

# IFT211/IFT776

## Programmation scientifique en Python

### Introduction

Gabriel Girard

Département d'informatique



4 janvier 2018

# Introduction

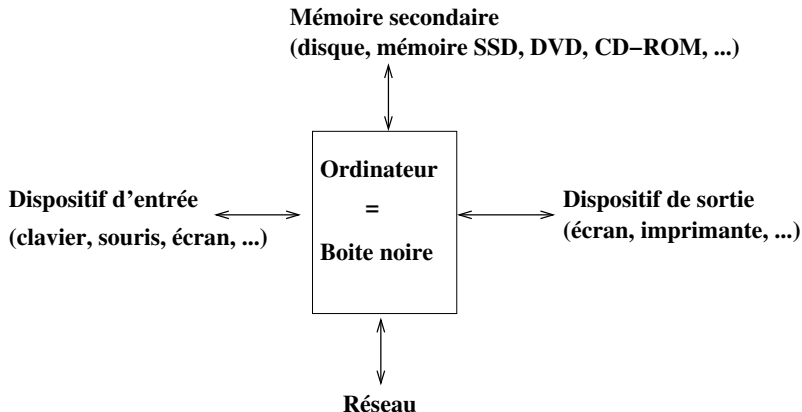
- 1 Environnement matériel
  - Mémoire centrale
  - UCT
  - Unités d'entrées et de sorties
  
- 2 Environnement logiciel

# Les ordinateurs

## Un ordinateur

- Est outil inanimé ;
- Exécute les instructions données ;
  - Programme = suite d'instructions ;
  - Programme écrit dans le langage de l'ordinateur.

# Environnement matériel

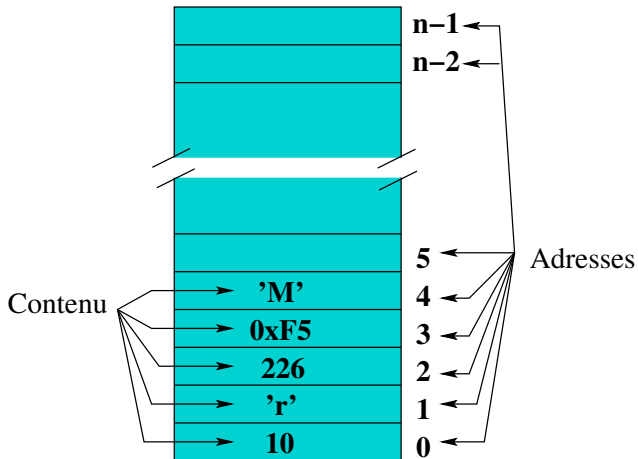


## Ordinateur et ses périphériques

# Mémoire centrale

- Séquence ordonnée de cellules (0 à N-1) ;
- N est la taille exprimée en Meg ou Gig ;
- Numéro de la cellule = adresse ;
- Cellule = 1 octet = 8 bits ;
- Valeur occupe une ou plusieurs cellules ;

# Mémoire centrale



# UCT

- Unité de contrôle ;
- Unités fonctionnelles (UAL, ...);
- Registres (généraux, PC, PS);

# Unités d'entrées et de sorties

- Clavier, souris, crayon, guichet, ...
- Écran, imprimante, ...



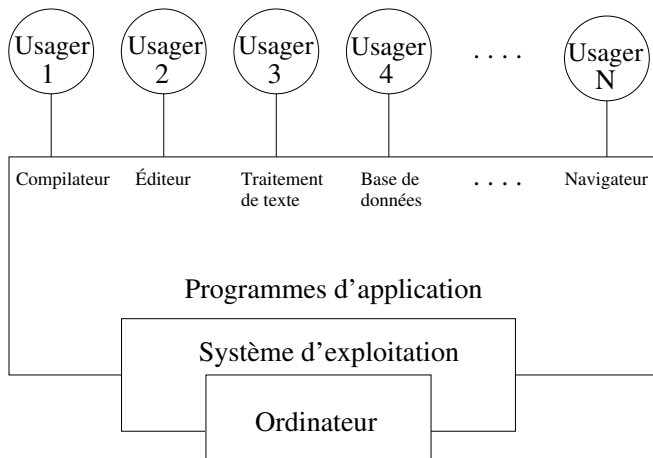
# Unités d'entrées et de sorties

- Disque souple ;
- Disque rigide ;
- Carte de mémoire (flash, ...);
- Bande magnétique ;
- Disque compact (CD, DVD, Blu-ray) ;
- Concept de fichiers

# Environnement Logiciel

- Système d'exploitation ;
- Applications ;
- Outils de développement  
(langage, compilateur, interpréteur,...) ;

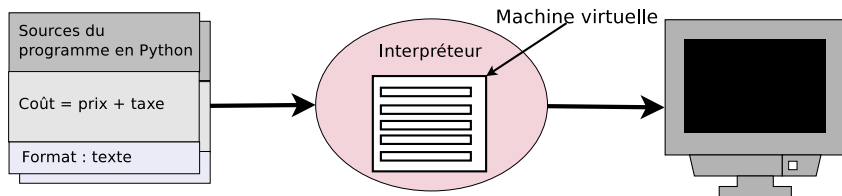
# Environnement Logiciel



# Environnement de développement

- Langage de programmation ;
- Compilateur/interpréteur ;
- Éditeurs, outil de mise au point, éditeur de liens, ... ;
- Environnement de développement intégré ;

# Environnement de développement



# Étude de cas

## Consommation automobile

Entre deux pleins d'essence un automobiliste désire connaître sa consommation d'essence.

# Étude de cas : Consommation automobile

```
1 #!/usr/bin/env python
2 # -*- coding: utf-8 -*-
3 """
4 ‘‘Consommation essence’’ **module description**
5
6 Ce script calcule et affiche la consommation d'essence.
7
8 Entrees :
9     (clavier) distance parcourue : reel
10    (clavier) volume utilise : reel
11
12 Sorties :
13    (ecran) consommation : reel
14
15 ...Auteur: Gabriel Girard
16     Janvier 2016
17
18 """
```

# Étude de cas : Consommation automobile

```
1  ## Declaration des variables
2  LITRE_A_GALLON = 0.22      ##Constante de conversion
3  KM_A_MILLE = 0.6214       ##Constante de conversion
4
5  ## Lecture de la distance parcourue
6  distance_kms = eval(input("Distance parcourue : "))
7  ## Lecture de la distance parcourue
8  plein_essence_li = eval(input("Quantite d'essence :"))
9
10 ## Conversion
11 distance_milles = distance_kms * KM_A_MILLE
12 plein_essence_gal = plein_essence_li * LITRE_A_GALLON
13 ## calcul de la consommation
14 consommation = distance_milles / plein_essence_gallons
15
16 ##Affichage du resultat
17 print("Consommation = ", consommation, " milles/galllon\n")
```





# Étude de cas : Consommation automobile

```
1 def main():
2     ## Declaration des variables
3     LITRE_A_GALLON = 0.22     ## Constante de conversion
4     KM_A_MILLE = 0.6214     ## Constante de conversion
5
6     ## Lecture de la distance parcourue
7     distance_kms = eval(input("Distance parcourue : "))
8     ## Lecture de la distance parcourue
9     plein_essence_li = eval(input("Quantite d'essence : "))
10    ## Conversion
11    distance_milles = distance_kms * KM_A_MILLE
12    plein_essence_gal = plein_essence_li * LITRE_A_GALLON
13    ## calcul de la consommation
14    consommation = distance_milles / plein_essence_gallons
15    ##Affichage du resultat
16    print("Consommation = ", consommation, " mi/gal")
17
18 if __name__ == '__main__':
19     main()
```



## Loi de Murphy (suite)

- Laissées à elles-mêmes, les choses tendent à aller de mal en pis.
- Même si quelque chose ne peut pas aller mal, cela ira mal quand même.
- Si tout semble fonctionner correctement, alors vous avez manifestement oublié quelque chose.
- La nature nous réserve toujours des pannes.

# Conclusion

- Edger W. Dijkstra  
“If debugging is the process of removing bugs, then programming must be the process of putting them in.”
- Alan J. Perlis  
“There are two ways to write error-free programs ; only the third one works.”

# Unités de mesure

Multiples de l'octet : préfixes SI et mésusages

Nom	Symbole	Valeur	Mésusage
kiloctet	ko	$10^3$	$2^{10}$
mégaoctet	Mo	$10^6$	$2^{20}$
gigaoctet	Go	$10^9$	$2^{30}$
téraoctet	To	$10^{12}$	$2^{40}$
pétaoctet	Po	$10^{15}$	
exaoctet	Eo	$10^{18}$	
zettaoctet	Zo	$10^{21}$	
yottaoctet	Yo	$10^{24}$	



# Unités de mesure

Multiples de l'octet : préfixes binaires

(Normes établies par la Commission électrotechnique internationale)

Nom	Symbole	Valeur
kibioctet	Kio	$2^{10}$
mébioctet	Mio	$2^{20}$
gibioctet	Gio	$2^{30}$
tébioctet	Tio	$2^{40}$
pébioctet	Pio	$2^{50}$
exbioctet	Eio	$2^{60}$
zébioctet	Zio	$2^{70}$
yobioctet	Yio	$2^{80}$

