

PYTHON CHEAT SHEET

Chương 1:NHẬP MÔN

- Công cụ lập trình (Development Tools):**
 - Giới thiệu Python, Jupyter Notebook, Visual Studio Code (VS Code).
- Các bước thiết lập môi trường lập trình Python:**
 - Tải VS Code
 - Cài Extensions: Python
 - Chọn thư mục làm việc (Workplace)
 - Tạo môi trường ảo (Virtual Environment)
- Chạy chương trình đầu tiên:**

Nhập: `print("Hello world! <Tôi là> ")`
→ Hello world! <Tôi là>

Chương 3: Thực Thi Điều Kiện

- Toán Tử So Sánh:**
 - `==`: Bằng với
 - `!=`: Khác với
 - `<`: Nhỏ hơn
 - `<=`: Nhỏ hơn hoặc Bằng
 - `>`: Lớn hơn
 - `>=`: Lớn hơn hoặc Bằng
- Quyết Định**
 - `if` điều_kiện:: Thực thi chỉ khi True.
 - `if... elif... if`:if sai thì xét elif.
 - `if... elif... else::` else là xét điều kiện còn lại.
- Xử Lý Lỗi**
 - `try: ... except::` Thực thi mã trong try. Nếu lỗi, nhảy sang except và tiếp tục chạy.

Chương 4: FUNCTIONS (HÀM TRONG PYTHON)

def – Định nghĩa hàm

- Là từ khóa dùng để tạo (định nghĩa) một hàm mới.
 - Sau `def` là tên hàm, rồi đến cặp ngoặc tròn (có thể chứa tham số).
- Bên dưới là khối lệnh (thân hàm) – phải thụt vào (indent).
- return Trả về giá trị**
- Dùng để kết thúc hàm và trả về một giá trị cho nơi gọi hàm.
 - Sau `return` là biểu thức hoặc giá trị cần trả về.
 - Nếu không có `return` → hàm không trả về gì (void function).

Các hàm chính

- `print()` – In dữ liệu ra màn hình.
`input()` – Nhập dữ liệu từ bàn phím.
`type()` – Kiểm tra kiểu dữ liệu của một biến hoặc giá trị.
`int()` – Chuyển giá trị sang kiểu số nguyên (integer).
`float()` – Chuyển giá trị sang kiểu số thực (float).
`str()` – Chuyển giá trị sang kiểu chuỗi (string).
`max()` – Trả về giá trị lớn nhất trong một dãy (chuỗi, list, tuple...).
`min()` – Trả về giá trị nhỏ nhất trong một dãy.

Chương 2: Biến, Biểu Thức và Câu Lệnh

1. Biến (Variables)

- Mục đích: Là nơi được đặt tên để lưu trữ dữ liệu trong bộ nhớ. Bạn có thể thay đổi giá trị của nó sau này.
- Quy tắc: Tên biến phải bắt đầu bằng chữ cái hoặc dấu gạch dưới, không được dùng từ khóa dành riêng (Reserved Words).

2. Kiểu Dữ Liệu (Data Types)

Các kiểu cơ bản là:

- Integer (int): Số nguyên.
- Float (float): Số thập phân.
- String (str): Chuỗi ký tự (đặt trong " hoặc '').
- Vấn đề kiểu: chuỗi ko thể + với số.
- Chuyển đổi kiểu: Sử dụng `int()`, `float()`, `str()` để chuyển đổi.

3. Biểu Thức & Toán Tử (Expressions & Operators)

Biểu thức là sự kết hợp của biến, hằng số và toán tử.

- Toán tử số học chính:
 - + `+`, `-`, `*` (Nhân).
 - + `/` (Chia).
 - + `**` (Lũy thừa).
 - + `%` (Số dư/Modulus).
- Thứ tự ưu tiên (Precedence) - Quy tắc vàng:
 - + Dấu ngoặc đơn ().
 - + Lũy thừa `**`.
 - + Nhân, Chia, Số dư `*`, `/`, `%`.
 - + Cộng, Trừ `+`, `-`.
 - + Nếu cùng mức ưu tiên, tính từ Trái sang Phải.

4. Câu Lệnh (Statements)

- Câu lệnh gán: Sử dụng dấu bằng (=) để gán giá trị của biểu thức bên phải cho biến bên trái.
- Nhập dữ liệu: Dùng `raw_input()` để đọc dữ liệu từ người dùng. Lưu ý: `raw_input()` luôn trả về chuỗi (str).
- Chú thích: Bắt đầu bằng `#`. Python sẽ bỏ qua nội dung này.

Chương 5:LOOP

- **while** – Tạo vòng lặp khi điều kiện còn đúng.
- **for** – Tạo vòng lặp duyệt qua một tập hợp phần tử.
- **break** – Thoát khỏi vòng lặp ngay lập tức.
- **continue** – Bỏ qua phần còn lại của lần lặp hiện tại, chuyển sang lần kế tiếp.
- **True, False** – Giá trị logic dùng trong điều kiện lặp.
- **None** – Giá trị đặc biệt, biểu thị không có giá trị.

Chương 7: Reading Files

- **open(filename, mode)** – Mở file.
- **read()** – Đọc toàn bộ nội dung file.
- **rstrip()** – Xóa khoảng trắng / \n bên phải chuỗi.
- **startswith(txt)** – Kiểm tra chuỗi bắt đầu bằng txt.
- **in** – Kiểm tra chuỗi con nằm trong chuỗi lớn.
- **for line in fhand:** – Duyệt từng dòng trong file.
- **continue** – Bỏ qua vòng lặp hiện tại.
- **input()** – Nhập dữ liệu từ bàn phím.
- **try / except** – Bắt lỗi khi mở file.
- **quit()** – Thoát chương trình.

Chương 6: Strings

1. Hàm Tích Hợp Sẵn (Built-in Function)

- **len(string)**: Trả về độ dài của chuỗi (số lượng ký tự).

2. Phương Thức Thư Viện Chuỗi (String Library Methods)

Các hàm này là các phương thức được gọi trên một biến chuỗi (ví dụ: `chuoi.lower()`). Chúng không thay đổi chuỗi gốc, mà trả về một chuỗi mới đã được biến đổi.

❖ Chức năng Chuyển Đổi Ký Tự

- **lower()**: Trả về một bản sao của chuỗi với tất cả các ký tự được chuyển thành chữ thường .
- **upper()**: Trả về một bản sao của chuỗi với tất cả các ký tự được chuyển thành chữ hoa .

❖ Chức năng Tìm Kiếm và Thay Thế

- **find(sub)**: Dùng để tìm kiếm vị trí (chỉ mục) xuất hiện đầu tiên của một chuỗi con (sub). Nếu không tìm thấy, nó trả về -1.
- **replace(old, new)**: Thay thế tất cả các lần xuất hiện của chuỗi tìm kiếm (old) bằng chuỗi thay thế (new).
- **startswith(prefix)**: Kiểm tra xem chuỗi có bắt đầu bằng tiền tố (prefix) được chỉ định hay không, trả về **True** hoặc **False**.

❖ Chức năng Xóa Khoảng Trắng (Stripping Whitespace)

- **lstrip()**: Loại bỏ khoảng trắng ở bên trái (bắt đầu) chuỗi.
- **rstrip()**: Loại bỏ khoảng trắng ở bên phải (kết thúc) chuỗi.
- **strip()**: Loại bỏ khoảng trắng ở cả hai đầu (bắt đầu và kết thúc) chuỗi.

Chương 8: List

1. Thao Tác Cơ Bản & Thông Tin (List & Chuỗi)

Các hàm dùng để lấy thông tin về cấu trúc hoặc nội dung, hoặc tạo ra một cấu trúc mới:

- **len(list/string)**: Đếm số phần tử/ký tự.
- **max(list)**: Tìm phần tử lớn nhất.
- **min(list)**: Tìm phần tử nhỏ nhất.
- **sum(list)**: Tính tổng các phần tử (số).
- **index(x)**: Tìm vị trí đầu tiên của x.
- **count(x)**: Đếm số lần xuất hiện của x.
- **list()**: Tạo list rỗng.
- **+** (Toán tử): Nối hai list.
- **in, not in** (Toán tử): Kiểm tra sự tồn tại của phần tử.

2. Thay Đổi/Chỉnh Sửa List (Methods thay đổi tại chỗ)

Các phương thức được gọi trên một list và làm thay đổi trực tiếp nội dung của list đó:

- **append(x)**: Thêm x vào cuối.
- **insert(i, x)**: Chèn x vào vị trí i.
- **remove(x)**: Xóa phần tử x đầu tiên tìm thấy.
- **pop(i)**: Xóa và trả về phần tử tại vị trí i.
- **reverse()**: Đảo ngược thứ tự các phần tử.
- **sort()**: Sắp xếp các phần tử theo thứ tự tăng dần.

3. Thao Tác Chuỗi (String Methods)

Các phương thức được gọi trên một chuỗi để tạo ra list:

- **split()/split(char)**: Tách chuỗi thành list dựa trên khoảng trắng hoặc ký tự phân cách được chỉ định.

Chương 9: Numpy Array

1. `import numpy as np`: Nhập thư viện NumPy (viết tắt là np).
2. `np.array([...])`: Tạo một NumPy Array từ một List Python.
3. `np.zeros(n)`: Tạo Array 1D với n phần tử là số 0.
4. `array_name.shape`: Thuộc tính trả về hình dạng (kích thước/số chiều) của Array.
5. `np.append(a1, a2)`: Nối hai Array, trả về Array mới.
6. `np.insert(a, i, v)`: Thêm giá trị v vào vị trí i, trả về Array mới.

7. `np.delete(a, i)`: Xóa phần tử tại vị trí i, trả về Array mới.
8. `a_data.sort()`: Sắp xếp các phần tử Array tại chỗ (in-place).
9. `a_data * scalar`: Phép nhân từng phần tử (Element-wise) với số vô hướng.
10. `np.min(a) / np.max(a)`: Tìm giá trị nhỏ nhất / lớn nhất trong Array.
11. `np.sum(a)`: Tính tổng các phần tử trong Array.
12. `a_data.reshape(...)`: Thay đổi hình dạng (shape) của Array, trả về Array mới.
13. `np.where(condition)`: Trả về chỉ số (index) của các phần tử thỏa mãn điều kiện.