

SNT Chapitre 0 : La programmation



Introduction :

La programmation est littéralement partout où on regarde, des services que vous utilisez aux objets autour de vous, l'informatique n'a jamais été aussi présente dans nos vies qu'aujourd'hui et ce n'est pas près de changer. Tout ce que vous possédez sur vos smartphones et vos ordinateurs utilisent des programmes : votre navigateur Internet (Chrome, Firefox, Opera, etc.), votre système d'exploitation (Android, MacOS, Windows, Linux, etc.), votre appareil photos, vos réseaux sociaux, vos jeux vidéos, etc.

Pour commencer votre parcours dans l'apprentissage de l'informatique, apprendre à coder est le véritable super pouvoir du 21e siècle.



Apprendre la programmation informatique permet de créer et de configurer des entités physiques ou virtuelles en utilisant du code.

Les langages **HTML**, **CSS** et **Javascript** vous permettront de faire *des sites web* avec un rendu visuel directement.

Le langage **Python** est facile à apprendre. Il permet de réaliser des programmes plus généralistes comme par exemple des manipulations de données, des applications, des scripts pour automatiser des tâches, des calculs en tout genre et des jeux. C'est un langage interprété, c'est à dire qu'un autre programme se charge de lire les lignes de codes pour demander à l'ordinateur de les mettre en application. L'avantage quand on commence à apprendre, c'est d'avoir le résultat tout de suite. Avec un langage compilé comme le **C** par exemple, il faut ajouter l'étape de compilation pour obtenir le résultat de notre programme. On se passe du cycle écriture/compilation/test/compilation qui est trop lourd. Même si les ordinateurs sont de plus en plus rapide, Python s'exécute beaucoup plus lentement qu'un programme compilé car la compilation s'effectue juste à temps à la volée. De même la détection d'erreur se fait plus tardivement.

<https://www.tiobe.com/tiobe-index/python/>

D'après le classement du site tiobe, Python est le langage le plus utilisé.

The Python Programming Language

Some information about Python:

- 📈 Highest Position (since 2001): #1 in Aug 2023
- 📉 Lowest Position (since 2001): #13 in Feb 2003
- 🏆 Language of the Year: 2007, 2010, 2018, 2020, 2021



Le programme :



Un programme est une suite **d'instructions simples** donnés à l'ordinateur afin qu'il exécute des actions.

Nos ordinateurs contiennent un circuit électronique qui est le *processeur*. Il permet d'effectuer de petits traitements de base qu'on appelle instructions.

On trouve des instructions mathématiques de base + - * / et d'autres instructions plus complexes comme des opérations permettant de comparer des valeurs ou des caractères = > < .

Il existe aussi des instructions de *saut* et de *boucles* qui permettent de répéter du code.

La *mémoire* dont le processeur se sert pour fonctionner ne peut contenir que des zéros ou des uns.

On doit alors utiliser la base 2, c'est à dire, *le binaire*, pour compter et stocker les *chaînes de caractères*. Heureusement dans la rédaction de nos programmes on peut continuer à utiliser les lettres de l'alphabet ainsi que les chiffres habituels. On dit que Python ou même JavaScript sont des **langages de haut niveau**. Tout sera converti par la suite en *langage machine* qui est en fait une série de zéros et de uns.



Les algorithmes :

L'algorithmique est une branche à part des mathématiques qui consiste à définir et à établir des algorithmes.

Un algorithme est une suite finie d'opérations permettant de résoudre un problème. On renseigne des paramètres, l'algorithme applique une recette (comme en cuisine) et produit un résultat qui peut être une action. Par exemple en voiture, si on appui très fort sur la pédale de freins, l'ABS se déclenche et une série d'algorithmes va déterminer le comportement de la voiture, c'est à dire éviter que celle-ci ne bloque les roues et éviter ainsi le dérapage. Appuyer très fort sur la pédale est un **paramètre**, le blocage des roues en est un second. Enfin éviter le dérapage est le résultat de l'algorithme.

L'intérêt principal des algorithmes est qu'ils sont très utiles lorsqu'ils sont en relation avec des ordinateurs. En effet, ces derniers peuvent exécuter des milliards d'instructions à la seconde, ce qui les rend bien plus rapides qu'un humain.

Ce qu'il faut retenir, c'est qu'un algorithme est une suite d'opérations destinée à résoudre un problème donné. Nous aurons l'occasion d'utiliser des algorithmes dans tous les cours.

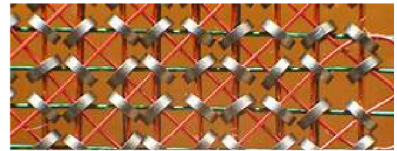
Pour représenter un algorithme indépendamment de tout langage, on utilise ce qu'on appelle un **pseudo-code**. Il s'agit de la description des étapes de l'algorithme en langage naturel.

Les Variables :



Notre ordinateur peut manipuler un peu de tout : du texte, de la vidéo, des nombres, etc. Manipuler du texte ou de la vidéo est complexe, et nécessite en plus de savoir comment manipuler des nombres. Python est un **langage à typage dynamique**, c'est à dire qu'il gère lui-même la quantité de mémoire à utiliser pour stocker toute sorte d'informations comme du texte, des nombres et des objets plus complexes. Une variable est un espace dans la mémoire réservé pour conserver des données. C'est un peu comme une boîte où l'on dépose des informations lorsque l'on veut les conserver pour les modifier plus tard par exemple. On donne un nom à la variable et le système d'exploitation ou **operating system**, remplacera ce nom par *une adresse* de début ainsi qu'une quantité contiguë de cases mémoire. En Python comme en JavaScript, nous n'aurons pas à nous soucier des ressources mémoire car tout est géré pour nous.

La mémoire : Bytes et octets



Un *binary digit* ou encore bit et l'élément élémentaire qui constitue la mémoire.

Par exemple, dans la mémoire RAM, les bits sont regroupés en « paquets » contenant une quantité fixe de bits : des « cases mémoires », aussi appelées *bytes*.

Généralement, les mémoires utilisent des **bytes de 8 bits**. Un groupe de 8 bits s'appelle **un octet**.

Avec un octet, on peut stocker 256 nombres différents. On peut stocker les lettres de l'alphabet, ainsi que les symboles alphanumériques. On peut par exemple stocker les nombres de 0 à 255, ou de -128 à 127, tout dépend de comment on les interprètes.

Pour stocker les nombres de -1024 à 1023 par exemple, on doit utiliser plusieurs octets. Nos données peuvent prendre plusieurs octets qui se suivent en mémoire sans aucun soucis : nos mémoires et nos processeurs sont parfaitement capable d'aller lire plusieurs octets consécutifs et de les manipuler.

Pour retrouver nos informations stockées dans la mémoire, il suffit juste de le demander au système d'exploitation et c'est ce qu'il se passe lorsque l'on utilise les variables. Pour aller plus loin, on peut dire aussi que nos lignes de codes sont des informations, elles sont donc aussi écrites quelque-part en mémoire et c'est vrai pour tout ce qui anime nos ordinateurs et donc tous les programmes ainsi que toutes les données.

Déclarer une variables

Pour déclarer une variable, il nous faut donc un **identifieur** qui sera son nom. Nous respectons quelques conventions qui peuvent différer selon les langages. Voici quelques exemple de nom de variables valide en Python, on ne pourra pas commencer par un chiffre ni utiliser les espaces :

JeSuisUneVariable ← Cette façon de nommer une variable s'appelle le **CamelCase**.

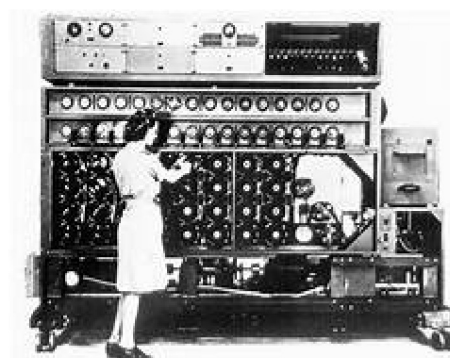
prenom

Nbre_vie

utilisateur1

utilisateur_3

__getAge



La bombe d'Alan Turing

Le type d'une variable :

Une fois que l'on a son nom, il faut lui renseigner son type, si l'on utilise Python ou JavaScript, c'est l'interpréteur lui-même qui s'en charge.

Voici quelques types d'objets simples:

int: *integer* en anglais représentent les entiers. exemple: 6

float: *floating point* en anglais représentent les réels. exemple: 3.14

str: *string* en anglais représentent les chaînes de caractères. exemple: "Hello!"

bool: *booleans* en anglais représentent les booléens. Il ne peuvent avoir que deux valeurs: *True* ou *False* (Vrai ou faux)



```
String prenom = "Paul";  
int age = 35;  
double poids = 74;  
double taille = 1.7;  
String email = "paul.fo@spl.to.t";  
String adresseLigne1 = "45 rue d";  
String adresseLigne2 = "quart. d";  
String adresseLigne3 = "APP 34 B";  
String telephone = "33 63 63 63 63";
```

L'affectation :

Enfin, une fois que notre variable possède un nom et un type, il ne reste plus qu'à écrire la donnée à l'intérieur c'est *l'affectation*. À chaque affectation, la variable va prendre une nouvelle valeur, on dit qu'elle va **muter**.

Vous voilà prêt maintenant à progresser en programmation tout au long de l'année, nous verrons les chapitres suivants :

Le langage de programmation Python.

Les données et le cloud

Internet

Le web ainsi que les langages HTML, CSS et JS (vous allez réaliser une page web)

Les réseaux sociaux

La photographie numérique

Le fonctionnement du GPS

L'informatique embarquée

