

# Computer Science Basic

## BÀI 1. GIỚI THIỆU VỀ COMPUTER SCIENCE VÀ NGÔN NGỮ PYTHON

### 1. Cấu Tạo và Hoạt Động của Máy Tính

#### PHẦN CỨNG



Là những thành phần ta có thể nhìn, cầm và sờ được  
Bao gồm: CPU, RAM, ổ cứng, chuột, bàn phím,...

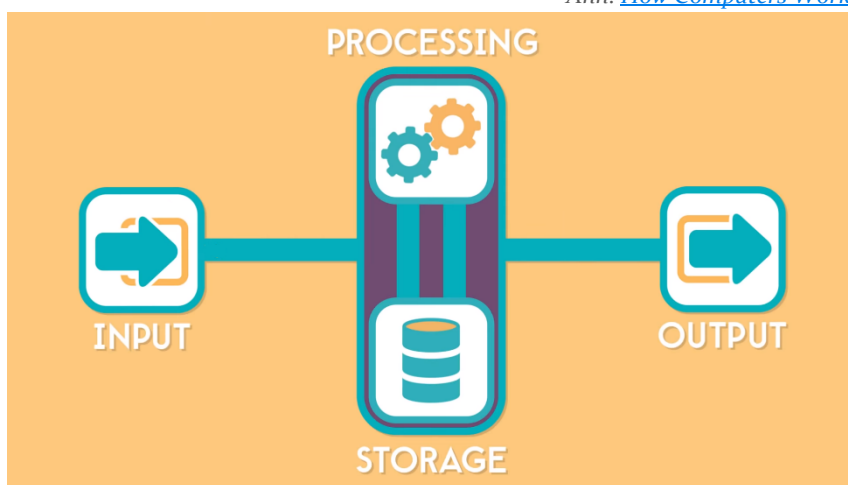
#### PHẦN MỀM



Là ứng dụng chạy bên trong máy tính  
Bao gồm: hệ điều hành, ứng dụng, phần mềm,...

#### CÁCH MÁY TÍNH HOẠT ĐỘNG

1. **Dữ liệu** được đưa vào máy tính dưới dạng các thao tác, các vấn đề, bài toán,...
2. Sau đó, máy tính thực hiện việc **lưu trữ** và **xử lý** dữ liệu.
3. **Kết quả** của quá trình xử lý được trả về theo định dạng phù hợp.



**Thuật toán** chính là cách máy tính tiếp nhận và xử lý dữ liệu để cho ta kết quả mong muốn.  
Thuật toán được đưa vào máy tính bằng công việc **lập trình**.

## 2. Lập Trình và Ngôn Ngữ Lập Trình

Máy tính không thể hiểu được ngôn ngữ tự nhiên của con người. Do đó, ngôn ngữ lập trình được sinh ra để tương tác và hướng dẫn máy tính hoạt động.

### CÁC LOẠI NGÔN NGỮ LẬP TRÌNH

	Đặc trưng	Ví dụ
<b>Ngôn ngữ máy</b>	<ul style="list-style-type: none"> <li>- Là mã nhị phân có thể chạy trực tiếp trên máy tính</li> <li>- Cực kỳ khó hiểu đối với con người</li> </ul>	<pre>0000010000001000 0000000000000000 0000010000001010 0000001000000000</pre>
<b>Ngôn ngữ lập trình bậc thấp</b>	<ul style="list-style-type: none"> <li>- Là ngôn ngữ lập trình, gần với ngôn ngữ máy</li> <li>- Cần được biên dịch thành ngôn ngữ máy để chạy</li> <li>- Dễ hiểu hơn ngôn ngữ máy, nhưng vẫn khó hiểu đối với con người</li> </ul>	<pre>BLE    r24,  r8,    FOR0E ADD    r15,  r8,    r10 SW     r14,  0(r15) ADDI   r8,   r8,    4</pre>
<b>Ngôn ngữ lập trình bậc cao</b>	<ul style="list-style-type: none"> <li>- Là ngôn ngữ lập trình, gần với ngôn ngữ con người</li> <li>- Cần được biên dịch thành ngôn ngữ máy để chạy</li> <li>- Dễ hiểu với con người</li> </ul>	<pre>#include &lt;stdio.h&gt; int main(){     printf("Hello, world\n");     return 0; }</pre>

## 3. Lập Trình với Python

**Python** là một ngôn ngữ lập trình bậc cao; có cú pháp đơn giản, dễ đọc, dễ viết.

Python có thể được dùng để tạo nên:



Ứng dụng di động



Ứng dụng desktop



Ứng dụng web



Game



Công cụ tự động hóa



Trí tuệ nhân tạo

### MÔI TRƯỜNG PHÁT TRIỂN

Để viết và thực thi một chương trình bằng ngôn ngữ Python, ta cần có:

- Một **công cụ soạn thảo** (text editor) để viết code Python
- Một **trình biên dịch** (compiler) để dịch code Python sang ngôn ngữ máy
- Một **môi trường thực thi** (runtime environment) để chạy code đã biên dịch

Trình biên dịch có thể được tải về từ [trang chủ](#) của Python.

Sau đó, ta có thể sử dụng một môi trường phát triển tích hợp như **Visual Studio Code** cho việc soạn thảo và thực thi chương trình.

<b>Chương trình</b>	<ul style="list-style-type: none"> <li>- Một chương trình Python được viết bằng các câu lệnh.</li> <li>- Mỗi lệnh được viết riêng trên một dòng.</li> </ul>
<b>Xuất dữ liệu</b>	<ul style="list-style-type: none"> <li>- Một chương trình đơn giản có thể giao tiếp với người dùng bằng cách xuất ra dữ liệu trên màn hình.</li> <li>- Sử dụng: <pre>print('Hello World! ') print('Welcome User. ', end='\n') print("mindx", "edu", "vn", sep=".") print("{} + {} = {}".format(1, 2, 3))</pre> </li> </ul>
<b>Biến và Kiểu dữ liệu</b>	<ul style="list-style-type: none"> <li>- <b>Biến</b> là nơi chứa dữ liệu. Mỗi biến đại diện cho một giá trị.</li> <li>- Cách đặt tên biến: <ul style="list-style-type: none"> <li>• Không trùng với từ khóa có sẵn</li> <li>• Phù hợp với chức năng của biến</li> <li>• Theo nguyên tắc snake_case</li> </ul> </li> <li>- <b>Kiểu dữ liệu</b> thể hiện đặc điểm của dữ liệu chứa trong biến. Ví dụ: <ul style="list-style-type: none"> <li>• 1970 thuộc kiểu dữ liệu số nguyên   <code>int</code></li> <li>• 3.14 thuộc kiểu dữ liệu số thực   <code>float</code></li> <li>• "MindX" thuộc kiểu dữ liệu chuỗi ký tự   <code>str</code></li> </ul> </li> <li>- Sử dụng: <pre>name = "Michael Jordan" height = 1.98 print(name) print("{} is {} meters tall".format(name, height))</pre> </li> </ul>
<b>Nhập dữ liệu</b>	<ul style="list-style-type: none"> <li>- Một chương trình có thể nhận dữ liệu từ người dùng để xử lý.</li> <li>- Hàm <code>input()</code> nhận dữ liệu nhập từ người dùng và trả về một chuỗi ký tự. Sau đó ta có thể ép kiểu để đưa giá trị về kiểu dữ liệu mong muốn.</li> <li>- Sử dụng: <pre>print("Input your name: ", end="") name = input() print("Input your height: ", end="") height = int(input())</pre> </li> </ul>

## THƯ VIỆN TURTLE

**Turtle** là một thư viện có sẵn của Python, giúp chúng ta luyện tư duy lập trình bằng cách vẽ hình.

Ví dụ: Vẽ một tam giác đều bằng turtle

```
from turtle import *
```

```
forward(100)
left(120)
forward(100)
left(120)
forward(100)
```

```
mainloop()
```



### Một số lệnh và hàm thông dụng

<code>from turtle import *</code>	Khai báo để sử dụng các hàm trong turtle
<code>forward(n)</code>	Tiến trước n pixel
<code>backward(n)</code>	Chạy lùi n pixel
<code>right(degree)</code>	Quay phải degree độ
<code>left(degree)</code>	Quay trái degree độ
<code>mainloop()</code>	Giữ cửa sổ mở sau khi vẽ
<code>shape(name)</code>	Thay đổi hình dạng con trỏ VD: <code>shape('turtle')</code>
<code>pencolor(color)</code>	Đổi màu đường vẽ VD: <code>pencolor('green')</code>
<code>pensize(n)</code>	Đổi kích thước đường vẽ thành n pixel