Numbers and Math

We encountered integers (int) and floating-point numbers (float) in the last section

integers (int) are numbers without a decimal place

floating-point numbers (float) are number with a decimal place

Arithmetic Operators

Aside from the +, -, /, * operators

You can raise a number to a power using the ** operator

In [1]:
```
2**3
```

Out[1]: 8

You can get the remainder of a division operation using the modulus operator (%)

In [2]:
```
20 % 7
```

Out[2]: 6

Formatting floats

You can show floats with a specified number of decimal places

In [3]:
```
pi = 3.14159265359
phrase = f"The value of pi with 2 decimal places is {pi:.2f}"
print(phrase)
```

```
The value of pi with 2 decimal places is 3.14
```

Functions

You have already seen some built-in functions like print() and len()

You can write your own functions, for example

In [4]:
```
def multiply(x, y):
    product = x * y
    return product

print(multiply(4, 3))
```

```
12
```

Note that when Python encounters the return statement it will stop running the function and any code below it in the function will not be run

In [5]:
```
def multiply(x, y):
    product = x * y
    return product
    product = product**19

print(multiply(4, 3))
```

```
12
```

Using the help() function on built-in Python functions displays a description of the function

In [6]:
```
help(len)
```

```
Help on built-in function len in module builtins:

len(obj, /)
    Return the number of items in a container.
```

You can add descriptions to user-defined functions using docstrings

In [7]:
```
def multiply(x, y):
    """Returns the product of two numbers"""
    product = x * y
    return product

help(multiply)
```

```
Help on function multiply in module __main__:

multiply(x, y)
    Returns the product of two numbers
```