

BRANCHING

1 Branching là gì?

Branching là cách máy tính kiểm tra một điều gì đó rồi quyết định làm việc này hay việc kia.

Trong đời thường ta cũng thường xuyên đưa ra những quyết định, chẳng hạn như nếu "đói" → "ăn", "kiểm tra" → "học bài",... Máy tính cũng có những quyết định tương tự như vậy

2 Condition

Trước khi thực hiện một quyết định nào đó, máy tính phải hỏi một câu hỏi ĐÚNG hay SAI. Ví dụ: $5 > 3?$ → Đúng; $2 > 10?$ → Sai.

Với điều kiện thì chỉ có 2 đáp án: Đúng hoặc Sai.

3 Nhánh If

Đây là nhánh cơ bản nhất. Đối với con người ta nói: Nếu trời mưa thì mang áo mưa. Đối với máy tính:

```
1  if troi_mua:
2      mang_ao_mua
```

Máy tính kiểm tra điều kiện, nếu điều kiện đúng thì sẽ làm việc bên dưới dấu ":", nếu sai thì bỏ qua không làm gì hết.

4 Nhánh if - else

Luồng suy nghĩ cũng tương tự khi con người nghĩ: Nếu hoàn thành bài tập → đi chơi, nếu không → ngồi học tiếp. Với máy tính chỉ khác ở cách thể hiện bằng dòng code:

```
1  if lam_xong_bai:
2      duoc_choi
3  else:
4      ngoi_hoc
```

Luôn có 1 trong 2 đường được chọn: điều kiện đúng thì chạy `if`, sai thì chạy `else`. Không có chuyện chạy cả hai.

5 Nhánh `if - else - else`

Tương tự nhánh `if - else` nhưng giờ không phải là 2 đường nữa mà là nhiều đường. Ta có thể tưởng tượng như sau:

- Nếu điểm $\geq 8 \rightarrow$ giỏi
- Nếu điểm $\geq 5 \rightarrow$ khá
- Còn lại \rightarrow yếu

Cách thể hiện trong Python sẽ như sau:

```
1  if diem >= 8:
2      gioi
3  elif diem >= 5:
4      kha
5  else:
6      yeu
```

Máy tính sẽ kiểm tra `if`, sai thì kiểm tra `elif`, sai hết thì vào `else`.

Quan trọng: chỉ chạy một nhánh đầu tiên đúng

6 So sánh

Ký hiệu	Tên gọi	Ý nghĩa
>	Lớn hơn	So sánh xem bên trái có lớn hơn bên phải không
<	Nhỏ hơn	So sánh xem bên trái có nhỏ hơn bên phải không
>=	Lớn hơn hoặc bằng	Lớn hơn hoặc bằng nhau
<=	Nhỏ hơn hoặc bằng	Nhỏ hơn hoặc bằng nhau
==	Bằng nhau	Hai giá trị có bằng nhau không
!=	Không bằng nhau	Hai giá trị có khác nhau không

Table 1: Các toán tử so sánh trong Python

7 Ghép nhiều điều kiện

- **AND - "VÀ"**: Cả hai đều phải đúng. Ví dụ: Nếu con ngoan VÀ học giỏi → được thưởng.

```
1         if ngoan and hoc_gioi:
2             duoc_thuong
```

Chỉ cần một cái sai → không được

- **OR - "HOẶC"**: Chỉ cần 1 cái đúng. Ví dụ: Nếu con ốm HOẶC trời mưa → nghỉ học.

```
1         if om or troi_mua:
2             nghi_hoc
```

- **NOT - "KHÔNG"**: Lật ngược đúng và sai

```
1         if not troi_mua:
2             di_choi
```

8 Branching lồng nhau

Giống như khi ta rẽ trái, lại gặp một ngã ba nữa.

```
1         if co_tien:
2             if muon_an:
3                 di_an
4             else:
5                 di_uong
6         else:
7             o_nha
```

Máy tính sẽ suy nghĩ từng tầng một.

9 Branching dùng để làm gì?

Branching giúp chương trình:

- Biết suy nghĩ

- Tạo game
- Kiểm tra dữ liệu
- Kiểm tra dữ liệu
- Quyết định cho phép/không cho phép
- Làm AI, robot, app, web,...

Không có branching → chương trình không có suy nghĩ logic, chỉ chạy từ trên xuống, không biết chọn gì cả.