
```

<input type="radio"/>	fruitfruitfruit
<input type="radio"/>	"fruitfruitfruit"
<input checked="" type="radio"/>	manguemanguemangue
<input type="radio"/>	"manguemanguemangue"

> **Solution n°6** (exercice p. 8)

Compléter la zone de texte pour que ce code fonctionne correctement

```
>>>nombre1 = 5
>>>nombre2 = 7
>>>print nombre1 + nombre2
12
```

> **Solution n°7** (exercice p. 8)

Quel est le résultat de ce code ?

```
>>>x = 2
>>>y = 3
>>>y = 5
>>>x = 4
>>>print(x*y)
20
```

> **Solution n°8** (exercice p. 9)

Écrivez le contenu de la variable identite ?

```
>>>nom = "YAO"
>>>prenom = " Jean Charles"
>>>identite = nom + prenom
```

```
>>>print(identite)
```

```
YAO Jean Charles
```