Operations

Operations are a fundamental aspect of programming and are used to perform mathematical logical and comparison operations in Python. This lesson will cover some of the most commonly used operations in Python.

Arithmetic Operations

Arithmetic operations are used to perform mathematical operations in Python. These include:

- Addition (+)
- Subtraction (-)
- Multiplication (*)
- Division (/)
- Floor Division (//)
- Modulus (%)
- Exponentiation (**)

Addition

```
a = 2
b = 3
c = a + b
print(c)
```

Subtraction

```
a = 4
b = 2
c = a - b
print(c)
```

Multiplication

```
a = 2
b = 3
c = a * b
print(c)
```

Division

```
a = 8
b = 4
c = a / b
print(c)
```

Floor Division

```
a = 9
b = 4
c = a // b
print(c)
```

Modulus / Remainder

```
a = 5
b = 2
c = a % b
print(c)
```

Exponentiation

```
a = 2
b = 3
c = a ** b
print(c)
```

Assignment Operators

All of the above examples use the = assignment operator to assign the result of the operation to a new variable. However, when updating variables, there are other assignment operators that can be used to perform operations and assign the result to the same variable - in a single step. These include:

- Addition (+=)
- Subtraction (-=)
- Multiplication (*=)
- Division (/=)
- Modulus (%=)
- Exponentiation (**=)

For example:

```
a = 2
b = 3
a += b
print(a)
```

Comparison Operations

Comparison operations are used to compare two values and return a boolean value. These include:

- Less than (<)
- Less than or equal to (<=)
- Equal to (==)
- Not equal to (!=)
- Greater than (>)
- Greater than or equal to (>=)

Less than

```
a = 2
b = 3
c = a < b
print(c)
```

Less than or equal to

```
a = 2
b = 3
c = a <= b
print(c)
```

Equal to

```
a = 2
b = 2
c = a == b
print(c)
```

Not equal to

```
a = 2
b = 3
c = a != b
print(c)
```

Greater than

```
a = 2
b = 3
c = a > b
print(c)
```

Greater than or equal to

```
a = 2
b = 3
c = a >= b
print(c)
```

Logical Operations

Logical operations are used to combine boolean values. These include:

- And (`and`)
- Or (`or`)
- Not (`not`)

And

```
a = True
b = False
c = a and b
print(c)
```

Or

```
a = True
b = False
c = a or b
print(c)
```

Not

```
a = True
b = not a
print(b)
```

Order of Operations

Python follows the PEDMAS order of operations for arithmetic expressions, which is as follows:

1. Parentheses (`()`)
2. Exponents (`**`)
3. Division (`/`)
    - Floor Division (`//`) and Modulus (`%`) are also considered division operations.

4. Multiplication (`*`)
5. Addition (`+`)
6. Subtraction (`-`)

For example, `2 + 3 * 4` is evaluated as `2 + (3 * 4) = 14` and not `(2 + 3) * 4 = 20`.

Arithmetic operations take precedence before logical operations. Of the logic operations we covered so far, the order of operations is as follows:

1. Equality Operators (`==`, `!=`)
2. Comparison Operators (`<`, `<=`, `>`, `>=`)
3. Boolean NOT (`not`)
4. Boolean AND (`and`)
5. Boolean OR (`or`)

For example `4 < 3 and 3 > 4 or 4 > 3` is evaluated as `((4 < 3) and (3 > 4)) or (4 > 3) = True` and not `(4 < 3) and ((3 > 4) or (4 > 3)) = False`.

Exercises

For each expression, first write down your prediction. Then, evaluate the expression in Python and compare your answer to your written prediction. Try to understand why the result is what it is.

1. What are the results of the following operations?

    - `5 + 6`
    - `1 - 3`
    - `4 * 4`
    - `8 / 10`
    - `11 // 4`
    - `16 % 3`
    - `4 ** 2`
    - `6 < 2`
    - `8 <= 8`
    - `4 == 9`
    - `9 != 8`
    - `8 > 1`
    - `3 >= 4`
    - `True and False`
    - `False or False`
    - `not False`

2. What are the results of the following operations?

  - `5 + 6 * 2`
  - `(5 + 6) * 2`
  - `5 + 6 * 2 == 17`
  - `5 + 6 * 2 == 18`
  - `5 + 6 * 2 == 17 and 5 + 6 * 2 == 18`
  - `5 + 6 * 2 == 17 or 5 + 6 * 2 == 18`
  - `not 5 + 6 * 2 == 17`

3. What are the results of the following operations?

   - `5 + 6 * 2 == 17 and 5 + 6 * 2 == 18 or 5 + 6 * 2 == 17`
   - `5 + 6 * 2 == 17 or 5 + 6 * 2 == 18 and 5 + 6 * 2 == 17`
   - `not 5 + 6 * 2 == 17 and 5 + 6 * 2 == 18 or 5 + 6 * 2 == 17`
   - `not 5 + 6 * 2 == 17 or 5 + 6 * 2 == 18 and 5 + 6 * 2 == 17`
   - `not 5 + 6 * 2 == 17 and 5 + 6 * 2 == 18 or not 5 + 6 * 2 == 17`
   - `not 5 + 6 * 2 == 17 or 5 + 6 * 2 == 18 and not 5 + 6 * 2 == 17`
   - `not 5 + 6 * 2 == 17 and not 5 + 6 * 2 == 18 or not 5 + 6 * 2 == 17`
   - `not 5 + 6 * 2 == 17 or not 5 + 6 * 2 == 18 and not 5 + 6 * 2 == 17`

Further Reading

Check out these links for further reading on operations in Python:

- W3Schools
- Programiz
- Real Python
- GeeksforGeeks
- Python Documentation

Also, check out these links for further reading on assignment operators in Python: - GeeksforGeeks - Real Python - Python Documentation

Finally, check out these links for further reading on order of operations in Python: - GeeksforGeeks - Real Python - Python Documentation

Additionally, you can use search terms such as “Python operators” or “Python operations” to find more resources online. Remember, there are many ways to learn Python and you should always try to find the resources that work best for you.