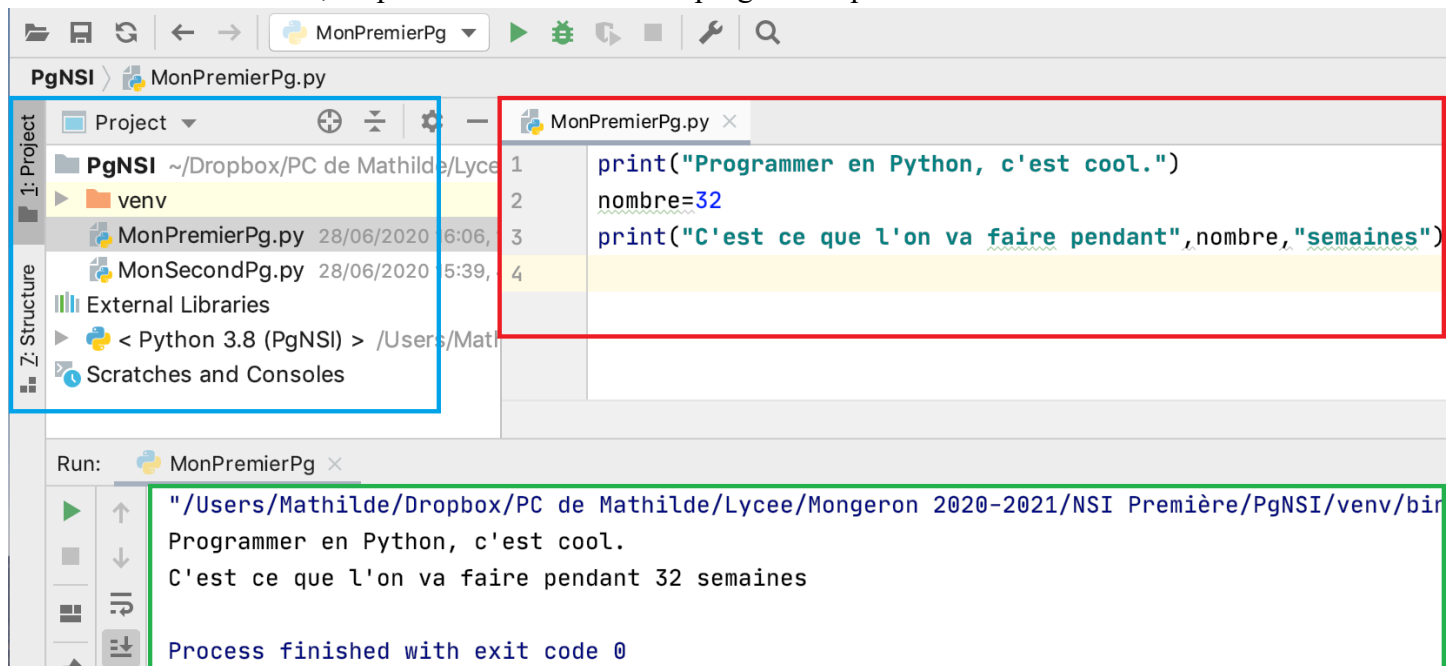


Nous allons travailler dans un éditeur qui s'appelle « PYCHARM » qui est un logiciel libre.

Il est gratuit et fonctionne sous Windows, MacOS et Linux.

Pour l'installer, il faut d'abord installer Python3.

Avec cet environnement, on peut facilement éditer le programme puis l'exécuter.



La fenêtre de l'éditeur est composée de plusieurs zones :

Zone bleue : Explorateur de fichiers

Zone rouge : Saisie du programme

Zone verte : Console Python où s'exécute le programme

### 1) Les sorties avec `print()`

La fonction `print` permet d'afficher un message dans la console.

Le résultat s'affiche dans la console.

**Exemple** : Saisissez et exécutez le programme suivant :

```
print("Programmer en Python, c'est cool.")
nombre = 32
print("C'est ce que l'on va faire pendant", nombre, "semaines")
```

Avec la première ligne, on affiche dans la console la phrase « Programmer en python c'est cool. »

Avec la troisième ligne, on affiche une phrase qui est modifiée suivant la valeur de la variable `nombre`.

### 2) Les variables

Pièces maîtresses des programmes, les variables servent à stocker des valeurs.

Pour que les programmes soient plus simples à comprendre, elles doivent avoir des noms explicites en lien avec les données qu'elles contiennent. Pour créer un nom de variable on peut utiliser les lettres de l'alphabet non accentuées, les chiffres et le caractère « `_` », pas d'autres caractères spéciaux qui sont réservés.

On affecte une variable (on lui donne une valeur) avec le signe `=`.

```
Nombre1 = 32
Nombre2 = 3.2
```

## a) Les nombres

Exemple : Saisissez et exécutez le programme suivant :

```
a = 2
b = 2.0
print(a, type(a))
print(b, type(b))
```

La fonction `type` permet d'afficher le type (genre) des variables.

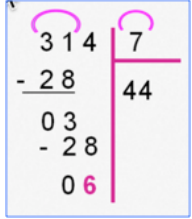
En Python, les variables sont « typées » automatiquement lors de leur première affectation.

La variable `a` est un nombre entier (integer en anglais).

La variable `b` est un nombre décimal (float en anglais).

### Opérations sur les nombres

<code>+</code> <code>-</code> <code>*</code> <code>/</code>	Opérations classiques
<code>a**b</code>	La puissance $a^b$
<code>//</code> <code>%</code>	Quotient et reste de la division euclidienne


$314 // 7 = 44$
$314 \% 7 = 6$

### Exercice 1

1) Calculer :

$15 // 2 =$        $31 // 2 =$        $54 // 2 =$        $125 // 2 =$

$15 \% 2 =$        $31 \% 2 =$        $54 \% 2 =$        $125 \% 2 =$

2) Compléter la phrase suivante :

Si  $n$  est pair alors  $n \% 2$  vaut      tandis que si  $n$  est impair alors  $n \% 2$  vaut



## b) Les chaînes de caractères

Les chaînes de caractères sont formées de symboles alphanumériques ou autres.

On les signale au choix avec des guillemets simples `'`, doubles `"` ou triples `'''`.

Exemple : Saisissez et exécutez le programme suivant :

```
phrase = "J'@lme la NSI"
print(phrase, type(phrase))
print(phrase[0])
print(phrase[11])
```

La variable `phrase` une chaîne de caractères (string en anglais).

On accède aux différents caractères en indiquant le rang du caractère entre crochets.

Attention, le premier caractère à pour **rang 0**.

### Exercice 2

Pour la phrase `"J'@lme la NSI"`, compléter le tableau suivant

rang	0	1	2	3	4	5	6	7	8	9	10	11	12
caractère													



### Opérateurs sur les chaînes de caractères

<code>a + b</code>	Concatène (colle) les deux chaînes de caractères <code>a</code> et <code>b</code>
<code>a * n</code>	Concatène $n$ fois la chaîne la chaîne <code>a</code>
<code>str(x)</code>	Transforme un nombre entier ou décimal <code>x</code> en chaîne de caractères
<code>\n</code>	Est le caractère « retour chariot » qui permet de passer à la ligne suivante
<code>len(chaîne)</code>	Retourne la longueur de la chaîne de caractère.

Exercice 3 : En le complétant, saisissez puis exécutez le programme suivant :

```
n = 5
phrase = "le triple de " +  + " est " +  + "\n"
print(phrase*n)
```



### c) Les variables booléennes

Les variables booléennes (boolean en anglais) ne peuvent prendre que deux valeurs True ou False. Ne pas oublier la majuscule.

Exemple : Saisissez et exécutez le programme ci-dessous :

```
test = 4<3
print(test, type(test))
print(test == False)
```

Ces variables sont surtout utilisées pour réaliser les tests (instructions « Si » et « Tant que »).

#### Opérateurs sur les variables booléennes

Retourne la valeur **True** si :

<b>A == B</b>		A et B sont égales	
<b>A &lt; B</b>	<b>A &gt; B</b>	A est strictement inférieure à B	A est strictement supérieure à B
<b>A &lt;= B</b>	<b>A &gt;= B</b>	A est inférieure ou égale à B	A est supérieure ou égale à B
<b>not A</b>		A est fausse	
<b>A and B</b>		A et B sont vraies en même temps	
<b>A != B</b>		A et B sont différentes	
<b>A or B</b>		A est vraie ou B est vraie ou A et B sont vraies	

#### Exercice 4

À l'aide du programme Python suivant, compléter la table de vérité :

```
A, B = True, True
C = (A and not B) or (not A and B)
print(C)
```

A	B	(A and not B) or (not A and B)
True	True	
True	False	
False	True	
False	False	



On verra plus tard que cette opération logique est la « porte XOR ».

### 3) Les entrées avec **input()**

La fonction **input** permet de récupérer une valeur saisie par l'utilisateur dans la console. On lui passe en paramètre une chaîne de caractères pour informer l'utilisateur de ce que l'on attend de lui.

#### Exercice 5

Saisissez et exécutez le programme ci-dessous.

```
nombre1 = input("Entrez un premier nombre : ")
nombre2 = input("Entrez un second nombre : ")
somme = nombre1 + nombre2
print(nombre1, "+", nombre2, "=", somme)
```

- 1) L'affichage correspond-il à ce que vous attendiez ?
- 2) Affichez les types de `nombre1` et `nombre2`.  
Les types de `nombre1` et `nombre2` expliquent-ils l'affichage obtenu ?
- 3) Modifier alors le programme pour obtenir la somme des deux entiers.




#### À retenir

La fonction **input** retourne toujours une chaîne de caractères. Il faut utiliser les fonctions **int** ou **float** pour obtenir des nombres.

#### 4) Les tests avec l'instruction « if »

<b>if</b> condition: BLOC	Effectue tout le BLOC d'instructions dès que la condition est vérifiée. Attention c'est l'indentation qui délimite le début et la fin du BLOC.
<b>if</b> condition: BLOC1 <b>else:</b> BLOC2	Effectue tout le BLOC1 d'instructions dès que la condition est vérifiée, dans le cas contraire, c'est le BLOC2 qui s'exécute.
<b>if</b> condition: BLOC1 <b>elif</b> condition: BLOC2 <b>else:</b> BLOC3	Effectue tout le BLOC1 d'instructions dès que la condition est vérifiée, dans le cas contraire, si la seconde condition est vérifiée c'est le BLOC2 qui s'exécute, dans le cas contraire, c'est le BLOC3 qui s'exécute.

**Attention aux indentations** : Pour les créer, on utilise la touche  ou 4 espaces.

Exemple : Saisissez et exécutez le programme ci-dessous.

```
age = int(input("Entrer votre âge"))
if age <= 12:
    print("Ce jeu est déconseillé aux moins de 12 ans !")
else:
    print("Début de la partie !")
```

#### Exercice 6

1) Saisissez et exécutez plusieurs fois le programme suivant.

```
nombre = int(input("Entrer votre chiffre bonheur : "))
if nombre == 7:
    print("Vous avez choisi le 7, comme 60 % des gens.")
print("C'est un bon choix.")
```

2) Modifiez et exécutez plusieurs fois le programme suivant.

```
nombre = int(input("Entrer votre chiffre bonheur : "))
if nombre == 7:
    print("Vous avez choisi le 7, comme 60 % des gens.")
print("C'est un bon choix.")
```

3) Pourquoi ces deux programmes ne donnent-ils pas le même résultat ?



#### Exercice 7

Complétez, saisissez puis exécutez le programme suivant :

```
nombre = int(input("Entrer un nombre entier : "))
if :
    print("Le nommbre", nombre, "est pair.")
:
    print("Le nommbre", nombre, "est impair.")
```



#### Exercice 8 : Je vérifie que j'ai bien compris

1) Complétez le tableau suivant :

Affectation	v = 5	v = "5"	v = 5.0	v = 3>5	v = str(5)
Type de v					

2) Donner les valeurs de 27//2 et 27%2

3) Si texte = "informatique", alors texte[5] vaut :

4) L'appel de print("5"\*3) affiche dans la console le message :

5) Après la commande v = input("Entrer un entier"), le type de la variable v est :



### Exercice 9

Écrire un programme qui demande une température nommée `temperature` et affiche l'état de l'eau à cette température, c'est-à-dire "SOLIDE", "LIQUIDE" ou "GAZEUX".



### Exercice 10

Dans un magasin de sport, pour inciter les clients à dépenser un peu plus, on leur propose une réduction de 10 % à partir de 100 euros d'achats et de 25% au-delà de 300 euros.

Réalisez un programme qui demande à l'utilisateur le montant de ses achats et lui permet de calculer le montant qu'il devra finalement régler.



### Exercice 11

Une société de location de vélos propose les tarifs suivants :

- 7,50 € par heure pour les 2 premières heures
- 5,00 € par heure pour les 3 suivantes
- 3,00 € par heure pour les heures au-delà de 5 heures.

Écrire un programme qui à partir du nombre d'heures de locations et affiche le montant à payer.



### Exercice 12

La masse et la taille d'un individu étant respectivement exprimées en *kg* et *m*, on calcule l'indice de masse corporelle d'un individu avec la formule  $IMC = \frac{masse}{taille^2}$ .

On considère qu'une personne est de corpulence « normale » si son IMC est compris entre 18 et 25.

Au-dessus de 25 elle est en surpoids, en-dessous de 18 elle est maigre.

Créez un programme de coaching qui :

- ✓ demande à l'utilisateur sa taille et son poids
- ✓ calcule son IMC et lui affiche un conseil.



### Exercice 13

Écrivez un programme, qui étant donnée la moyenne d'un élève au BAC, affiche son résultat :

[0, 8[	[8, 10[	[10, 12[	[12, 14[	[14, 16[	[16, 18[	[18, 20]
Recalé	Oral	Passable	Assez Bien	Bien	Très bien	Félicitations



### Exercice 14

Écrire un programme qui demande deux entiers et teste si le plus grand est divisible par le plus petit.

### Exercice 15



Un distributeur propose des pièces et billets de 1, 2, 5, 10 et 20 euros.

On suppose que le stock de pièces et billets est illimité.

Écrire un programme qui distribue la somme demandée avec le moins de billets possibles.

On pourra penser à utiliser les opérations // et %.



### Exercice 16

Un permis de chasse à points remplace désormais le permis de chasse traditionnel.

Chaque chasseur possède au départ un capital de 10 points.

S'il tue une poule, il perd 1 point, 3 points pour un chien, 5 points pour une vache et 10 points pour un ami.

S'il perd plus de 10 points, il doit déboursier 10 euros par point supplémentaire.

Le permis coûte 200 euros.

Écrire un programme qui récupère le nombre de poules, chiens, vaches ou amis tués et affiche la somme totale que le chasseur aura déboursée pour chasser.



### Exercice 17

Effectuez une petite recherche sur les chiffres romains.

Écrivez un programme qui convertit un entier compris entre 1 et 3999 en chiffres romains.

