

# Variables et fonctions du jour 2

## Fonctions intégrées

Dans Python, nous avons beaucoup de fonctions intégrées. Les fonctions intégrées sont disponibles à l'échelle mondiale pour votre utilisation, ce qui signifie que vous pouvez utiliser les fonctions intégrées sans importation ni configuration. Certaines des fonctions intégrées Python les plus couramment utilisées sont les

Suivant: `print()`, `len()`, `type()`, `int()`, `float()`, `str()`, `input()`, `list()`, `dict()`, `min()`, `max()`, `sum()`, `e()`, `open()`, `fil e()`, `open()`, `fil` `help()`, et `dir()`. Dans le tableau suivant, vous verrez une liste exhaustive des fonctions intégrées Python tirées de la documentation Python.

Built-in Functions				
<code>abs()</code>	<code>delattr()</code>	<code>hash()</code>	<code>memoryview()</code>	<code>set()</code>
<code>all()</code>	<code>dict()</code>	<code>help()</code>	<code>min()</code>	<code>setattr()</code>
<code>any()</code>	<code>dir()</code>	<code>hex()</code>	<code>next()</code>	<code>slice()</code>
<code>ascii()</code>	<code>divmod()</code>	<code>id()</code>	<code>object()</code>	<code>sorted()</code>
<code>bin()</code>	<code>enumerate()</code>	<code>input()</code>	<code>oct()</code>	<code>staticmethod()</code>
<code>bool()</code>	<code>eval()</code>	<code>int()</code>	<code>open()</code>	<code>str()</code>
<code>breakpoint()</code>	<code>exec()</code>	<code>isinstance()</code>	<code>ord()</code>	<code>sum()</code>
<code>bytearray()</code>	<code>filter()</code>	<code>issubclass()</code>	<code>pow()</code>	<code>super()</code>
<code>bytes()</code>	<code>float()</code>	<code>iter()</code>	<code>print()</code>	<code>tuple()</code>
<code>callable()</code>	<code>format()</code>	<code>len()</code>	<code>property()</code>	<code>type()</code>
<code>chr()</code>	<code>frozenset()</code>	<code>list()</code>	<code>range()</code>	<code>vars()</code>
<code>classmethod()</code>	<code>getattr()</code>	<code>locals()</code>	<code>repr()</code>	<code>zip()</code>
<code>compile()</code>	<code>globals()</code>	<code>map()</code>	<code>reversed()</code>	<code>__import__()</code>
<code>complex()</code>	<code>hasattr()</code>	<code>max()</code>	<code>round()</code>	

Laissez-nous ouvrir le pyth

sur la coquille et commencer à utiliser certains des **Bois plus utiles**

```
Last login: Wed Nov 20 20:41:35 on ttys002
asabeneh@Asabeneh:~$ python
Python 3.7.5 (default, Nov 1 2019, 02:16:32)
[Clang 11.0.0 (clang-1100.0.33.8)] on darwin
Type "help", "copyright", "credits" or "license" for more information.
>>> print('Hello, World!') # it prints the text value Hello, World!
Hello, World!
>>> len('Hello, World!') # it counts the number of characters including space
13
>>> type('Hello, World!') # it checks the data type
<class 'str'>
>>> str(10) # it converts number to string
'10'
>>> int('10') # it converts to number
10
>>> float(10) # it converts integer to decimal
10.0
>>> input('Enter your name:') # it takes user input
Enter your name:Asabeneh
'Asabeneh'
>>> exit()
```

Laissez-nous pratiquer davantage en utilisant différentes fonctions intégrées

```
Last login: Wed Nov 20 20:42:55 on ttys002
asabeneh@Asabeneh:~$ python
Python 3.7.5 (default, Nov 1 2019, 02:16:32)
[Clang 11.0.0 (clang-1100.0.33.8)] on darwin
Type "help", "copyright", "credits" or "license" for more information.
>>> help('keywords') # prints all python reserved words

Here is a list of the Python keywords. Enter any keyword to get more help.

False          class          from           or
None           continue      global         pass
True           def           if             raise
and            del           import         return
as             elif          in             try
assert         else          is             while
async          except        lambda         with
await          finally       nonlocal      yield
break          for           not

>>> help(str) # give informaton about string

>>> dir(str) # give information about string
['__add__', '__class__', '__contains__', '__delattr__', '__dir__', '__doc__', '__eq__', '__format__', '__ge__', '__getattribute__', '__getitem__', '__getnewargs__', '__gt__', '__hash__', '__init__', '__init_subclass__', '__iter__', '__le__', '__len__', '__lt__', '__mod__', '__mul__', '__ne__', '__new__', '__reduce__', '__reduce_ex__', '__repr__', '__rmod__', '__rmul__', '__setattr__', '__sizeof__', '__str__', '__subclasshook__', 'capitalize', 'casefold', 'center', 'count', 'encode', 'endswith', 'expandtabs', 'find', 'format', 'format_map', 'index', 'isalnum', 'isalpha', 'isascii', 'isdecimal', 'isdigit', 'isidentifier', 'islower', 'isnumeric', 'isprintable', 'isspace', 'istitle', 'isupper', 'join', 'ljust', 'lower', 'lstrip', 'maketrans', 'partition', 'replace', 'rfind', 'rindex', 'rjust', 'rpartition', 'rsplit', 'rstrip', 'split', 'splitlines', 'startswith', 'strip', 'swapcase', 'title', 'translate', 'upper', 'zfill']
>>>
```

Comme vous pouvez le voir sur le terminal ci-dessus, Python a des mots réservés. Nous n'utilisons pas de mots réservés pour déclarer des variables ou des fonctions. Nous couvrirons les variables dans la section suivante.

Je crois, vous connaissez maintenant les fonctions intégrées. Faisons une autre pratique des fonctions intégrées et nous passerons à la section suivante.

```
asabeneh — Python — 80x16
Last login: Wed Nov 20 21:12:11 on ttys000
asabeneh@Asabeneh:~$ python
Python 3.7.5 (default, Nov 1 2019, 02:16:32)
[Clang 11.0.0 (clang-1100.0.33.8)] on darwin
Type "help", "copyright", "credits" or "license" for more information.
>>> min(20, 30, 40, 50) # gives the minimum value
20
>>> max(20, 30, 40, 50) # gives the maximum value
50
>>> min([20, 30, 40, 50]) # it takes list as an argument and return min
20
>>> max([20, 30, 40, 50]) # it takes list as an argument and return max
50
>>> sum([20, 30, 40, 50]) # it takes only list as an argument and return the sum
140
>>> exit()
```

## Variables

Les variables stockent les données dans une mémoire informatique. Des variables mnémoniques sont recommandées à utiliser dans de nombreux langages de programmation. Une variable mnémonique est un nom de variable qui peut être facilement rappelé et associé. Une variable fait référence à une adresse mémoire dans laquelle les données sont stockées. Numéro Au début, caractère spécial, les traits d'union ne sont pas autorisés lors du nom d'une variable. Une variable peut avoir un nom court (comme x, y, z), mais un nom plus descriptif (FirstName, LastName, Age, Country) est fortement recommandé.

### Règles de nom de variable Python

- Un nom de variable doit commencer par une lettre ou le caractère de soulignement
- Un nom de variable ne peut pas commencer par un nombre
- Un nom de variable ne peut contenir que des caractères alpha-numériques et des sous-traits (A-Z, 0-9 et \_)
- Les noms variables sont-sensibles (FirstName, FirstName, FirstName et FirstName) sont des variables différentes) Variables) sont différentes) Variables) Différentes).

Voici un exemple de noms de variables valides:

```
Firstname Last
name Age Country City First
_name
```

nom de famille

```
Capital_City _if # Si nous voulons utiliser le mot réservé comme variable anné  
e_2021 Année2021 Current_Year_2021 Birth_year num1 num2
```

Noms de variables non valides

```
Premier-nom Fir  
st @ Name First  
$ Name num-1 1  
NUM
```

Nous utiliserons le style de dénomination de variable Python standard qui a été adopté par de nombreux développeurs Python. Les développeurs Python utilisent la convention de dénomination de la variable Case Snake (Snake\_Case). Nous utilisons un caractère de soulignement après chaque mot pour une variable contenant plus qu'un mot (par exemple, premier\_name, last\_name, moteur\_rotation\_speed). L'exemple ci-dessous est un exemple de dénomination standard des variables, un soulignement est requis lorsque le nom de la variable est supérieur à un mot.

Lorsque nous attribuons un certain type de données à une variable, elle est appelée déclaration de variable. Par exemple, dans l'exemple ci-dessous, mon prénom est attribué à une variable First\_name. Le signe égal est un opérateur d'affectation. Attribuer des moyens de stockage de données dans la variable. Le signe égal dans Python n'est pas l'égalité comme en mathématiques.

*Example:*

```
# Variables dans Python first_name = 'Asabeneh' last_name = 'yetayeh' cou  
ntry = 'Finland' City = 'Helsinki' Age = 250 IS_MARRIED = 'Python'] Pers  
on_info = {'FirstName': 'Asabeneh', 'LastName': 'YEAYEH', 'COUNTRY':  
'Finlande', 'City': 'Helsinki'}
```

Utilisons les fonctions intégrées *print()* et *len()*. La fonction d'impression prend un nombre illimité d'arguments. Un argument est une valeur que nous pouvons être transmises ou mettre à l'intérieur de la parenthèse de fonction, voir l'exemple ci-dessous.

Exemple:

```
Print («Bonjour, monde!») # Le texte bonjour, monde! est un argument imprimé ('Hello', ',', 'World', '!') # Cela peut prendre plusieurs arguments, quatre arguments ont été adoptés imprimés (Len ('Hello, World!')) # Il ne faut qu'un seul argument
```

Imprimez et trouvons également la longueur des variables déclarées en haut:

Exemple:

```
# Impression des valeurs stockées dans les variables

print ('prénom:', first_name) print ('le prénom Longueur:', Len (first_name))
print ('nom de famille:', last_name) print ('le nom de famille:', Len (last_name)) print ('country:', country) print ('City:', City) print ('Âge:', Âge) imprimer ('marié:', is_married) print ('compétences:', compétences) imprimer ('Person Information:', personne_info)
```

## Déclarant plusieurs variables dans une ligne

Plusieurs variables peuvent également être déclarées en une seule ligne:

Exemple:

```
First_name, Last_name, Country, Age, Is_Married = 'Asabeneh', 'YEAYEH', 'Helsinki', 250, True
print (first_name, last_name, country, âge, is_married) print ('prénom:', first_name)
print ('nom de famille:', last_name) print ('country:', country) print ('Âge:', Âge) print ('marié:', is_married)
```

Obtenir une entrée utilisateur à l'aide de la fonction intégrée *input()*. Attribuez les données que nous obtenons d'un utilisateur dans les variables First\_name et Age. Exemple:

```
First_name = entrée ('Quel est votre nom:') Âge = entrée ('Quel âge avez-vous?')
```

## Types de données

Il existe plusieurs types de données dans Python. Pour identifier le type de données, nous utilisons la fonction intégrée *type*. Je voudrais vous demander de vous concentrer très bien sur la compréhension des différents types de données. En ce qui concerne la programmation, il s'agit de types de données. J'ai introduit des types de données au tout début et cela revient, car chaque sujet est lié aux types de données. Nous couvrirons plus en détail les types de données dans leurs sections respectives.

## Vérification des types de données et casting

- Vérifier les types de données: Pour vérifier le type de données de certaines données / variable, nous utilisons les exemples *type*:

```
# Différents types de données Python # Déclarons les variables avec divers types de données
first_name = 'Asabeneh' # str Last_name = 'yetayeh' # Str Country = 'Finland' # Str
print(type('asabeneh')) # str print(type(first_name)) # str print(type(10)) # int print
(type(3.14)) # float print(type(1 + 1j)) # complexe print(type)) # bool print(type([1,
2, 3, 4])) # list print(type({'nom': 'asabeneh'})) # dic print(type((1,2))) # Tuple im
print(type(zip([1,2], [3,4]))) # zip
```

- Coulage: conversion d'un type de données en un autre type de données. Nous utilisons *int()*, *float()*, *str()*, *list*, *set* Lorsque nous faisons des opérations arithmétiques, les nombres de chaîne doivent d'abord être convertis en int ou float sinon il renverra une erreur. Si nous concaténons un nombre avec une chaîne, le nombre doit d'abord être converti en chaîne. Nous parlerons de la concaténation dans la section des cordes.

## Exemples:

```
# int to float
num_int = 10
print('num_int', num_int)          # 10
num_float = float(num_int)
print('num_float:', num_float)     # 10.0
```

```
# float vers int Gravity = 9.81 imprimer (int (Gravity))
# 9

# int à str num_int = 10 print (num_int) # 10 num_str = str
(num_int) print (num_str) # '10'

# str to int ou float num_str = '10.6' num_float = float (num_str)
print (' num_float ', float (num_str)) # 10.6 num_int = int (num_f
loat) print (' num_int ', int (num_int)) # 10

# str to lister First_name = 'asabeneh' print (first_name) # 'ASABENEH' First_Name_To_
List = List (First_Name) Print (First_Name_To_List) # ['A', 'S', 'A', 'B', 'E', 'n', 'e', 'H' ]
```

## Nombres

Types de données numériques dans Python:

1. ENTegers: ENTeger (négatif, zéro et positif) Exemple de nombres: ... -3, -2, -1, 0, 1, 2, 3 ...
2. Nombres de points flottants (numéros décimaux) Exemple: ... -3.5, -2.25, -1.0, 0, 0, 1.1, 2.2, 3.5 ...
3. Numéro de complexe Exemple: 1 + j, 2 + 4j, 1 - 1j

Vous êtes génial. Vous venez de terminer les défis du jour 2 et vous avez deux pas en avant pour la grandeur. Faites maintenant quelques exercices pour votre cerveau et vos muscles.

## Exercices - Jour 2

### Exercices: niveau 1

1. À l'intérieur de 30 jours, créez un dossier appelé day\_2. À l'intérieur de ce dossier, créez un fichier nommé variables.py 2. Écrivez un commentaire Python disant «Jour 2: 30 Days of Python Programming» 3. Déclarez une variable de prénom et attribuez-lui une valeur 4. Déclarez une variable de nom de famille et attribuez-lui une valeur 5. Déclarez une variable de nom complet et attribuez une valeur à lui 6. Déclare une variable de pays et y attribuez une valeur à une valeur de 7. Déclarer une variable municipale et attribuer une valeur à un pays. et attribuer une valeur à lui 9. Déclarez une variable an et lui attribuez une valeur 10. Déclarez une variable est\_Married et attribuez-lui une valeur 11. Déclarez une variable is\_true et attribuez-lui une valeur 12. Déclarez une variable IS\_light\_on et y attribuez une valeur 13. Déclarer la variable multiple sur une seule ligne

### Exercices: niveau 2

1. Vérifiez le type de données de toutes vos variables à l'aide de Type () Fonction intégrée 2. À l'aide de la fonction intégrée len(), trouvez la longueur de votre prénom 3. Comparez la longueur de votre prénom et votre nom de famille 4. Déclare 5 comme num\_one et 4 comme num\_two 5. Add num\_one et num\_two et attribuez la valeur à une variable 6. Soustraire Num\_two à partir de NUM\_ON et attribuez la valeur à une variable 6. Soustraire. Multiplier num\_two et num\_one et attribuer la valeur à un produit de variable 8. Divisez num\_one par num\_two et attribuez la valeur à une division variable 9. Utilisez la division du module pour trouver num\_two divisé par num\_one et attribuer la valeur à une variable reste 10. Trouver le plan Variable Floor\_Division



12. Le rayon d'un cercle est de 30 mètres.

je. Calculez la zone d'un cercle et attribuez la valeur à un nom de variable de *area\_of\_circle* ii. Calculez la circonférence d'un cercle et attribuez la valeur à un nom de variable de *circum\_of\_circle* iii. Prenez le rayon comme entrée utilisateur et calculez la zone.

13. Utilisez la fonction d'entrée intégrée pour obtenir le prénom, le nom de famille, le pays et l'âge d'un utilisateur et stockez la valeur à leurs noms de variables correspondants

14. Exécutez l'aide («mots clés») dans Python Shell ou dans votre fichier pour vérifier les mots ou mots clés réservés Python

 Félicitations!