

# Biến, hằng và các kiểu dữ liệu



TH.S TRẦN ĐỨC LỢI  
PYTHONVIETNAM.INFO

# Mục đích bài học



- Tìm hiểu về biến, hằng và các biểu thức trong python

# Hằng



- Hằng là các giá trị số, boolean, ký tự hoặc chuỗi không đổi
- Ví dụ: 123, 'tôi yêu em', «Tôi ghét em»
- Số 123 biểu diễn giá trị 123

# Số trong python



- Integer: 2 (32bit = long in C)
- Long (123L)
- Floating point 1.23, 1.2E10 (double in C)
- Complex (1.2-3.4j)
- .real & .imag
- Sys.maxint
- 12j = ?

# Số trong python



- `Int()`
- `Long()`
- `Float()`
- `Complex(re,im)`
- `Abs()`
- `Divmod(x,y)`
- `Pow(x,y)`

# Số trong python



- Module math
- `Math.trunc()`
- `Math.ceil()`
- `Math.floor()`
- `Math.round(1.2345,2)`
- `Math.floor(1+2J)?`

# Số trong python



- $X | y$ : bitwise or
- $X \& y$ : bitwise and
- $X \wedge y$ : bitwise xor
- $X \ll n$ : shift left
- $X \gg n$ : shift right
- $\sim x$ : invert =  $-x-1$

# String



- String là một chuỗi các ký tự.
- Nháy đơn
- Nháy đôi
- Nháy ba
- Ứng dụng
  - 'I am'
  - "I'am"
  - ""I
  - Am""
- Escape sequences (in print command)
  - \
  - \n



# String



- Raw string (r or R)
  - R'new\n'
  - Luôn sử dụng raw string với regular expressions
- Unicode
  - U'unicode'
- Tính immutable
- String concatenation

# Biến



- Hằng là immutable
- Biến: là một vị trí trong bộ nhớ mà chương trình có thể lưu dữ liệu và sau đó đọc ra dựa vào tên của biến.
- Coder được quyền chọn tên của biến theo quy tắc.
- Ví dụ: `x = 1`
- `X = 2`
- `Print x?`

# Nguyên tắc đặt tên



- Phải bắt đầu bằng ký tự chữ hoặc dấu \_
- Phân biệt hoa thường
- Chỉ được bao gồm ký tự, chữ số và \_
- Tránh các từ dành riêng.
- Một số ví dụ:
- `_a`, `_A`, ...
- `A` `b`, `2a`, `a-b`, ...

# Các từ giành riêng



- and del for is raise assert elif from lambda return break else global not try class except if or while continue exec import pass yield def finally in print as with
- không được sử dụng để đặt tên biến, hàm

# Dòng vật lý và dòng logic



- Dòng vật lý?
- Dòng logic?
- Python coi mỗi dòng vật lý là 1 dòng logic (không tường minh)
- Muốn nhiều dòng logic trên 1 dòng vật lý?
- Khuyến cáo

# Căn lề



- Vai trò của Dấu cách trong chương trình python
- Ví dụ:
  - `A = 1`
  - ✦ `A = 2`
  - `B = 2`
- Căn lề hợp lý:
  - 1 tab
  - 4 spaces
  - 2 spaces
  - Dùng một kiểu

# Biểu thức



- Một dòng lệnh (logical line) trong python thường chứa một biểu thức.
- Một biểu thức sẽ bao gồm toán tử (operator) và toán hạng (operand)
- Toán tử là một chức năng nào đó được biểu diễn bằng ký tự hoặc từ khóa.
- Ví dụ toán tử: `+-*/ **`, `//`, `%`
- Toán hạng là dữ liệu để cung cấp cho toán tử thực hiện.
- Ví dụ:
- `2+3`: Thì `2,3` là toán hạng, Dấu `+` là toán tử

# Phép gán



- Sử dụng dấu bằng (=) cho phép gán với biến bên trái, giá trị bên phải.
- Bên phải có thể và thường là một biểu thức (expression)
- Ví dụ:
- $X = 2 * (x + 1)$



# Danh sách toán tử và ý nghĩa



# Thứ tự thực hiện biểu thức



- Thứ tự thực hiện trong biểu thức toán học thông thường?
- Thứ tự thực hiện trong python theo các nguyên tắc cơ bản:
  - Dấu ngoặc
  - Exp
  - Nhân, chia, lấy dư
  - Cộng, trừ
  - Trái sang phải
- Nêu ví dụ đơn giản

# Kiểu trong python



- Thực hiện trong python phép tính  $3/2$  kết quả sẽ bằng 1?
- Giải thích lý do?
- Muốn có kết quả đúng?
- Như vậy kiểu thương số 2 đã được **tự động** convert sang kiểu float **trước khi thực hiện** phép tính hay sau khi thực hiện phép tính?

# Khai báo biến



- Xem xét khai báo biến trong C++
- So sánh với khai báo biến trong python
- Tự nhận kiểu trong python
- Hàm `type()`