

Le langage Python

Introduction



Pr Ilham Oumaira

Introduction



- On appelle « **langage informatique** » un langage destiné à décrire l'ensemble des actions consécutives qu'un ordinateur doit exécuter.
- À chaque instruction correspond une action du processeur
- Ecrire un programme informatique revient à écrire un algorithme dans un langage compréhensible par un ordinateur

Programme = Algorithme + Langage

- Un programme informatique est la traduction de l'algorithme dans le langage cible.
- Contrairement à un algorithme qui s'écrit dans un langage de pseudo-code ou en langage naturel, un programme doit être correct syntaxiquement.
- Un même algorithme peut être écrit dans des dizaines de langages de programmations différents: C, java, javascript, C++, caml, python,

Types de langages





1- langage machine

- C'est une suite de bits qui est interprétée par le processeur d'un ordinateur exécutant un programme informatique. C'est le **langage natif** d'un processeur, c'est-à-dire le seul qu'il puisse traiter. Il est composé d'instructions et de données à traiter codées en binaire.
- Chaque processeur possède son propre langage machine, dont un code machine qui ne peut s'exécuter que sur la machine pour laquelle il a été préparé. Si un processeur *A* est capable d'exécuter toutes les instructions du processeur *B*, on dit que *A* est compatible avec *B*.



2- Langage assembleur

- Langage proche du langage machine qui peut être directement interprété par le processeur de l'ordinateur tout en restant lisible par un humain, crée pour faciliter le travail des programmeurs. Il consiste à représenter les combinaisons de bits employées en langage binaire par des symboles.

- Exemple : `ADD r0,r1,0x62`

signifie "ajouter la valeur du registre r1 avec la valeur hexadécimale 0x62 et mettre le résultat dans le registre r0"



3- Langage haut niveau

- Les langages de haut niveau sont plus proches des langues naturelles, ce qui facilite et vulgarise l'écriture des programmes.
- Ils sont généralement indépendants de la machine : le même programme pourra être utilisé tel quel sur plusieurs types d'ordinateurs — quoique les programmes puissent également être conçus pour un système d'exploitation en particulier.

Référence: http://fr.wikipedia.org/wiki/Langage_de_haut_niveau



Traduction en langage binaire

Les langages informatiques peuvent se classer en deux catégories : les **langages interprétés** et les **langages compilés**.

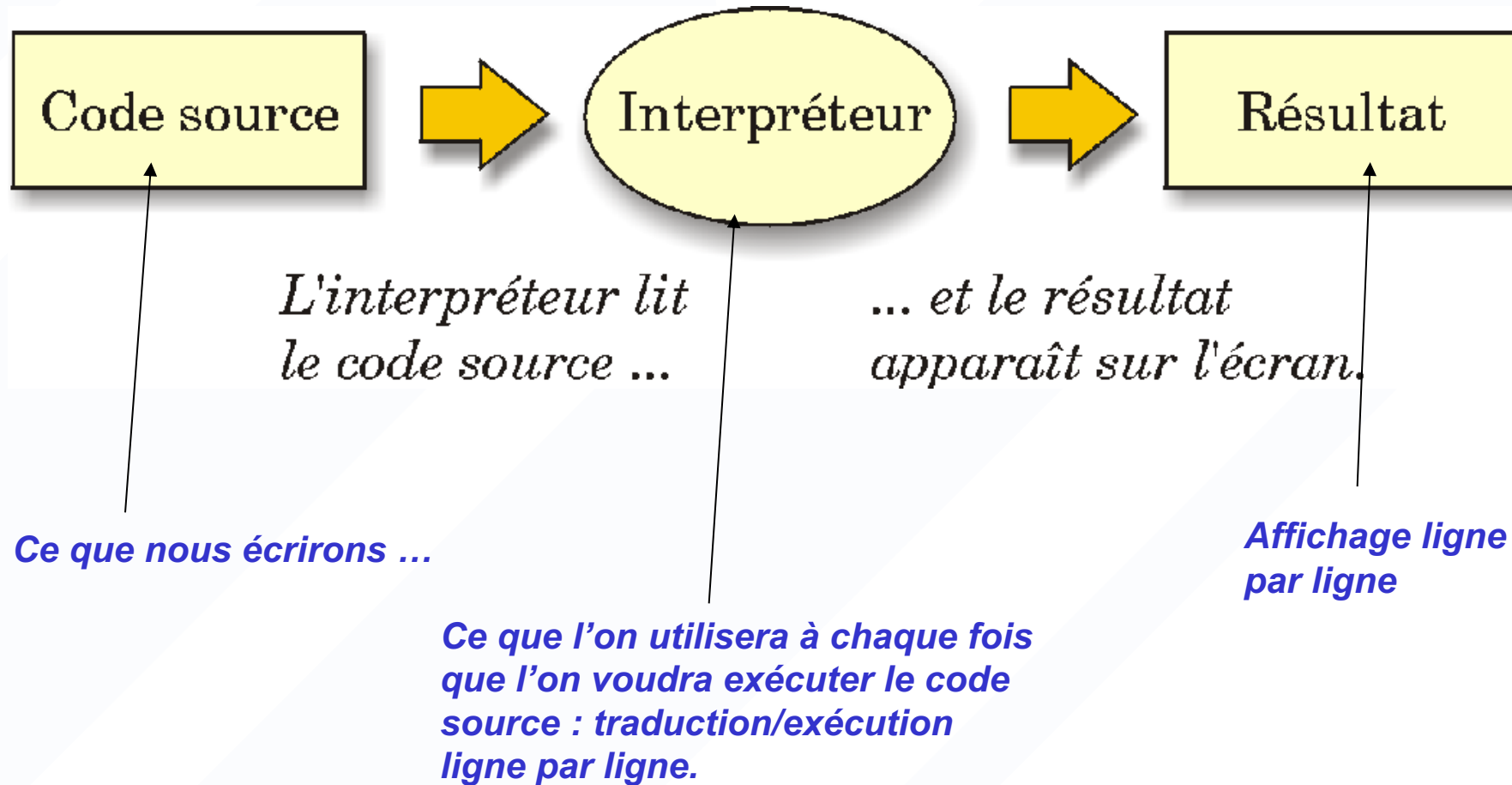
- **Langage compilé**

Un programme écrit dans un langage dit « **compilé** » va être traduit une fois pour toutes par un programme annexe, appelé **compilateur**, afin de générer un nouveau fichier qui sera autonome, c'est-à-dire qui n'aura plus besoin d'un programme autre que lui pour s'exécuter; on dit que ce fichier est **exécutable**.

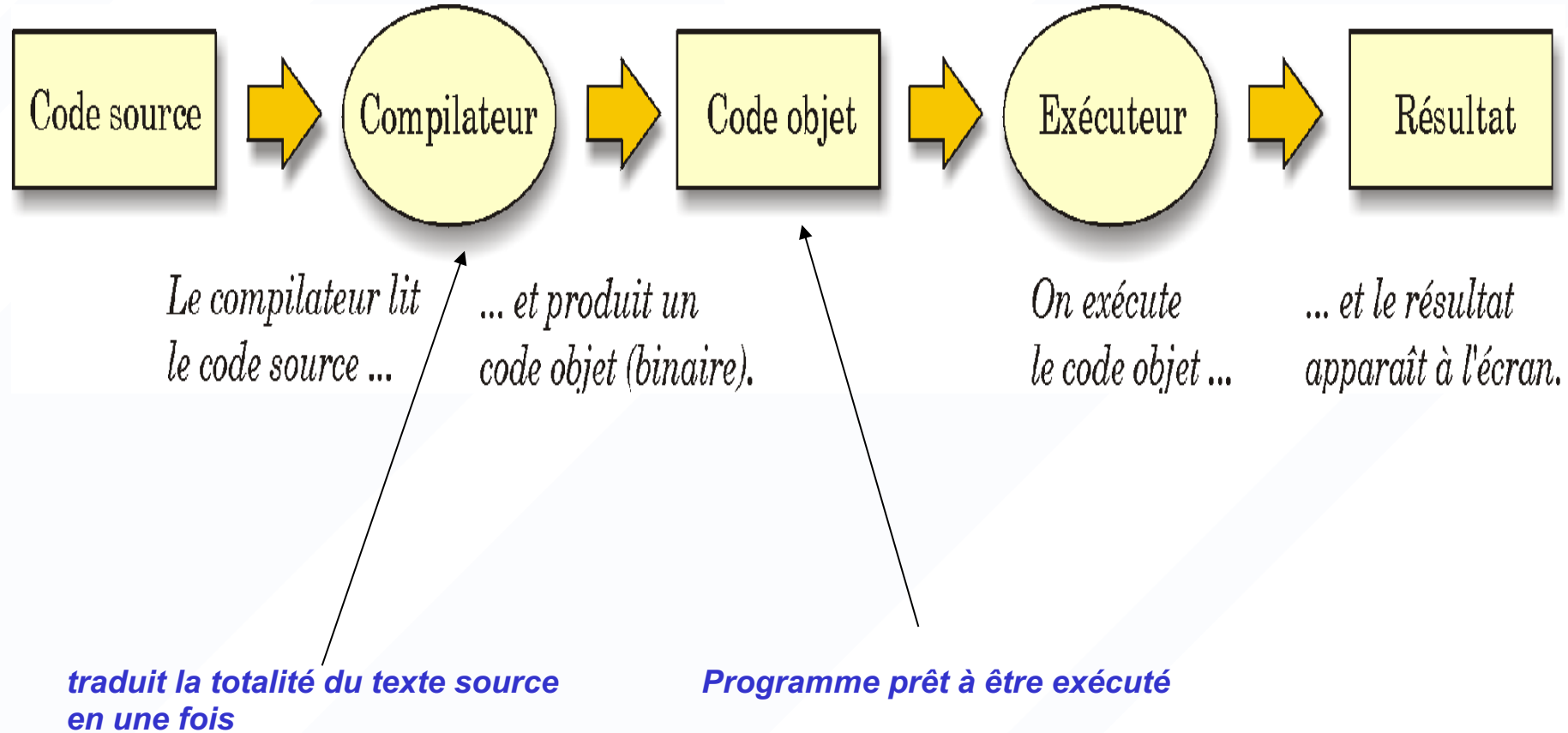
- **Langage interprété**

Un programme écrit dans un langage interprété a besoin d'un programme auxiliaire (l'interpréteur) pour traduire au fur et à mesure les instructions du programme.

Interprétation



Compilation



Comparaison Compilation et interprétation



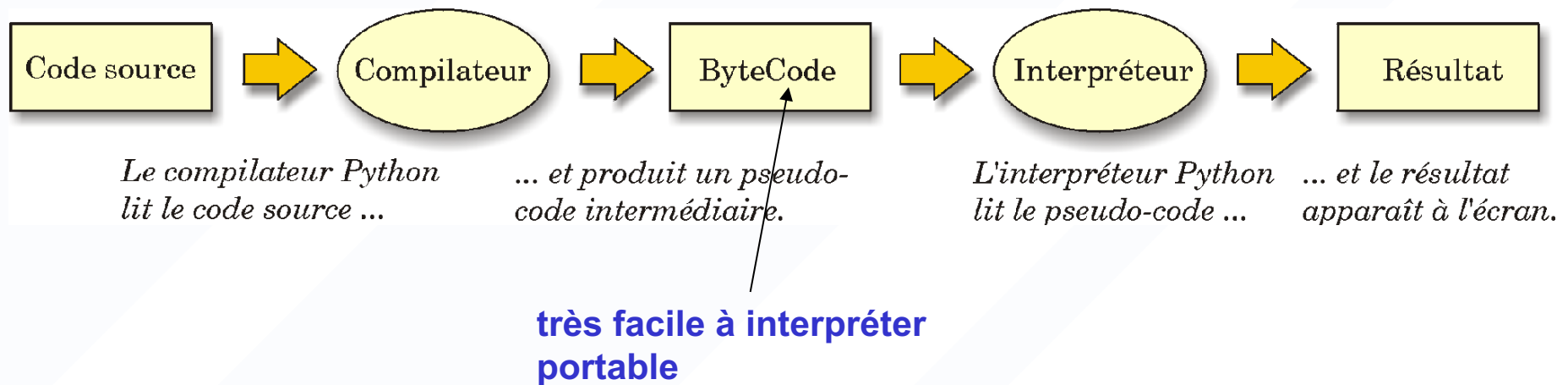
chacune de ces deux techniques a ses avantages et ses inconvénients

- L'interprétation est idéale lorsque l'on est en phase d'apprentissage du langage. Avec cette technique, on peut tester immédiatement toute modification apportée au programme source, sans passer par une phase de compilation qui demande toujours plus de temps.
- Par contre, lorsqu'un projet comporte des fonctionnalités complexes qui doivent s'exécuter rapidement, la compilation est préférable : il est clair qu'un programme compilé fonctionnera toujours nettement plus vite que son homologue interprété, puisque dans cette technique l'ordinateur n'a plus à (re)traduire chaque instruction en code binaire avant qu'elle puisse être exécutée.
- un programme compilé a pour avantage de garantir la sécurité du code source. En effet, un langage interprété, étant directement intelligible (lisible), permet à n'importe qui de connaître les secrets de fabrication d'un programme et donc de copier le code voire de le modifier

langages intermédiaires



Certains langages **combinent**
les deux techniques afin de garder le
meilleur de chacune. On parle de
langages intermédiaires



Mise au point d'un programme. Recherche des erreurs «debug»



3 types d'erreurs :

1. **syntaxe** : se réfère aux règles que les auteurs du langage ont établies pour la structure du programme.
2. **sémantique** : c'est une erreur de logique, i.e, le programme est sans erreurs mais les résultats sont inattendus.
3. **d'exécution ou « Run-time error »** : lorsque votre programme fonctionne déjà, mais que des circonstances particulières se présentent (par exemple, votre programme essaie de lire un fichier qui n'existe plus, une division par zéro).



Quelques exemples de langages couramment utilisés

Langage	Domaine d'application principal	Compilé/interprété
ADA	Le temps réel	Langage compilé
BASIC	Programmation basique à but éducatif	Langage interprété
C	Programmation système	Langage compilé
C++	Programmation système objet	Langage compilé
Cobol	Gestion	Langage compilé
Fortran	Calcul	Langage compilé
Java	Programmation orientée internet	Langage intermédiaire
MATLAB	Calcul mathématique	Langage interprété
Mathematica	Calcul mathématique	Langage interprété
LISP	Intelligence artificielle	Langage intermédiaire
Pascal	Enseignement	Langage compilé
PHP	Développement de sites web dynamiques	Langage interprété
Prolog	Intelligence artificielle	Langage interprété
Perl	Traitement de chaînes de caractères	Langage interprété

Source: http://www.commentcamarche.net/contents/617-les-langages-informatiques#simili_main

Le langage Python



2- le langage Python



- Python est un langage **multiplateforme**, c'est-à-dire disponible sur plusieurs architectures (compatible PC, tablettes, smartphones, ordinateur low cost Raspberry Pi...) et systèmes d'exploitation (Windows, Linux, Mac, Android...).
- Il est doté d'un typage dynamique fort
- Le langage Python est gratuit, sous **licence libre**.
- C'est un des langages informatiques les plus populaires avec C, C++, C#, Objective-C, Java, PHP, JavaScript, Delphi, Visual Basic, Ruby et Perl
- Actuellement, Python en est à sa version 3. Cependant, la version 2 est encore largement utilisée.

Attention : Python 2 n'est pas compatible avec Python 3 !



2-1 Historique du langage

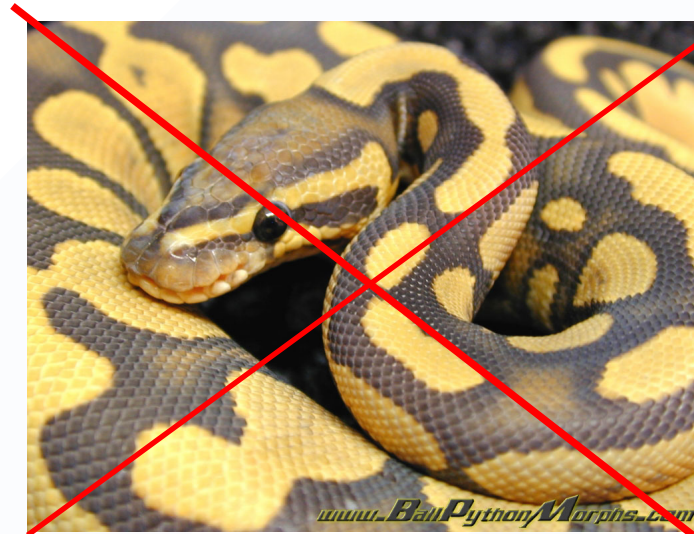
- En 1989, Guido Van Rossum (née le 31 janvier 1956 au [Pays-Bas](#)) commença à travailler sur Python qui n'était alors qu'un projet lui servant d'occupation durant les vacances de Noël pendant lesquelles son bureau était fermé.
- Le but de Guido était d'inventer un successeur au langage ABC, un langage d'apprentissage peu apprécié dans le milieu académique.
- Pour cela, il fit appel directement à des utilisateurs Unix habitués au langage C. il a voulu que Python soit facilement utilisable dans d'autres langages et environnement contrairement à ABC. Il y réussit globalement...



2-1 Historique du langage



- Fan de la série télévisée des [Monty Python](#), il décide de baptiser ce projet Python



- Python est Maintenu par une communauté de bénévoles, sous couvert de la Python Software Foundation



2-2 Pourquoi le Python ?

- Simple mais puissant (évolutif)
- Efficace, et fourni avec de nombreuses bibliothèques (Battery included !)
- Code très lisible, pédagogique
- De plus en plus utilisé (Google, FreeCAD, ...)
- Logiciel libre, disponible sur toutes les plateformes (Windows, Linux, Mac OS, etc.)

Classement des langages de programmation les plus populaire (2015- 2016)



<http://www.tiobe.com/index.php/content/paperinfo/tpci/>

Mar 2016	Mar 2015	Change	Programming Language
1	2	^	Java
2	1	v	C
3	4	^	C++
4	5	^	C#
5	8	^	Python
6	6		PHP
7	9	^	Visual Basic .NET
8	7	v	JavaScript
9	12	^	Perl
10	18	^^	Ruby
11	13	^	Delphi/Object Pascal
12	28	^^	Assembly language
13	10	v	Visual Basic
14	23	^^	Swift
15	3	v	Objective-C
16	20	^^	R
17	36	^^	Groovy
18	19	^	MATLAB
19	17	v	PL/SQL
20	31	^^	D

Classement des langages de programmation (2016- 2017)



Apr 2017	Apr 2016	Change	Programming Language
1	1		Java
2	2		C
3	3		C++
4	4		C#
5	5		Python
6	6		PHP
7	10	▲	Visual Basic .NET
8	7	▼	JavaScript
9	11	▲	Delphi/Object Pascal
10	8	▼	Perl
11	9	▼	Ruby
12	15	▲	Swift
13	12	▼	Assembly language
14	13	▼	Objective-C
15	18	▲	R
16	14	▼	Visual Basic

Classement des langages de programmation (2019- 2020)



Mar 2020	Mar 2019	Change	Programming Language
1	1		Java
2	2		C
3	3		Python
4	4		C++
5	6	^	C#
6	5	v	Visual Basic .NET
7	7		JavaScript
8	8		PHP
9	9		SQL
10	18	^^	Go
11	14	^	R
12	12		Assembly language
13	17	^^	Swift
14	15	^	Ruby
15	11	v	MATLAB

2-3 Que peut-on faire avec Python ?



- du calcul scientifique (bibliothèque [NumPy](#))
- des graphiques (bibliothèque [matplotlib](#))
- du traitement du son, de la synthèse vocale (bibliothèque eSpeak)
- du traitement d'image (bibliothèque [PIL](#)), de la vision artificielle par caméra (framework [SimpleCV](#))
- de la bio-informatique (bibliothèque [Biopython](#))
- des applications avec interface graphique GUI (bibliothèques [Tkinter](#), [PyQt](#), [wxPython](#), [PyGTK](#)...)
- des jeux vidéo en 2D (bibliothèque [Pygame](#))

2-3 Que peut-on faire avec Python ?



- des applications Web (serveur Web [Zope](#) ; frameworks Web [Flask](#), [Django](#) ; framework JavaScript [Pyjamas](#))
- interfacier des systèmes de gestion de base de données (bibliothèque [MySQLdb](#)...)
- des applications réseau (framework [Twisted](#))

Des **dizaines de milliers** de bibliothèques sont disponibles sur le dépôt officiel [PyPI](#).

De belles références



- La NASA,
- Google,
- YouTube
- edx
-



De quoi a-t-on besoin pour programmer en python

Programmer en python nécessite :

- un éditeur de texte (ex : gedit, wordpad, emacs, ...)
- un interpréteur python (un programme capable de comprendre les programmes écrits en python).



- IDLE est un environnement de développement intégré (IDE en anglais : Integrated Development Environment) pour Python.

IDLE propose un certain nombre d'outils :

- un éditeur de texte (pour écrire le programme)
 - un interpréteur (pour exécuter le programme)
 - un débogueur (pour tester le programme)
- Il existe d'autres IDE pour Python : Eclipse/Pydev, NetBeans, Eric...

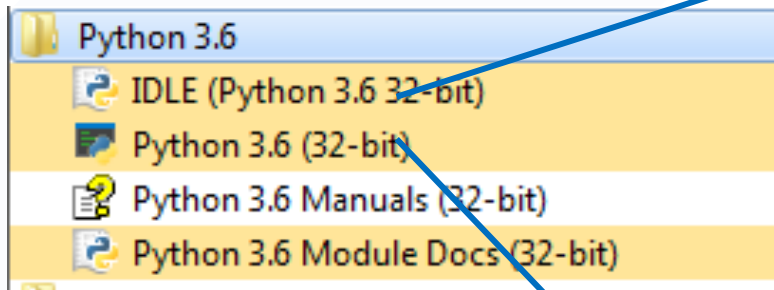
Installation



- Télécharger la dernière version (3.6.1) du site officiel

<https://www.python.org/downloads/>

Après installation:



Mode interactif

```
Python 3.6.1 (v3.6.1:69c0db5, Mar 21 2017, 17:54:52)
[MSC v.1900 32 bit (Intel)] on win32
Type "copyright", "credits" or "license()" for more i
nformation.
>>> a=2
>>> b=7
>>> c=a+b
>>> c
9
>>> |
```

```
Python 3.6.1 (v3.6.1:69c0db5, Mar 21 2017, 17:54:52) [MS
on win32
Type "help", "copyright", "credits" or "license" for mor
>>> a=2
>>> b=7
>>> c=a+b
>>> c
9
>>>
```

Modes d'exécution en python



- Il existe deux modes d'exécution de python :
 - **Mode interactif** : ouvrez un terminal, et évaluer des expressions en interagissant avec python.

Interpréteur = Calculatrice améliorée

- **Mode batch** : ouvrez un terminal et demander à python d'exécuter les instructions d'un programme contenu dans un fichier (.py) On parlera aussi de scripts python.

Mode interactif : Exemple



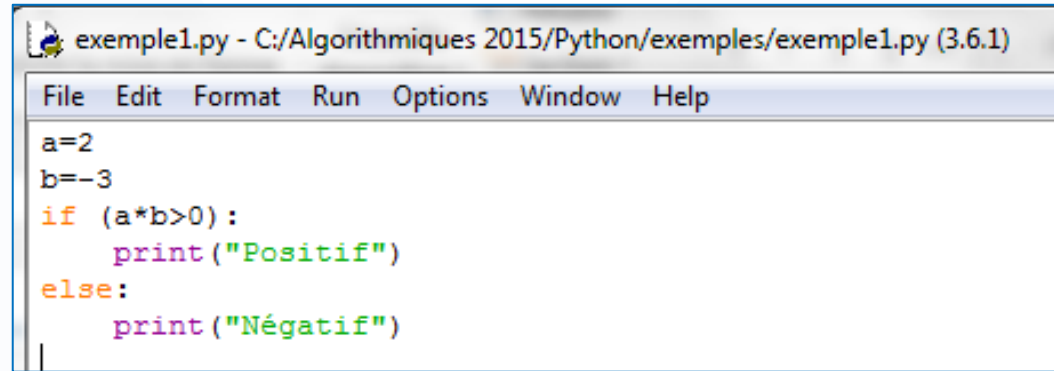
```
Python 3.6.1 Shell
File Edit Shell Debug Options Window Help
Python 3.6.1 (v3.6.1:69c0db5, Mar 21 2017, 17:54:52) [MSC v.1900
on win32
Type "copyright", "credits" or "license()" for more information.
>>> a=2
>>> b=-3
>>> if (a*b>0):
        print("positif")
else:
        print("négatif")

négatif
>>> |
```

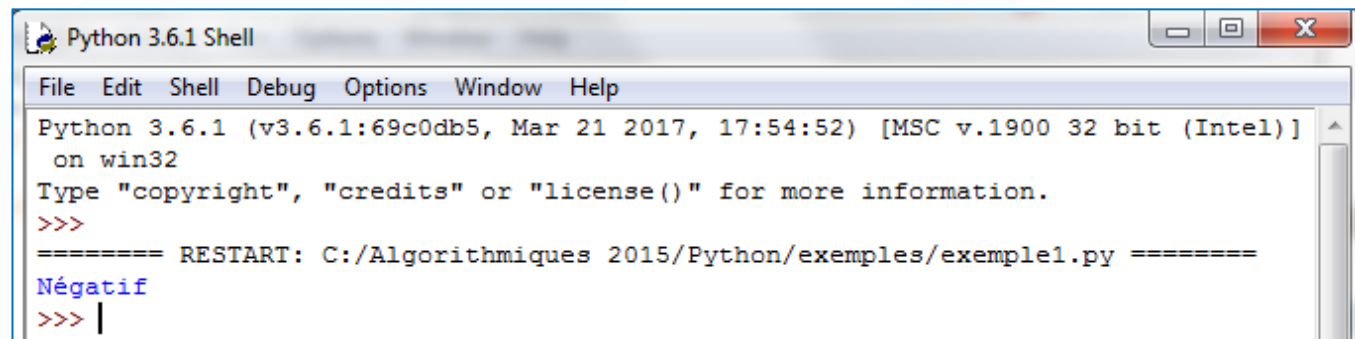


Mode Batch ou script

- Lancer IDLE
- Cliquer sur File -> New File
- Enregistrer le fichier
- Tapez le code source
- Pour interpréter cliquez sur: Run ->Run Module (ou F5)



```
exemple1.py - C:/Algorithmiques 2015/Python/exemples/exemple1.py (3.6.1)
File Edit Format Run Options Window Help
a=2
b=-3
if (a*b>0):
    print("Positif")
else:
    print("Négatif")
```



```
Python 3.6.1 Shell
File Edit Shell Debug Options Window Help
Python 3.6.1 (v3.6.1:69c0db5, Mar 21 2017, 17:54:52) [MSC v.1900 32 bit (Intel)]
on win32
Type "copyright", "credits" or "license()" for more information.
>>>
===== RESTART: C:/Algorithmiques 2015/Python/exemples/exemple1.py =====
Négatif
>>> |
```