# Numbers and Operators

**Numbers**

**Arithmetic Operators**

**Assignment Operators**

**Comparison Operators**

**Casting**

***References***

---

## Numbers

**Python has four types of numbers:**

```
Integers - or Int, is a whole number, positive or negative, without decimals.

Floats - or "floating point number" is a number, positive or negative, containing one or mor
e decimals.

Complex - are numbers written with a "j" as the imaginary part:

Boolean - True or False.
```

In [3]:
```
x = 13    # int
y = 4.6   # float
z = 5j    # complex
a = True  # boolean

print(type(x))    # The type() function is used for troubleshooting.
print(type(y))    #  by printing the variable data type:
print(type(z))    #                                    <class '??'>
print(type(a))
```

```
<class 'int'>
<class 'float'>
<class 'complex'>
<class 'bool'>
```

---

## Arithmetic Operators

Arithmetic operators are used with numeric values to perform common mathematical operations:

| Operator | Name | Example |
|---|---|---|
| + | Addition | x + y |
| - | Subtraction | x - y |
| * | Multiplication | x * y |
| / | Division | x / y |
| % | Modulus | x % y |
| ** | Exponentiation | x ** y |
| // | Floor division | x // y |

In [2]:
```
x = 7
y = 2

print(x + y)
```

```
9
```

In [3]:
```
x = 7
y = 2

print(x - y)
```

```
5
```

In [4]:
```
x = 7
y = 2

print(x * y)
```

```
14
```

In [5]:
```
x = 7
y = 2

print(x / y)
```

```
3.5
```

In [6]:
```
x = 7
y = 2

print(x % y)      #  yields the remainder when the first operand is divided by the second
```

```
1
```

In [7]:
```
x = 7
y = 2

print(x ** y)     #  x to the power y
```

```
49
```

In [8]:
```
x = 7
y = 2

print(x // y)     # returns floor value for both integer and floating point arguments
```

```
3
```

---

## Assignment Operators

Arithmetic operators are used with numeric values to perform common mathematical operations:

| Operator | Name | Example |
|---|---|---|
| + | Addition | x + y |
| - | Subtraction | x - y |
| * | Multiplication | x * y |
| / | Division | x / y |
| % | Modulus | x % y |
| ** | Exponentiation | x ** y |
| // | Floor division | x // y |

---

## Comparison Operators

Arithmetic operators are used with numeric values to perform common mathematical operations:

| Operator | Name | Example |
|---|---|---|
| + | Addition | x + y |
| - | Subtraction | x - y |
| * | Multiplication | x * y |
| / | Division | x / y |
| % | Modulus | x % y |
| ** | Exponentiation | x ** y |
| // | Floor division | x // y |

---

## Casting

Sometimes we need to change the data type of the variable. This is done by casting. Casting gives you the ability to change the data type to the type you need. There is three types of casting functions:

**int() - constructs an integer number from an integer, a float, or a string**

**float() - constructs a float number from an integer, a float or a string (providing the string represents a float or an integer)**

**str() - constructs a string from a wide variety of data types, including strings, integer and float**

In [9]:
```
x = 13    # int
y = 4.6   # float
z = 5j    # complex

#convert from int to float:
a = float(x)

#convert from float to int:
b = int(y)

#convert from int to complex:
c = complex(x)

print(a)
print(b)
print(c)

print("x is an integer but cast as: ", type(a))
print("y is a float but cast as: ", type(b))
print("z is a complex but cast as: ", type(c))
```

```
13.0
4
(13+0j)
x is an integer but cast as:  <class 'float'>
y is a float but cast as:  <class 'int'>
z is a complex but cast as:  <class 'complex'>
```

---

## Resources

### Sources for deeper learning:

1. W3Schools.com Python Operators

1. Python Tutorial: Integers and Floats - Working with numbers.

**Sources used in this document:**

University of Hawaii

W3Schools.com Python Operators