

BÀI 5: BIẾN, QUY TẮC ĐẶT TÊN VÀ GÁN DỮ LIỆU

Tài liệu hướng dẫn lập trình Python cơ bản

1. Khái niệm về biến trong lập trình Python

Trong lập trình, biến được hiểu đơn giản là một ô nhớ trong bộ nhớ máy tính được đặt tên nhằm lưu trữ dữ liệu tạm thời. Giá trị được lưu trữ trong biến có thể thay đổi liên tục trong suốt quá trình chương trình thực thi. Khác với nhiều ngôn ngữ lập trình khác như C++ hay Java, Python không yêu cầu người viết phải khai báo kiểu dữ liệu cho biến trước khi sử dụng. Kiểu dữ liệu của biến sẽ tự động được hệ thống xác định dựa vào giá trị được gán cho biến đó.

Hãy hình dung biến giống như một chiếc hộp có dán nhãn ở bên ngoài. Nhãn dán chính là tên của biến, còn món đồ chứa bên trong chiếc hộp chính là giá trị hay dữ liệu mà biến đang nắm giữ. Khi chúng ta thay đổi đồ vật bên trong hộp, nhãn dán bên ngoài vẫn giữ nguyên.

2. Quy tắc đặt tên biến trong Python

Việc đặt tên biến trong Python cần phải tuân thủ nghiêm ngặt theo các quy định của ngôn ngữ để tránh xảy ra lỗi cú pháp khi biên dịch.

Tên biến chỉ được phép bao gồm các ký tự chữ cái (từ a đến z, từ A đến Z), các chữ số (từ 0 đến 9) và dấu gạch dưới.

Tên biến bắt buộc phải khởi đầu bằng một chữ cái hoặc dấu gạch dưới, hoàn toàn không được bắt đầu bằng một chữ số.

Python là một ngôn ngữ lập trình có phân biệt chữ hoa và chữ thường, do đó hai tên biến khác nhau về cách viết hoa sẽ được hiểu là hai ô nhớ độc lập.

Tên biến tuyệt đối không được trùng với các từ khóa đã được định nghĩa sẵn của ngôn ngữ Python như if, else, for, while, class, import.

Ví dụ về cách đặt tên biến đúng chuẩn và tên biến vi phạm quy tắc: `tuoi_hoc_sing = 16` `_he_so_luong = 2.34` `10_nam_truoc = "Quá khứ"` (Lỗi vì bắt đầu bằng chữ số) `class = "Lớp học"` (Lỗi vì trùng từ khóa hệ thống)

3. Cách gán dữ liệu cho biến

Để thực hiện việc lưu trữ dữ liệu vào một biến, Python sử dụng toán tử gán là dấu bằng đơn. Cú pháp cơ bản có dạng tên biến nằm ở vế trái, tiếp theo là dấu bằng, và vế phải là giá trị hoặc biểu thức cần tính toán để lưu vào biến.

Khi câu lệnh gán được thực thi, hệ thống sẽ tính toán giá trị ở vế bên phải trước, sau đó mới đem kết quả đó lưu vào ô nhớ của biến ở vế bên trái.

Ví dụ minh họa về các câu lệnh gán dữ liệu cơ bản: `ho_va_ten = "Nguyễn Văn A"` `so_luong_ton_kho = 150` `diem_trung_binh = 8.5` `trang_thai_hoat_dong = True`

Bên cạnh cách gán dữ liệu đơn lẻ cho từng biến, Python còn hỗ trợ phương thức gán đồng thời nhiều giá trị cho nhiều biến khác nhau trên cùng một dòng lệnh nhằm giúp mã nguồn trở nên gọn gàng hơn.

Ví dụ về gán dữ liệu đồng thời cho nhiều biến độc lập: `chieu_dai, chieu_rong = 20, 10` `toa_do_x, toa_do_y, toa_do_z = 0, 0, 0` ho, `ten_dem, ten = "Trần", "Thị", "B"` `tinh_trang, so_lan = "Chờ", 5`

4. Bảng tổng hợp các kiểu dữ liệu phổ biến khi gán cho biến

- **Kiểu số nguyên (int):** Lưu trữ các số nguyên dương hoặc âm, không có phần thập phân. Ví dụ câu lệnh gán dữ liệu cụ thể: `nam_sinh = 2010`
- **Kiểu số thực (float):** Lưu trữ các số có chứa phần thập phân hoặc dạng số mũ. Ví dụ câu lệnh gán dữ liệu cụ thể: `pi = 3.14159`
- **Kiểu chuỗi ký tự (str):** Lưu trữ văn bản, đặt trong cặp dấu nháy đơn hoặc nháy kép. Ví dụ câu lệnh gán dữ liệu cụ thể: `tin_nhan = "Xin chào"`
- **Kiểu logic (bool):** Chỉ nhận một trong hai giá trị duy nhất là True hoặc False. Ví dụ câu lệnh gán dữ liệu cụ thể: `co_loi = False`

Lưu ý quan trọng cho người học: Biến trong Python mang tính chất động. Điều này có nghĩa là bạn hoàn toàn có thể gán một giá trị kiểu chuỗi ký tự vào một biến đang chứa số nguyên mà không gặp bất kỳ lỗi nào. Hệ thống sẽ tự động giải phóng vùng nhớ cũ và chuyển đổi kiểu dữ liệu của biến sang kiểu dữ liệu mới một cách linh hoạt.