

Introduction à la science informatique

MPSI-Lycée La Merci



Ce dont nous allons parler...

- 1 Quelques généralités sur la science informatique

Ce dont nous allons parler...

- 1 Quelques généralités sur la science informatique
- 2 Présentation du langage python

Ce dont nous allons parler...

- 1 Quelques généralités sur la science informatique
- 2 Présentation du langage python
- 3 Quelle version utiliser ?

Ce dont nous allons parler...

- 1 Quelques généralités sur la science informatique
- 2 Présentation du langage python
- 3 Quelle version utiliser ?
- 4 Premiers pas avec l'interpréteur Python

Ce dont nous allons parler...

- 1 Quelques généralités sur la science informatique
- 2 Présentation du langage python
- 3 Quelle version utiliser ?
- 4 Premiers pas avec l'interpréteur Python
 - opérations et comparaisons sur les nombres

Ce dont nous allons parler...

- 1 Quelques généralités sur la science informatique
- 2 Présentation du langage python
- 3 Quelle version utiliser ?
- 4 Premiers pas avec l'interpréteur Python
 - opérations et comparaisons sur les nombres
 - l'affectation et la notion de variable
 - bibliothèques

Ce dont nous allons parler...

- 1 Quelques généralités sur la science informatique
- 2 Présentation du langage python
- 3 Quelle version utiliser ?
- 4 Premiers pas avec l'interpréteur Python
 - opérations et comparaisons sur les nombres
 - l'affectation et la notion de variable
 - bibliothèques
 - un peu de dessin avec turtle

Ce dont nous allons parler...

- 1 Quelques généralités sur la science informatique
- 2 Présentation du langage python
- 3 Quelle version utiliser ?
- 4 Premiers pas avec l'interpréteur Python
 - opérations et comparaisons sur les nombres
 - l'affectation et la notion de variable
 - bibliothèques
 - un peu de dessin avec turtle
 - la notion de fonction

Ce dont nous allons parler...

- 1 Quelques généralités sur la science informatique
- 2 Présentation du langage python
- 3 Quelle version utiliser ?
- 4 Premiers pas avec l'interpréteur Python
 - opérations et comparaisons sur les nombres
 - l'affectation et la notion de variable
 - bibliothèques
 - un peu de dessin avec turtle
 - la notion de fonction
- 5 Les tests

Ce dont nous allons parler...

- 1 Quelques généralités sur la science informatique
- 2 Présentation du langage python
- 3 Quelle version utiliser ?
- 4 Premiers pas avec l'interpréteur Python
 - opérations et comparaisons sur les nombres
 - l'affectation et la notion de variable
 - bibliothèques
 - un peu de dessin avec turtle
 - la notion de fonction
- 5 Les tests
- 6 Les itérations

Ce dont nous allons parler...

- 1 Quelques généralités sur la science informatique
- 2 Présentation du langage python
- 3 Quelle version utiliser ?
- 4 Premiers pas avec l'interpréteur Python
 - opérations et comparaisons sur les nombres
 - l'affectation et la notion de variable
 - bibliothèques
 - un peu de dessin avec turtle
 - la notion de fonction
- 5 Les tests
- 6 Les itérations

Les quatre piliers de l'informatique

Les quatre piliers de l'informatique

- la machine

Les quatre piliers de l'informatique

- la machine
- l'information (les données)

Les quatre piliers de l'informatique

- la machine
- l'information (les données)
- l'algorithmique

Les quatre piliers de l'informatique

- la machine
- l'information (les données)
- l'algorithmique
- le langage et la programmation

Les quatre piliers de l'informatique

- la machine
- l'information (les données)
- l'algorithmique
- le langage et la programmation

Quelques questions soulevées

- Qu'est-ce qu'un ordinateur ? Un réseau ? Un robot ?

Quelques questions soulevées

- Qu'est-ce qu'un ordinateur ? Un réseau ? Un robot ?
- Comment représente-t-on un nombre, un texte, une image ou un son dans un ordinateur ?

Quelques questions soulevées

- Qu'est-ce qu'un ordinateur ? Un réseau ? Un robot ?
- Comment représente-t-on un nombre, un texte, une image ou un son dans un ordinateur ?
- Comment gère-t-on de grandes quantités de données ?

Quelques questions soulevées

- Qu'est-ce qu'un ordinateur ? Un réseau ? Un robot ?
- Comment représente-t-on un nombre, un texte, une image ou un son dans un ordinateur ?
- Comment gère-t-on de grandes quantités de données ?
- Comment trier une liste de manière efficace ? Comment mesurer cette efficacité ?

Quelques questions soulevées

- Qu'est-ce qu'un ordinateur ? Un réseau ? Un robot ?
- Comment représente-t-on un nombre, un texte, une image ou un son dans un ordinateur ?
- Comment gère-t-on de grandes quantités de données ?
- Comment trier une liste de manière efficace ? Comment mesurer cette efficacité ?
- Quel temps pour décomposer un entier en facteurs premiers avec un ordinateur ?

Quelques questions soulevées

- Qu'est-ce qu'un ordinateur ? Un réseau ? Un robot ?
- Comment représente-t-on un nombre, un texte, une image ou un son dans un ordinateur ?
- Comment gère-t-on de grandes quantités de données ?
- Comment trier une liste de manière efficace ? Comment mesurer cette efficacité ?
- Quel temps pour décomposer un entier en facteurs premiers avec un ordinateur ?
- Comment dialoguer avec un ordinateur ? Quelles sont les instructions fondamentales des langages ?

- Premières automatisations :
 - machine à calculer, la pascaline 1645
 - machine à tisser, Jacquard 1801, avec des cartes perforées
- Premiers ordinateurs : ENIAC 1943
- Premiers ordinateurs connectés en réseau (arpanet) : 1970
- Début de la démocratisation de l'Internet avec le World Wide Web et le html : 1990

- Le principe

En base 10, on a $523 = 5 \times 10^2 + 2 \times 10^1 + 3 \times 10^0$.

- Le principe

En base 10, on a $523 = 5 \times 10^2 + 2 \times 10^1 + 3 \times 10^0$.

En base 7, on a $(523)_7 = 5 \times 7^2 + 2 \times 7^1 + 3 \times 7^0 = (262)_{10}$.

- Le principe

En base 10, on a $523 = 5 \times 10^2 + 2 \times 10^1 + 3 \times 10^0$.

En base 7, on a $(523)_7 = 5 \times 7^2 + 2 \times 7^1 + 3 \times 7^0 = (262)_{10}$.

- Écriture binaire : deux chiffres 0 et 1

On a $(1011)_2 = 1 \times 2^3 + 0 \times 2^2 + 1 \times 2^1 + 1 \times 2^0 = 11$.

- Le principe

En base 10, on a $523 = 5 \times 10^2 + 2 \times 10^1 + 3 \times 10^0$.

En base 7, on a $(523)_7 = 5 \times 7^2 + 2 \times 7^1 + 3 \times 7^0 = (262)_{10}$.

- Écriture binaire : deux chiffres 0 et 1

On a $(1011)_2 = 1 \times 2^3 + 0 \times 2^2 + 1 \times 2^1 + 1 \times 2^0 = 11$.

Le prédécesseur de $(1011)_2$ est

- Le principe

En base 10, on a $523 = 5 \times 10^2 + 2 \times 10^1 + 3 \times 10^0$.

En base 7, on a $(523)_7 = 5 \times 7^2 + 2 \times 7^1 + 3 \times 7^0 = (262)_{10}$.

- Écriture binaire : deux chiffres 0 et 1

On a $(1011)_2 = 1 \times 2^3 + 0 \times 2^2 + 1 \times 2^1 + 1 \times 2^0 = 11$.

Le prédécesseur de $(1011)_2$ est $(1010)_2$.

- Le principe

En base 10, on a $523 = 5 \times 10^2 + 2 \times 10^1 + 3 \times 10^0$.

En base 7, on a $(523)_7 = 5 \times 7^2 + 2 \times 7^1 + 3 \times 7^0 = (262)_{10}$.

- Écriture binaire : deux chiffres 0 et 1

On a $(1011)_2 = 1 \times 2^3 + 0 \times 2^2 + 1 \times 2^1 + 1 \times 2^0 = 11$.

Le prédécesseur de $(1011)_2$ est $(1010)_2$.

Le successeur de $(1011)_2$ est

- Le principe

En base 10, on a $523 = 5 \times 10^2 + 2 \times 10^1 + 3 \times 10^0$.

En base 7, on a $(523)_7 = 5 \times 7^2 + 2 \times 7^1 + 3 \times 7^0 = (262)_{10}$.

- Écriture binaire : deux chiffres 0 et 1

On a $(1011)_2 = 1 \times 2^3 + 0 \times 2^2 + 1 \times 2^1 + 1 \times 2^0 = 11$.

Le prédécesseur de $(1011)_2$ est $(1010)_2$.

Le successeur de $(1011)_2$ est $(1100)_2$.

Le codage des entiers naturels

- Le principe

En base 10, on a $523 = 5 \times 10^2 + 2 \times 10^1 + 3 \times 10^0$.

En base 7, on a $(523)_7 = 5 \times 7^2 + 2 \times 7^1 + 3 \times 7^0 = (262)_{10}$.

- Écriture binaire : deux chiffres 0 et 1

On a $(1011)_2 = 1 \times 2^3 + 0 \times 2^2 + 1 \times 2^1 + 1 \times 2^0 = 11$.

Le prédécesseur de $(1011)_2$ est $(1010)_2$.

Le successeur de $(1011)_2$ est $(1100)_2$.

Le produit $2^2 \times (1011)_2$ vaut

- Le principe

En base 10, on a $523 = 5 \times 10^2 + 2 \times 10^1 + 3 \times 10^0$.

En base 7, on a $(523)_7 = 5 \times 7^2 + 2 \times 7^1 + 3 \times 7^0 = (262)_{10}$.

- Écriture binaire : deux chiffres 0 et 1

On a $(1011)_2 = 1 \times 2^3 + 0 \times 2^2 + 1 \times 2^1 + 1 \times 2^0 = 11$.

Le prédécesseur de $(1011)_2$ est $(1010)_2$.

Le successeur de $(1011)_2$ est $(1100)_2$.

Le produit $2^2 \times (1011)_2$ vaut $(101100)_2$.

- Le principe

En base 10, on a $523 = 5 \times 10^2 + 2 \times 10^1 + 3 \times 10^0$.

En base 7, on a $(523)_7 = 5 \times 7^2 + 2 \times 7^1 + 3 \times 7^0 = (262)_{10}$.

- Écriture binaire : deux chiffres 0 et 1

On a $(1011)_2 = 1 \times 2^3 + 0 \times 2^2 + 1 \times 2^1 + 1 \times 2^0 = 11$.

Le prédécesseur de $(1011)_2$ est $(1010)_2$.

Le successeur de $(1011)_2$ est $(1100)_2$.

Le produit $2^2 \times (1011)_2$ vaut $(101100)_2$.

Multiplier par 2^n «rajoute n zéros sur la droite»

Le plus grand entier naturel codé sur un octet (8 bits) est :

Le plus grand entier naturel codé sur un octet (8 bits) est :

$$(1111\ 1111)_2 = 2^8 - 1$$

car son successeur est $(1\ 0000\ 0000)_2 = 2^8$.

Le plus grand entier naturel codé sur un octet (8 bits) est :

$$(1111\ 1111)_2 = 2^8 - 1$$

car son successeur est $(1\ 0000\ 0000)_2 = 2^8$.

Par exemple, en Java, le plus grand entier non signé sur 64 bits est $2^{64} - 1 = 18446744073709551615$.

En Python, les entiers n'ont pas de restriction de taille, au détriment du temps d'exécution.

On utilise 16 chiffres $\{0, 1, \dots, 9, A, B, \dots, F\}$.

$$(5B3F)_{16} = 5 \times 16^3 + 11 \times 16^2 + 3 \times 16^1 + 15 \times 16^0.$$

En regroupant par paquets de 4 bits à partir de la droite, on obtient une correspondance facile entre base 2 et base 16.

$$(5B3F)_{16} = (\underbrace{0101}_5 \underbrace{1011}_{11} \underbrace{0011}_3 \underbrace{1111}_{15})_2.$$

L'avantage de la base 16 est qu'elle est plus lisible pour un humain.

On peut retenir l'approximation suivante

Approximation

On a $2^{10} = 1024 \approx 10^3$.

Ainsi $2^{64} = 2^4 \times 2^{60} = 2^4 \times (2^{10})^6 \approx 16 \times (10^3)^6 = 16 \times 10^{18}$.

On a donc montré «à la main que» $2^{64} \approx 1.6 \times 10^{19}$.

C'est pas mal du tout puisque

$$2^{64} = 18446744073709551614 \approx 1.84 \times 10^{19}.$$

On attribue à chaque caractère un nombre appelé index, qui permet de le repérer. La norme d'encodage que l'on utilisera est UTF 8, qui est rétrocompatible avec les codages ASCII et unicode qui sont plus anciens.

Par exemple, les lettres majuscules 'A', 'B', ..., 'Z' ont pour index 65, 66, ..., 90.

La fonction `ord` («ordinal») de Python permet d'obtenir ces index :

```
>>> ord('A')
65

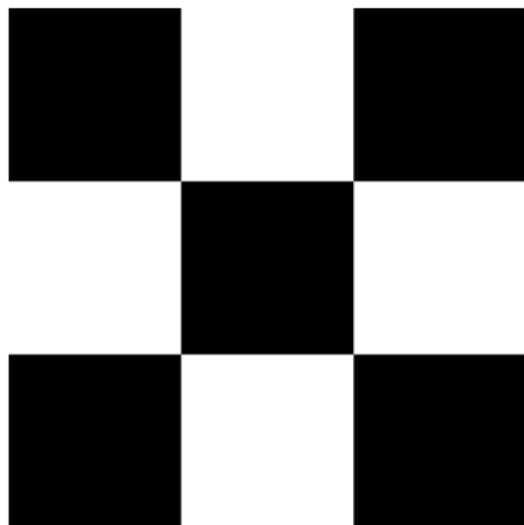
for lettre in "bonjour":
    print(lettre, ord(lettre))

b 98
o 111
n 110
j 106
o 111
u 117
r 114
```

Coder une image

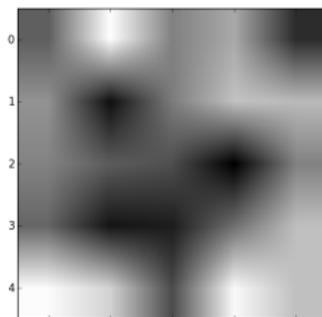
En noir et blanc

$$\begin{pmatrix} 0 & 1 & 0 \\ 1 & 0 & 1 \\ 0 & 1 & 0 \end{pmatrix}$$



En niveau de gris sur 8 bits :
intensité de gris de 0 à 255

$$\begin{pmatrix} 98 & 250 & 130 & 169 & 50 \\ 147 & 23 & 126 & 193 & 185 \\ 133 & 96 & 83 & 7 & 131 \\ 104 & 27 & 35 & 116 & 190 \\ 247 & 208 & 80 & 245 & 190 \end{pmatrix}$$



Efficacité d'un algorithme : calcul de pgcd

Algo 1 : pour calculer le pgcd de a et b (avec $a \geq b$), on cherche quels sont les entiers qui divisent à la fois a et b , puis on renvoie le plus grand de ces diviseurs communs

```
d = 1
Pour k de 2 à b
    Si k divise a et b
        d = k
    Fin Si
Fin Pour
Affiche d
```

Complexité :

Efficacité d'un algorithme : calcul de pgcd

Algo 1 : pour calculer le pgcd de a et b (avec $a \geq b$), on cherche quels sont les entiers qui divisent à la fois a et b , puis on renvoie le plus grand de ces diviseurs communs

```
d = 1
Pour k de 2 à b
    Si k divise a et b
        d = k
    Fin Si
Fin Pour
Affiche d
```

Complexité : on effectue $b - 1$ itérations, et donc $2(b - 1)$ divisions. On parle de complexité linéaire.

Algo 2 : l'algorithme d'Euclide

Complexité :

$u = a$

$v = b$

Tant que v non nul

$r = u \text{ modulo } v$

$u = v$

$v = r$

Fin Tant que

Affiche u

Algo 2 : l'algorithme d'Euclide

$u = a$

$v = b$

Tant que v non nul

$r = u \text{ modulo } v$

$u = v$

$v = r$

Fin Tant que

Affiche u

Complexité : on peut démontrer (théorème de Lamé) que le nombre de divisions est inférieur à

$$5(\log(b) + 1).$$

On parle de complexité logarithmique, c'est beaucoup mieux qu'une complexité linéaire.

Implémentations de l'algorithme d'Euclide

Algo : l'algorithme d'Euclide

Données : a et b deux entiers
naturels

Résultat : le pgcd de a et b

```
u <-- a
```

```
v <-- b
```

```
Tant que v non nul
```

```
    r <-- u modulo v
```

```
    u <-- v
```

```
    v <-- r
```

```
Fin Tant que
```

```
Affiche u
```

Implémentations de l'algorithme d'Euclide

Algo : l'algorithme d'Euclide

Données : a et b deux entiers naturels

Résultat : le pgcd de a et b

```
u <-- a
```

```
v <-- b
```

```
Tant que v non nul
```

```
    r <-- u modulo v
```

```
    u <-- v
```

```
    v <-- r
```

```
Fin Tant que
```

```
Affiche u
```

En langage C

```
int pgcd(int a, int b){
```

```
    int u = a;
```

```
    int v = b;
```

```
    while (v != 0){
```

```
        int r = u % v;
```

```
        u = v;
```

```
        v = r;
```

```
    }
```

```
    return u;
```

```
}
```

```
int main(){
```

```
    int resultat = pgcd(18459,380
```

```
    printf("le pgcd vaut %d", res
```

```
    return 0;
```

```
}
```

Implémentations de l'algorithme d'Euclide

Algo : l'algorithme d'Euclide

Données : a et b deux entiers naturels

Résultat : le pgcd de a et b

```
u <-- a
```

```
v <-- b
```

```
Tant que v non nul
```

```
    r <-- u modulo v
```

```
    u <-- v
```

```
    v <-- r
```

```
Fin Tant que
```

```
Affiche u
```

En Python :

```
def euclide(a,b):
```

```
    while b !=0:
```

```
        r = a % b
```

```
        a = b
```

```
        b = r
```

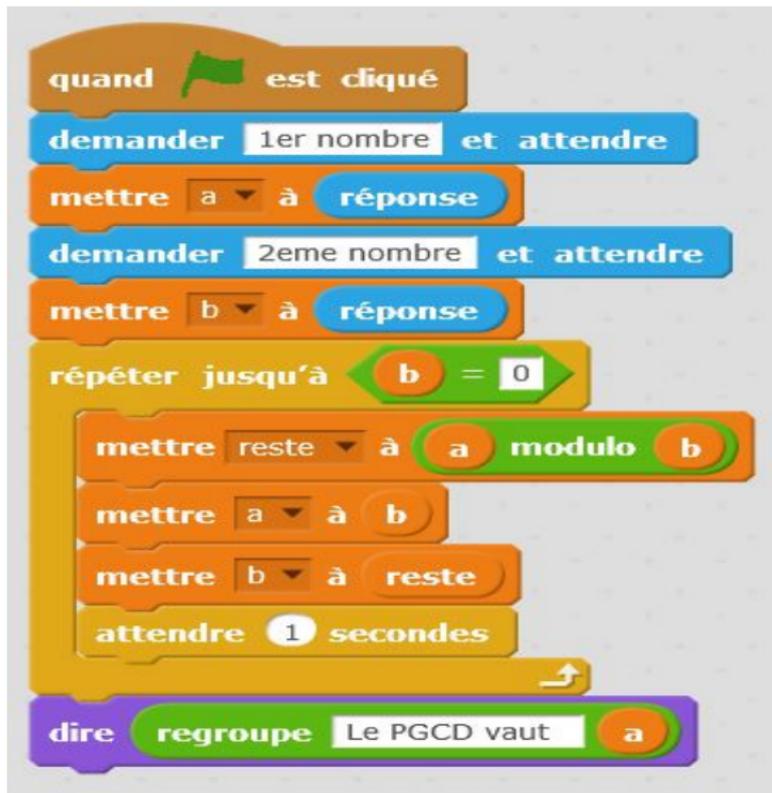
```
    return a
```

```
print(euclide(18459,3809))
```

```
Out[1]: 293
```

Implémentations de l'algorithme d'Euclide

Pour finir en Scratch,



Quelques exemples de langage

Citons par ordre chronologique :

Assembleur, Fortran (1954), Lisp (1958), Cobol (1960) (applications gestions), Basic (1964), C (1970), SQL (1978) (langage de requêtes), C++ (1985), Python (1990), Java (1995), JavaScript (1995), PHP (1995), C# (2000).

Description du langage Python

Python¹ est un langage de programmation créé dans les années 1990 par l'informaticien néerlandais Guido Van Rossum². Citons quelques-une de ses caractéristiques :

-
1. Le nom Python est un hommage à la série anglaise des Monty Python.
 2. Après avoir travaillé chez Google de 2005 à 2012, Guido vient de rejoindre l'entreprise Dropbox.

Description du langage Python

Python¹ est un langage de programmation créé dans les années 1990 par l'informaticien néerlandais Guido Van Rossum². Citons quelques-une de ses caractéristiques :

- langage de script, c'est-à-dire interprété par opposition à langage compilé

1. Le nom Python est un hommage à la série anglaise des Monty Python.
2. Après avoir travaillé chez Google de 2005 à 2012, Guido vient de rejoindre l'entreprise Dropbox.

Description du langage Python

Python¹ est un langage de programmation créé dans les années 1990 par l'informaticien néerlandais Guido Van Rossum². Citons quelques-une de ses caractéristiques :

- langage de script, c'est-à-dire interprété par opposition à langage compilé
- multiplateforme (linux, windows, mac os, android,...)

1. Le nom Python est un hommage à la série anglaise des Monty Python.
2. Après avoir travaillé chez Google de 2005 à 2012, Guido vient de rejoindre l'entreprise Dropbox.

Description du langage Python

Python¹ est un langage de programmation créé dans les années 1990 par l'informaticien néerlandais Guido Van Rossum². Citons quelques-une de ses caractéristiques :

- langage de script, c'est-à-dire interprété par opposition à langage compilé
- multiplateforme (linux, windows, mac os, android,...)
- typage dynamique

1. Le nom Python est un hommage à la série anglaise des Monty Python.
2. Après avoir travaillé chez Google de 2005 à 2012, Guido vient de rejoindre l'entreprise Dropbox.

Description du langage Python

Python¹ est un langage de programmation créé dans les années 1990 par l'informaticien néerlandais Guido Van Rossum². Citons quelques-une de ses caractéristiques :

- langage de script, c'est-à-dire interprété par opposition à langage compilé
- multiplateforme (linux, windows, mac os, android,...)
- typage dynamique
- libre et gratuit

1. Le nom Python est un hommage à la série anglaise des Monty Python.
2. Après avoir travaillé chez Google de 2005 à 2012, Guido vient de rejoindre l'entreprise Dropbox.

Description du langage Python

Python¹ est un langage de programmation créé dans les années 1990 par l'informaticien néerlandais Guido Van Rossum². Citons quelques-une de ses caractéristiques :

- langage de script, c'est-à-dire interprété par opposition à langage compilé
- multiplateforme (linux, windows, mac os, android,...)
- typage dynamique
- libre et gratuit
- doté de nombreuses bibliothèques

1. Le nom Python est un hommage à la série anglaise des Monty Python.
2. Après avoir travaillé chez Google de 2005 à 2012, Guido vient de rejoindre l'entreprise Dropbox.

Description du langage Python

Python¹ est un langage de programmation créé dans les années 1990 par l'informaticien néerlandais Guido Van Rossum². Citons quelques-une de ses caractéristiques :

- langage de script, c'est-à-dire interprété par opposition à langage compilé
- multiplateforme (linux, windows, mac os, android,...)
- typage dynamique
- libre et gratuit
- doté de nombreuses bibliothèques
- syntaxe légère et épurée, qui rend le code assez lisible

1. Le nom Python est un hommage à la série anglaise des Monty Python.
2. Après avoir travaillé chez Google de 2005 à 2012, Guido vient de rejoindre l'entreprise Dropbox.

Description du langage Python

Python¹ est un langage de programmation créé dans les années 1990 par l'informaticien néerlandais Guido Van Rossum². Citons quelques-une de ses caractéristiques :

- langage de script, c'est-à-dire interprété par opposition à langage compilé
- multiplateforme (linux, windows, mac os, android,...)
- typage dynamique
- libre et gratuit
- doté de nombreuses bibliothèques
- syntaxe légère et épurée, qui rend le code assez lisible
- orienté objet

1. Le nom Python est un hommage à la série anglaise des Monty Python.
2. Après avoir travaillé chez Google de 2005 à 2012, Guido vient de rejoindre l'entreprise Dropbox.

Description du langage Python

Python¹ est un langage de programmation créé dans les années 1990 par l'informaticien néerlandais Guido Van Rossum². Citons quelques-une de ses caractéristiques :

- langage de script, c'est-à-dire interprété par opposition à langage compilé
- multiplateforme (linux, windows, mac os, android,...)
- typage dynamique
- libre et gratuit
- doté de nombreuses bibliothèques
- syntaxe légère et épurée, qui rend le code assez lisible
- orienté objet

1. Le nom Python est un hommage à la série anglaise des Monty Python.
2. Après avoir travaillé chez Google de 2005 à 2012, Guido vient de rejoindre l'entreprise Dropbox.

Python, pour quoi faire ?

C'est un véritable langage de programmation utilisé dans l'industrie (par des entreprises telles que Google, Youtube, EDF, NASA, CEA). Par exemple, Odoo (une «Enterprise Resource Planning» pour la gestion des entreprises) ou Mercurial sont des applications entièrement développées en Python.

Python est aussi très prisé par la communauté scientifique.

Enfin, depuis quelques années, il est utilisé dans l'éducation nationale pour l'apprentissage de l'algorithmique.

Le classement TIOBE août 2018 sur l'utilisation des langages

Il mesure la popularité d'un langage

Aug 2018	Aug 2017	Change	Programming Language	Ratings	Change
1	1		Java	16.881%	+3.92%
2	2		C	14.966%	+8.49%
3	3		C++	7.471%	+1.92%
4	5	▲	Python	6.992%	+3.30%
5	6	▲	Visual Basic .NET	4.762%	+2.19%
6	4	▼	C#	3.541%	-0.65%
7	7		PHP	2.925%	+0.63%
8	8		JavaScript	2.411%	+0.31%
9	-	▲	SQL	2.316%	+2.32%
10	14	▲	Assembly language	1.409%	-0.40%
11	11		Swift	1.384%	-0.44%
12	12		Delphi/Object Pascal	1.372%	-0.45%
13	17	▲	MATLAB	1.366%	-0.25%
14	18	▲	Objective-C	1.358%	-0.15%
15	10	▼	Ruby	1.182%	-0.78%
16	9	▼	Perl	1.175%	-0.82%
17	16	▼	Go	0.996%	-0.65%
18	15	▼	R	0.965%	-0.80%
19	13	▼	Visual Basic	0.922%	-0.89%
20	21	▲	PL/SQL	0.702%	-0.51%

Comment obtenir Python ?

Voici plusieurs solutions selon les besoins :

- télécharger la dernière version de Python et son éditeur Idle sur le site officiel.
- télécharger la distribution EDUPYTHON à l'adresse suivante :
<https://edupython.tuxfamily.org/>
- télécharger l'environnement de travail Pyzo qui comporte un éditeur, un interpréteur Python et de nombreuses bibliothèques scientifiques déjà installées :
 - soit sur le site officiel de Pyzo
 - soit à cette adresse
<https://bitbucket.org/pyzo/pyzo/downloads/> pour la version 2015 par exemple

C'est le moment de se lancer

Voilà, il est temps de faire la connaissance avec l'interpréteur Python et d'apprendre à taper du code Python.

La devise Shadok de la semaine



LA PLUS GRAVE MALADIE
DU CERVEAU C'EST DE
REFLÉCHIR.

sa Production