

M1 – Self preparation (After Lec#1 and before Lec#2)

Don't attempt to review this self-study until we introduce Wing IDE in class which should happen during Lec#1.

In this Module we will explore basic python mathematical operations and functions.

Try the following in the interaction window:

10 / 2

11 / 3

11.345 / 4

Write down the purpose of `x / y`:

Try the following in the interaction window:

10 // 2

11 // 3

11.25 // 4

Write down the purpose of `x // y`:

Note: `//` usually used with Integers.

Try the following in the interaction window:

10 % 5

17 % 3

2 % 11

0 % 11

Note: although `%` works with negatives and Float, but we will use it only with natural numbers.

Write down the purpose of `a % b`:

Try the following in the interaction window:

```
2 ** 3
```

```
2.5 ** 4
```

```
2 ** -4
```

```
8 ** -0.5
```

Write down the purpose of `x ** y`:

Try the following in the definition window:

```
abs(-3.45)
```

```
abs(10)
```

```
abs(0)
```

Write down the purpose of `abs(n)`:

Try the following in the interaction window:

```
len("cs 116")
```

```
len('cs 116')
```

```
len("")
```

Write down the purpose of `len(s)`:

Compare your answer with the result of calling `help(len)` in the interaction window

Try the following in the interaction window:

```
type(6)
type(3.78)
type(4.0)
type("abc")
```

Write down the purpose of `type(t)`:

Try the following in the interaction window:

```
min(2,0,-1,10,5)
min(2)
max(2,0,-1,10,5)
max(2)
```

Write down the purpose of `max` and `min`:

import math module by writing: `import math` in the interaction window

Try each of the following expressions in the interaction window and write down its purpose:

`math.pi`

`math.e`

`math.sqrt(100)`

`math.sqrt(77)`

`math.sqrt(-10)`

`math.log(32,2)`

`math.floor(123.987)`

`math.floor(-10.23)`

`math.factorial(5)`

`math.factorial(0)`

`math.factorial(-3)`

`math.cos(math.pi)`

`math.exp(3)`

`math.pow(3,4)`

`math.pow(-4,2)`

`math.pow(3.75, 2.5)`

Purposes:

`sqrt:`

`log:`

`floor:`

`factorial:`

`cos:`

`exp:`

`pow:`

Note1: Make sure, for each mathematical function, you try it with integers, floats, positives, negatives, zeros, and different number of arguments to explore its behavior completely.

Note2: compare your definition for each function with the formal definition. For example: after you write your definition for `math.sqrt`, write in the interaction window `help(math.sqrt)` and compare.

Note3: write `dir(math)` in the interaction window, skip all the names in the list that start with `_`, what is left are valid mathematical functions that you can explore by yourself.

Casting and conversion functions:

Try the following in the interaction window, write the purpose of each function:

```
float(1)
float(10.34)
float("34.678")
float("2.4.5")
float("34")
int(4.7)
int(-12.7)
int(-12.3)
int("45")
int("2.3")
str(4)
str(34.76)
str(-5)
```

Purposes:

float:

int:

str: