Python has a a lot of built-in operations for you to perform various operations on values and variables. Here are the the list of common operands that you can use as a quick reference.

Examples of all operators will be shown at the bottom

Arithmetic

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

Examples

```
# Addition
>> 5 + 8
13

# Subtraction
>> 8 - 5
3

# Multiplication
>> 5 * 8
40

# Division
>> 10/5
2

# Modulus
# Modulus performs integer division and and returns the remainder as the
result
# Note: Do not confuse floating point and remainder
# To find the remainder, try doing long division but do go further than
1 decimal
>> 10%3 # 10/3 = 3.3333... floating point, 10/3 = 3 remainder 1
```

```
1

>> 7%4 # 7/4 = 1.75 floating point, 7/4 = 1 remainder 3
3

>> 25 % 6 # 25/6 = 4.16666667 floating point, 25/6 = 4 remainder 1
1

# Exponention
>> 5**2 # 5 * 5 OR 5^2
25

>> 8**3 # 8 * 8 * 8 OR 8^3
512

# Floor Division
# Floor Division returns the quotient of the result discarding the
remainder
>> 10//3 # 10/3 = 3 remainder 1
3

>> 7//4 # 7/4 = 1 remainder 3
1

>> 25//6 # 25/6 = 4 remainder 1
4
```

Assignment

- `=` (Assign a value to a variable)
- `+=` (Add and assign)
- `-=` (Subtract and assign)
- `*=` (Multiply and assign)
- `/=` (Divide and assign)
- `%=` (Modulus and assign)
- `**=` (Exponentiate and assign)
- `//=` (Floor divide and assign)

```
# =
x = 5 # Assigns the integer 5 into x

# +=
x += 10 # Assigns x + 5 into x (x = x + 10)

# -=
x -= 34 # Assigns x - 34 into x (x = x - 34)

# *=
x *= 5 # Assigns x * 5 into x (x = x * 5)

# /=
x /= 9 # Assigns x / 9 into x (x = x / 9)

# %=
x %= 2 # Assigns x % 2 into x ( x = x % 2)

# **=
x **= 3 # Assigns x ** 3 into x (x = x ** 3)

# //=
x //= 2 # Assigns x // 2 into x (x = x // 2)
```

Comparison

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

```
# ==
>> 5 == 5 # True | 5 is equal to 5
>> 5 == 3 # False | 5 is not equal to 3
>> "Hello" == "Hello" # True
```

Logical

- `and` (Logical AND)
- `or` (Logical OR)
- `not` (Logical NOT)

```
# and
True and False # False
True and True # True
False and True # False
False and False # False

# or
True or False # True
False or True # True
False or False # False
True or True # True

# not
not True # False
not False # True
```

Bitwise

- `&` (Bitwise AND)
- `|` (Bitwise OR)
- `^` (Bitwise XOR)
- `~` (Bitwise NOT)
- `<<` (Bitwise left shift)
- `>>` (Bitwise right shift)

Do some further reading

Membership

- `in` (Check if an element is present in a sequence)
- `not in` (Check if an element is not present in a sequence)

```
# Create list
>> x = [1,2,3,4]

# in
>> 3 in x # True
>> 5 in x # False

# not in
>> 3 not in x # False
>> 5 not in x # True
```

Identity

- `is` (Check if two objects are the same object)
- `is not` (Check if two objects are not the same object)

Combining operations

```
x = 10
y = 3

z = x + y # 13
z = (x/3) * y # 10
```

See Also

2. Functions