Institut National Universitaire Champollion

Chapitre 1

Notion de programme variables & opérations

Plan du chapitre

Notion de programme - langages

Notion de programme - langages

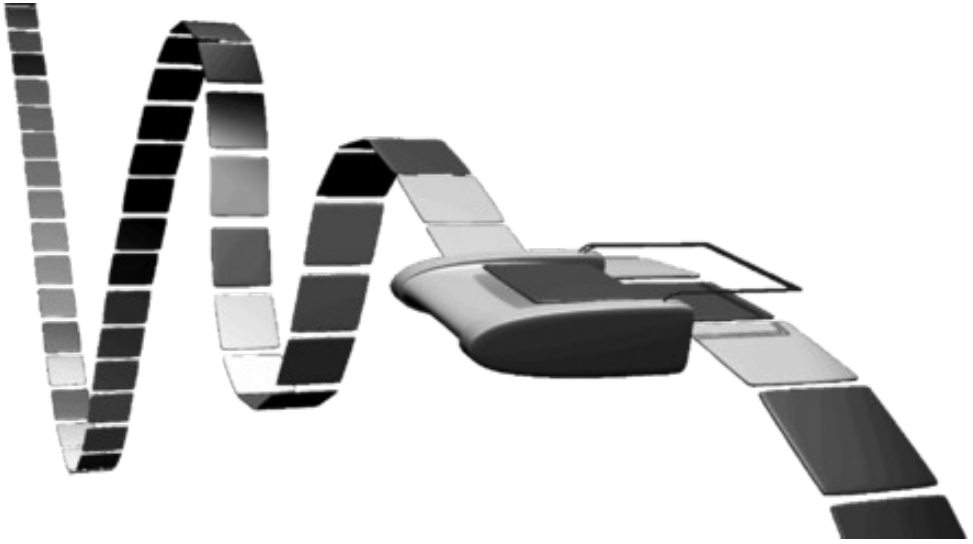
Définition :

Un programme est une suite d'instructions qui spécifient étape par étape les opérations à exécuter par un ordinateur.

Source wikipédia : https://fr.wikipedia.org/wiki/Programme_informatique

Notion de programme - langages

Machine de Turing :



- Un ruban divisé en cases contenant les symboles d'un alphabet.
- Une tête de lecture/écriture
- Un registre d'« état » qui mémorise l'état de la machine
- Une table d'action qui indique le symbole à écrire, comment déplacer la tête, et quel est le nouvel état en fonction du symbole lu.

Source wikipédia : https://fr.wikipedia.org/wiki/Machine_de_Turing

Notion de programme - langages

Pour concevoir un programme ayant la puissance de calcul d'une machine de Turing, il faut et il suffit des notions de :

- des variables,
- la séquence,
- l'affectation,
- l'alternative,
- la boucle.

Tous les langages dit "impératifs" contiennent ces notions. Les autres notions sont là pour simplifier la vie du programmeur.

Notion de programme - langages

Types de langages.

- impératifs

- fonctionnels

- logiques

- orientés objets

Notion de programme - langages

Les langages impératifs

Le programme est une suite d'instructions exécutées les unes après les autres.

Par exemple le **langage C** :

```
#include <stdio.h>
#define TAUX 6.55957

int main () {
   float francs;

   francs=0;
   while (francs<=10) {
      printf("%4.1f francs = %.2f euros\n",francs,francs/TAUX);
      francs=francs+0.5;
   }

   return 0;
}
```

Notion de programme - langages

Les langages fonctionnels

Le programme est une fonction qui utilise d'autres fonctions. Le résultat d'une exécution est le résultat de l'évaluation de la fonction.

Exemple **CAML** :

successeur : $N \rightarrow N$

$x \mapsto x + 1$

Programme :

```
# let successeur (x) = x + 1;;
successeur : int -> int = <fun>
```

éxecution :

```
# successeur (2);;
- : int = 3
```

Notion de programme - langages

Les langages logiques

Un programme est une suite d'axiomes, de demandes et de règles de déduction. Son exécution est une suite de recherches.

Exemple de programme **Prolog** :

```
frère_ou_soeur(X,Y) :- parent(Z,X), parent(Z,Y), X \= Y.
parent(X,Y) :- père(X,Y).
parent(X,Y) :- mère(X,Y).
mère(trude, sally).
père(tom, sally).
père(tom, erica).
père(mike, tom).
```

son exécution :

```
?- frère_ou_soeur(sally, erica).
     oui.
```

Notion de programme - langages

Les langages objets

Un programme est un objet qui communique avec d'autres objets par échange de messages. Les procédures (méthodes) agissent sur les données et le tout est cloisonné dans des objets. Les objets sont définis par des classes.

Exemple de programme java :

```
public class Chaine{

  private String name;

  public Chaine(String name){
      this.name = name;
  }

  public String getName(){
      return(name);
  }

  public void setName (String name){
      this.name = name;
  }

  public static void main(String[] args){
      Chaine ch;
      ch = new Chaine("Bonjour !");
      System.out.println(ch.getName());
      ch.setName("Au revoir !");
      System.out.println(ch.getName());
  }
}
```

Notion de programme - langages

Exécution d'un programme

L'exécution d'un programme se passe par

- Un compilateur
- Un interpréteur

Notion de programme - langages

Exécution d'un programme

L'exécution d'un programme se passe par

- Un compilateur
- Un interpréteur

Notion de programme - langages

Exécution d'un programme : compilateur

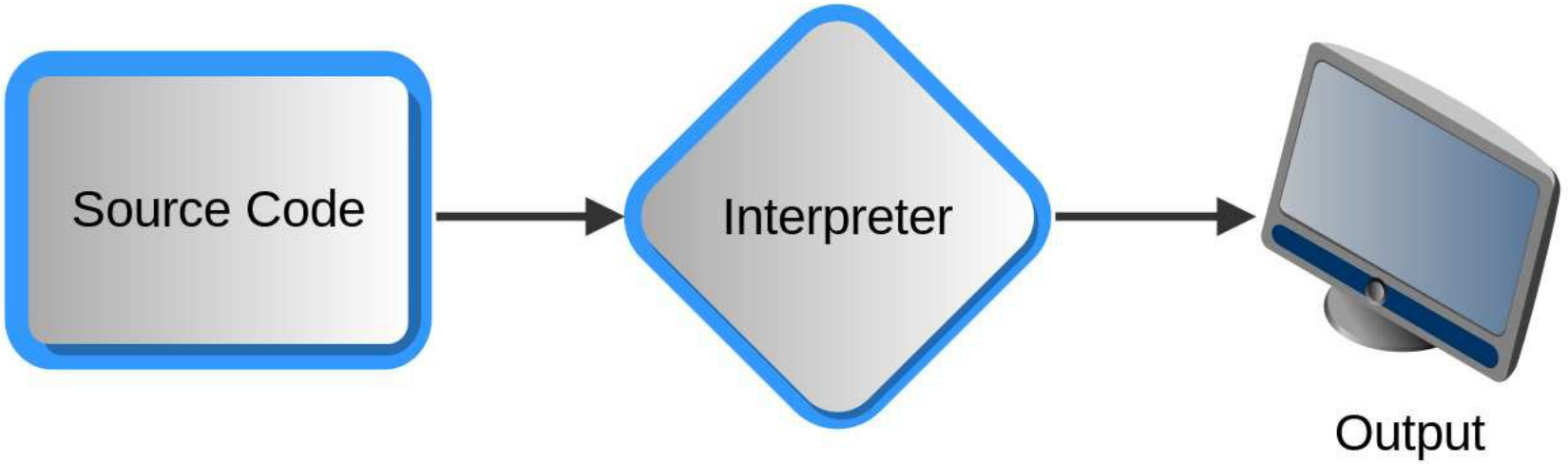
Un compilateur est un programme qui traduit le texte (code source) dans un langage qui permettra son exécution, tel le langage machine, le bytecode ou le langage assembleur.

Exemples : C, C++, Java (javac), Pascal ….

Notion de programme - langages

Exécution d'un programme : interpréteur

Un interpréteur est Un programme qui exécute les instructions demandées. Il joue le même rôle qu'une machine qui reconnaîtrait ce langage.



Exemples : PHP, Perl, Javascript, Python ...

Notion de programme - langages

Exécution d'un programme :

Langage interprété vs langage compilé

- Dans un langage interprété, le même code source pourra marcher directement sur tout ordinateur. Avec un langage compilé, il faudra (en général) tout recompiler à chaque fois ce qui pose parfois des soucis.
- Dans un langage compilé, le programme est directement exécuté sur l'ordinateur, donc il sera en général plus rapide que le même programme dans un langage interprété.

Notion de programme - langages

Exécution d'un programme :

Attention,Tout n'est pas aussi simple :

- un langage compilé peut aussi être interprété
- la compilation est syntaxique ou/et lexicale
- il existe des compilateurs de langages interprétés

Langage python : variables et opérations

Langage python : variables et opérations

Environnement de travail :

feuille jupyter Notebook dans un premier temps.

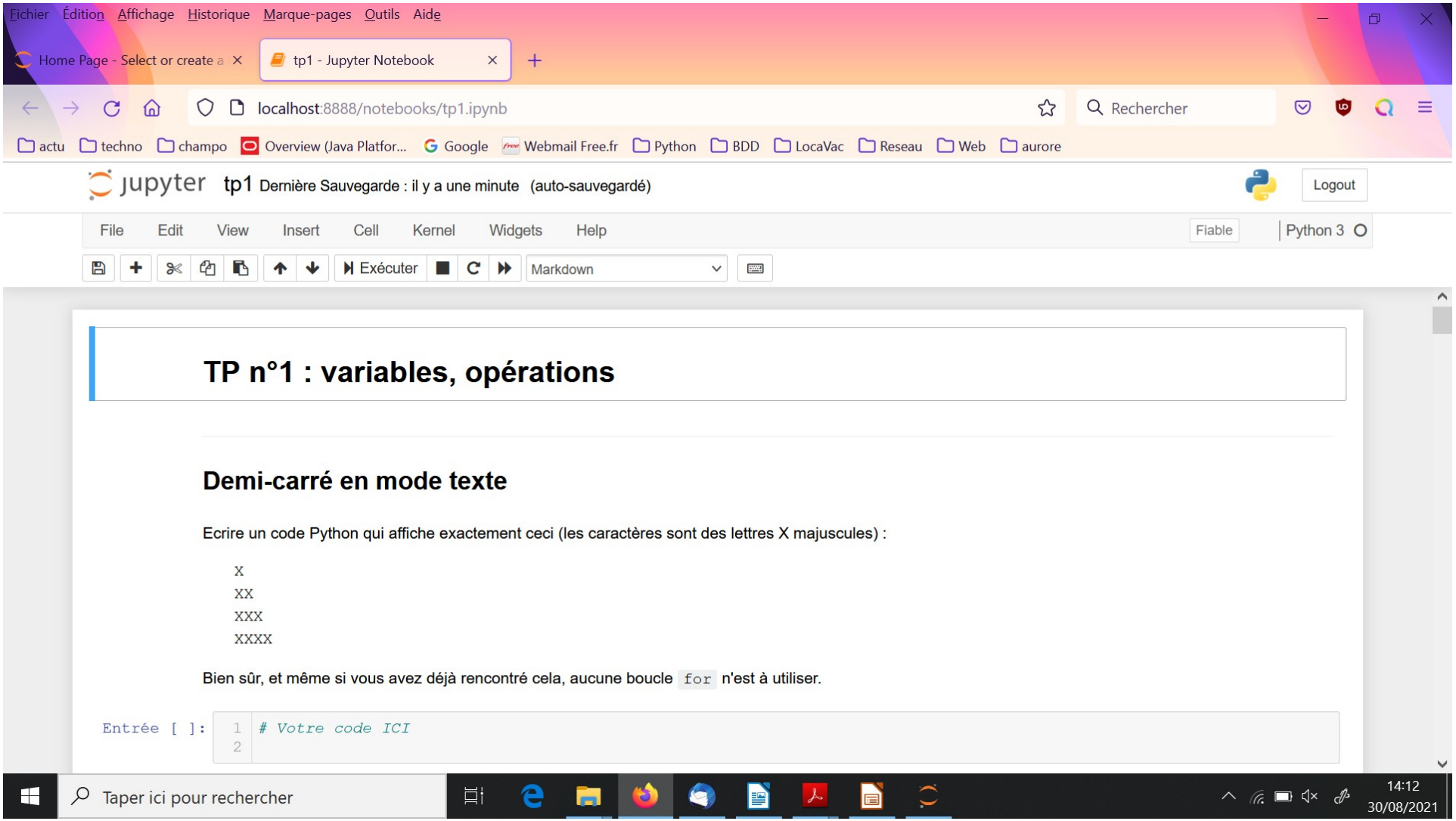
puis IDE (Integrated Development Environment)
Spyder ou autre

le mode d'emploi de Jupyter Notebook
est précisé sur Moodle …

lien pour Jupyter en ligne : https://jupyter.org/try

Langage python : variables et opérations

Jupyter Notebook



Langage python : variables et opérations

Variables

les variables permettent de stocker des valeurs et les résultats de calculs.

En python les variables sont typées au moment de leur première **affectation** (opérateur **=**) :

Par exemple :

x = 7. la valeur entière 7 est affectée à la variable x.

sur une feuille Jupyter, cette instruction ne produit aucun affichage.

Langage python : variables et opérations

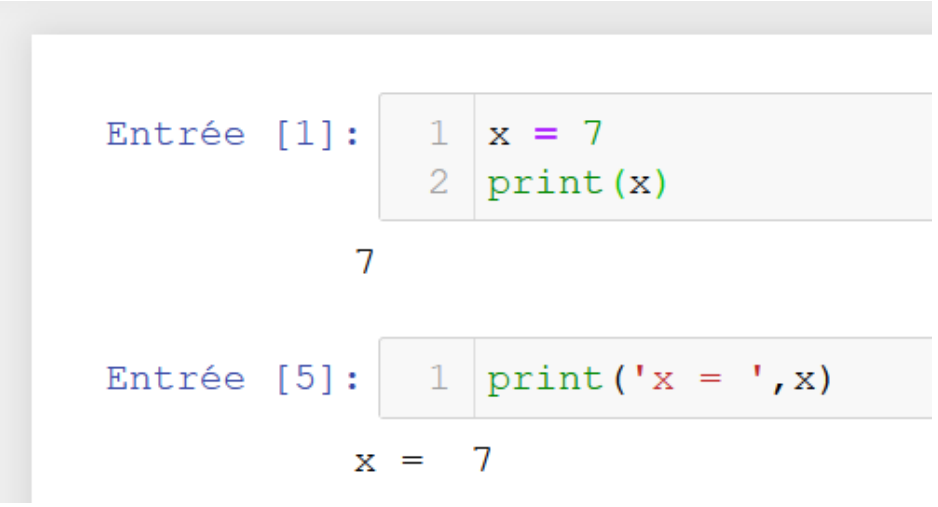
Variables et opérateurs

Pour afficher la valeur de x, il faut utiliser la fonction **print** :

print(x).

si vous voulez l'affichage : x = 7,
il faut mettre dans la commande print la chaîne de caractère
'x = ' (ou "x = ")

cela donne :



Langage python : variables et opérations

Variables et opérateurs

les types de base sont :

- les entiers (**int**) (positifs ou négatifs) : -3, 6, ...
- les réels (ou flottant) (**float**) : 3.1415926, -4.7 …
- les bouléens (**bool**) : True, False
- les chaînes de caractères (**String**) : ‘Bonjour’ ...

liste bien sûr non exhaustive ….

Langage python : variables et opérations

Variables et opérateurs

une variable peut tour à tour prendre des valeurs de type différents (typage dynamique) :

Entrée [6]:

```
1 x = 7
2 print('x = ',x)
3 x = 3.14
4 print('x = ',x)
5 x = 'Bonjour'
6 print('x = ',x)
```

```
x =  7
x =  3.14
x =  Bonjour
```

mais à éviter pour plus de clarté dans votre programme !!!

Langage python : variables et opérations

Variables et opérateurs

les opérateurs classiques : **+**, **-**, *, **/** , ** (puissance) s'appliquent sur les entiers et les réels.

Attention au parenthésage :

```
Entrée [8]:
1  x = 12/3*2
2  print('x = ',x)
3  x = 12/(3*2)
4  print('x = ',x)
```

```
x =  8.0
x =  2.0
```

Langage python : variables et opérations

Variables et opérateurs

lorsque l'on utilise à la fois des entiers et des réels dans une expression, le résultat est réel :

Entrée [12]:

```
1 x = 3.5*5
2 print('x = ',x)
3 y = 9
4 z = y**0.5
5 print('la racine carrée de ',y,' est : ',z)
```

```
x =  17.5
la racine carrée de  9  est :  3.0
```

Langage python : variables et opérations

Variables et opérateurs

la division entière : le quotient de l'entier a par b est a // b

Entrée [16]:

```
1 x = 10//3
2 print('x = ',x)
3 y = 10//9
4 print('y = ',y)
5 z = 10//5
6 print('z = ',z)
```

```
x =  3
y =  1
z =  2
```

Langage python : variables et opérations

Variables et opérateurs

le reste (fonction modulo) dans la division entière de a par b est a%b :

Entrée [18]:

```
1 x = 10%4
2 print('x = ',x)
3 y = 10%9
4 print('y = ',y)
5 z = 10%5
6 print('z = ',z)
```

```
x =  2
y =  1
z =  0
```

Langage python : variables et opérations

Variables et opérateurs

la concaténation de chaînes de caractères **+**:

Entrée [20]:

```
1 x = "Bonjour "+"tout "+"le "+"monde"
2 print('x = ',x)
3 msg1 = "Bonjour "
4 msg2 = "tout "
5 msg3 = "le "
6 msg4 = "monde"
7 msg = msg1+msg2+msg3+msg4
8 print(msg)
9
```

```
x =  Bonjour tout le monde
Bonjour tout le monde
```

Langage python : variables et opérations

Variables et opérateurs

opération « et » et « ou »
sur les bouléens : **or**, **and**

la négation : fontion **not()**

Entrée [27]:

```
1  t = True
2  print(not(t))
3
```

```
False
```

Entrée [25]:

```
1  t = True
2  f = False
3  print('Table de vérité du OU :')
4  print('')
5  print(t ,' or ',t,' : ',t or t)
6  print(t ,' or ',f,' : ',t or f)
7  print(f ,' or ',t,' : ',f or t)
8  print(f ,' or ',f,' : ',f or f)
9  print('')
10 print('Table de vérité du ET :')
11 print('')
12 print(t ,' and ',t,' : ',t and t)
13 print(t ,' and ',f,' : ',t and f)
14 print(f ,' and ',t,' : ',f and t)
15 print(f ,' and ',f,' : ',f and f)
16
```

```
Table de vérité du OU :

True  or  True  :  True
True  or  False  :  True
False  or  True  :  True
False  or  False  :  False

Table de vérité du ET :

True  and  True  :  True
True  and  False  :  False
False  and  True  :  False
False  and  False  :  False
```

Langage python : variables et opérations

Variables et opérateurs

opérateur de test d'égalité et différend de : **==** ,**!=**

Entrée [29]:

```
1 x = 1
2 y = 2
3 z = y-x
4 print('x = y ? ' , x==y)
5 print('x = z ? ', x==z)
6 print('y != z ? ', y!=z)
7
```

```
x = y ?  False
x = z ?  True
y != z ?  True
```

Langage python : variables et opérations

Variables et opérateurs

attention aux test d'égalité est de différend de avec les réels :

Entrée [38]:
```
1 print('6.0*0.7 = 4.2 ? ', 6.0*0.7 == 4.2)
2
```

```
6.0*0.7 = 4.2 ?  False
```

Entrée [42]:
```
1 print('4.2/0.7 = 6.0 ? ', 4.2/0.7 == 6.0)
```

```
4.2/0.7 = 6.0 ?  False
```

cela vient du fait que la représentation binaire des réels en base 10, n'est pas forcément stockable de façon exacte ...

Langage python : variables et opérations

Consignes pour les TPS

- vos noms de variables doivent être « parlants »
- votre code doit être commenté correctement

Entrée [44]:

```
# Exercice 1 : Calcul d'une moyenne
#
#les notes :
#
note_Math = 12.5
note_Prog = 11.0
note_Numeration = 9.5
#
#la somme des notes :
#
somme_Notes = note_Math+note_Prog+note_Numeration
#
#calcul de la moyenne :
#
moyenne = somme_Notes/3.0
#
#impression de la moyenne :
#
print('la moyenne est : ',moyenne)

```

```
la moyenne est :  11.0
```

Langage python : variables et opérations

Consignes pour les TPS

convention :

le nom d'une variable commence toujours par une minuscule.

Si le nom d'une variable est un composé de plusieurs mots, la première lettre de chaque mot est en majuscule (sauf pour le premier mot)