Learn To Program With Python 3

Operators & Operands

Operators

- In computer science, an operator is a character or characters that determine the action that is to be performed or considered.

- Types of operators we will use:

  - Assignment – used to assign values to variables (=, +=, -=, etc)
  - Arithmetic – mathematical operations (+, -, *, /, **)
  - Comparison – compares two or more values (>, <, >=, <=, ==, !=)
  - Logic – checks if two statements are true/false (10 > 5 and 5 < 6)

Operands

- An operands is the quantity (or value) on which an operation is to be done.

- In most cases, the “operands” are the values on the left and the right of the “operator”

- 5 + 10
  - + is a mathematical operator for addition
  - 5 is the left operand
  - 10 is the right operand
- 10 != 5
  - != is a comparison operator for “not equal to”
  - 10 is the left operand
  - 5 is the right operand

Operator Precedence

- What if we have two operators in one expression: 4 + 3 * 2
  - Two mathematical operators: addition and multiplication
  - Three operands: 4, 3, and 2
  - How do we solve this?
    - Left to right? 4 + 3 * 2 = 14
    - Right to left? 4 + 3 * 2 = 10
    - Some other order? PEMDAS (Parentheses, Exponents, Multiplication & Division, Addition * Subtraction)

Operator Precedence: 4 + 3 * 2

- PEMDAS is a way we teach children the order in which operators are used.
  - Also known as: order of operations, or operator precedence
- Using PEMDAS: 4 + 3 * 2 = 10
- PEMDAS is taught in mathematics, and thus only addresses mathematical operators.
- Harder Example: my_variable = 4 + 3 > 10
  - How many operators are there?
  - What types of operators are there?
  - How many operands are there?
  - What will my_variable evaluate to?

Operator Precedence

Operator	Description
()	Parentheses (grouping)
f(args...)	Function call
x[index:index]	Slicing
x[index]	Subscription
x.attribute	Attribute reference
**	Exponentiation
~x	Bitwise not
+x, -x	Positive, negative
*, /, %	Multiplication, division, remainder
+, -	Addition, subtraction
<<, >>	Bitwise shifts
&	Bitwise AND
^	Bitwise XOR
|	Bitwise OR
in, not in, is, is not, <, <=, >, >=, <>, !=, ==	Comparisons, membership, identity
not x	Boolean NOT
and	Boolean AND
or	Boolean OR
lambda	Lambda expression
Assignment	=

- The higher up the list, the higher the order of precedence. Example: see how Multiplication, division, and remainder are ABOVE addition, and subtraction?

Source: https://www.mathcs.emory.edu/~valerie/courses/fall10/155/resources/op_precedence.html

Operator Precedence: 4 + 3 * 2

- Operators: = , + , >

- Addition is done first: 5 + 3 = 8
  - my_variable = 8 > 10

```
>>> my_variable = 5 + 3 > 10
>>> print(my_variable)
False
```

- Comparison is done second: 8 > 10
  - my_variable = False

- Assignment is done last
  - The value False is assigned to the variable “my_variable”

Operator Associativity

- What do we do with operators of the same precedence?
- Associativity is the order in which an expression is evaluated that has multiple operators of the same precedence
- Example: 5 + 5 – 5
- **Almost all operators in python have** left-to-right associativity**.** They will be evaluated from left to right.
- Don’t worry too much about associativity, just know that is a thing

Learn To Program With Python 3

Expressions & Statements

Expressions

- Expression example: 5 + 3 * 3

- An expression is a combination of one or more constants, variables, operators, and functions that Python evaluates to produce a NEW value.

- 5 + 3 * 3 is evaluated to produce the new value 14

Statement

- A statement is a syntactic unit that expresses some action to be carried out.

- Examples we have seen:

  - print(“Hello World”) – Action: Print the string “Hello World” to the console.

  - my_variable = 5 – Action: Assign the value 5 to the variable my_variable

- Expressions evaluate to something (produce a new value)
- Statements do something (print to the console, assign a variable)

Learn To Program With Python 3

Learn Full Stack Web Development

www.projectfullstack.com