Integers, floats, and basic math

SHARP SIGHT

What you'll learn

- Integers and floating point numbers

- Math operators
  - operators that work on integers and floating point numbers (AKA, floats)

- How to do simple calculation in Python
  - calculate with “literal” numbers
  - calculate with variables
  - order of operations

Integers and floating point numbers

Floats and integers are numeric data types

- Integers and floating point numbers are two of Python's primary built-in data types
  - they are numbers on which we can perform calculations

data type	description	example
`int`	An integer (whole number)	`42`
`float`	A number with a decimal	`2.72`

We can perform calculations with floats and integers

- Numeric calculation is one of the most basic tasks we can perform in Python
  - addition
  - subtraction
  - etc

- We perform calculations on numeric data with "math operators"

Basic math operators

Python has several basic math operators

Basic math operators

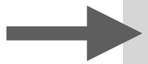
operator	what it does	example	result
+	addition	`3 + 2`	`5`
-	subtraction	`9 - 3`	`6`
*	multiplication	`2 * 6`	`12`
/	division	`15 / 2`	`7.5`
//	division (truncating)	`15 // 2`	`7`
%	modulus	`15 % 2`	`1`
**	exponentiation	`4**3`	`64`

The – sign specifies negative numbers

- Integers are positive by default
  - Inserting a + sign does nothing

- The – sign specifies negative numbers

If we put a – in front of an integer or float, it creates a negative number



```
+789
789


-543
-543
```

There are several ways to "calculate" in Python

- We can calculate with "literal" numbers
- We can calculate with variables
- We can calculate with a mix of numbers and variables

How to calculate with literal numbers

- Use the math operators with numbers

```
5 + 2
7


15 / 2
7.5
```

- Note: here, we calculating on literal *integers*

How to calculate with variables

`a` and `b` are both variables →

```
a = 2
b = 7
```

Here we're calculating with two variables →

```
a + b
9
```

How to calculate with a mix of numbers and variables

a is a variable →

```
a = 2
```

Here we're calculating with a combination of a variable and an integer →

```
a + 3
5
```

Use the = sign to assign new values to variables

- Calculations don't automatically change the values stored in variables

- To change a variable value, you need to assign a new value using **=**

Here, we're assigning an initial value to a

```
a = 5
```

Here, we're assigning a new value to a using the equal sign (=)

```
a = a + 7
print(a)
12
```

Use the += operator to calculate and assign at the same time

The += operator is a way to calculate and assign at the same time

```
#Traditional method
a = 5
a = a + 7
12


#Shorthand: assignment with calculation
b = 3
b += 1
b
4
```

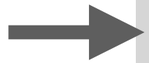
There are similar operations for multiplication, subtraction, etc. Try them!

Order of operations

You can perform multiple math operations on a single line, in series

- Execute as many operations as you want

Here, we're performing 2 operations on the same line



```
5 + 2 + 9
16
```

Math operators have an *order of operations*

- Python has an order that determines which operations are performed first

Higher Precedence

Lower Precedence

An abridged list of operators, with precedence

operator	notes
`[], {}, ()`	brakets, parenthesis
`**`	exponentiation
`*, /, //, %`	multiplication, division, modulus
`+, -`	addition / subtraction

Note: the complete list includes: logical operators, comparisons, etc

Order of operations will change your calculations

Note that the numbers are the same, but the result changes depending on where we put the parenthesis

```
2 + 5 * 3
17


(2 + 5) * 3
21


2 + (5 * 3)
17
```

Be careful!

Using explicit parenthesis is a "best practice"

- It's always best to use explicit parenthesis in your calculations
- This makes your code easier to read
- Also makes your code less error prone

✗
```
2 + 5 * 3
17
```

✓
```
(2 + 5) * 3
21
```

✓
```
2 + (5 * 3)
17
```

Recap

Recap of what we learned

- Math operators
  - Operators that work on integers and floating point numbers

- How to do simple calculation in Python
  - calculate with “literal” numbers
  - calculate with variables
  - order of operations