Questions for you! (Poll Everywhere!)

https://pollev.com/yujeansong834

Please copy and paste this URL to attend the Quiz!

Questions for you!

Respond at **PollEv.com/yujeansong834**

1+1 = ?

Poll locked. Responses not accepted.

1+1 = ?

- Error
- 2
- 11
- None of the above

Total Results: 0

Questions for you!

Respond at **PollEv.com/yujeansong834**

"1" + "1" = ?

Respond at **PollEv.com/yujeansong834**

"1" + "1" = ?

- "2"
- "11"
- 2
- 11
- None of the above

4

Total Results: 0

Questions for you!

Respond at **PollEv.com/yujeansong834**

1 + "1" = ?

1

Respond at **PollEv.com/yujeansong834**

1 + "1" = ?

- Error
- 2
- 1"1"
- "2"
- None of the above

Total Results: 0

Questions for you!

Respond at **PollEv.com/yujeansong834**

"1" * 3 = ?

5

Respond at **PollEv.com/yujeansong834**

"1" * 3 = ?

- "3"
- "1""1""1"
- "111"
- 3
- None of the above

Total Results: 0

MBSI Python Coding Workshop #2

—

Data Types and Math - Yujean Song and Joshua Chan

Week 1 Recap

Week 1 Concepts	Definition	Example
string	Characters within quotation marks	`"Hello there"` `"General Kenobi!"`
print()	A function that takes data inside the brackets and displays it back to the user	`print("It's over Anakin! I have the high ground!")` `It's over Anakin! I have the high ground!`

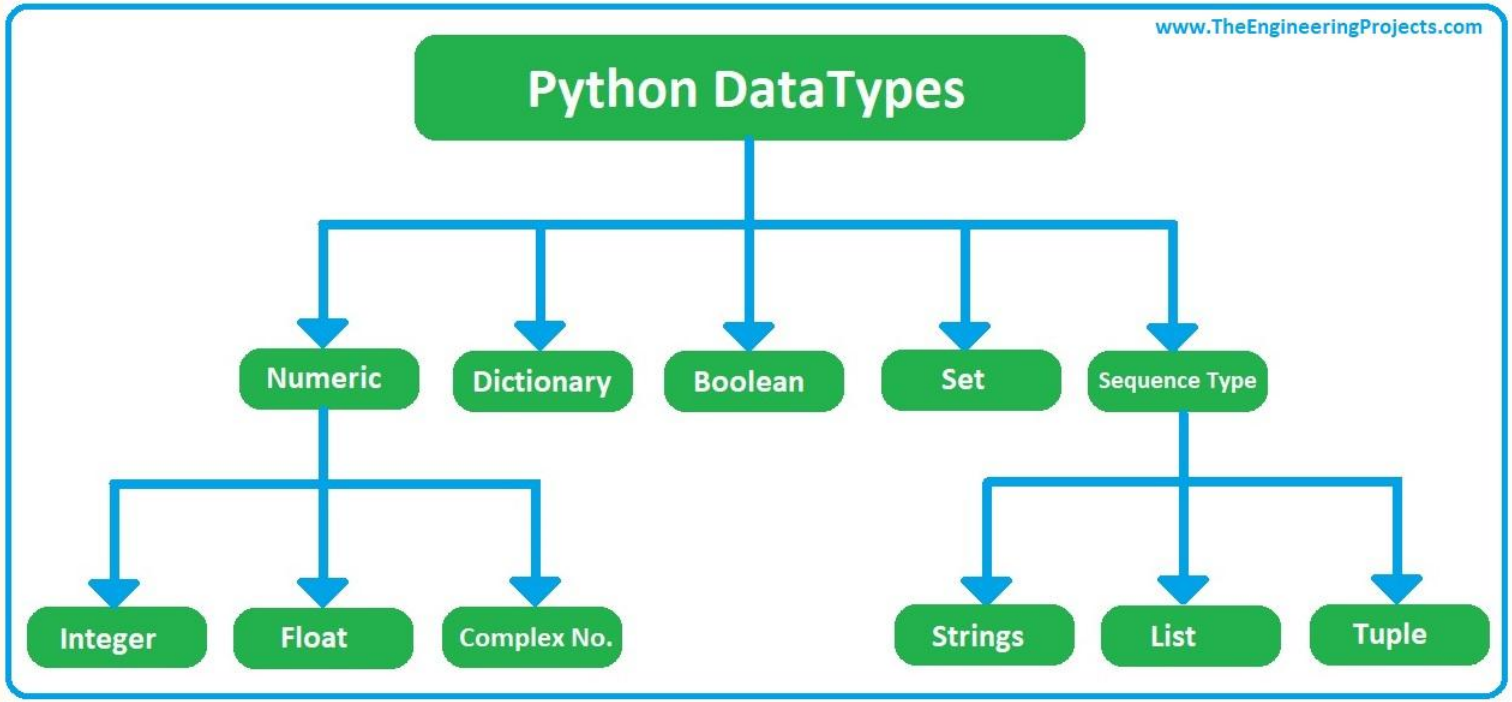
variable	A “container” that stores data values	Variable → `pain_onset = "5 hours ago"` ← Data value
		`print(pain_onset)`
		`5 hours ago`
		Is defined as

input()	A function that prompts the user to enter an input	`x = input("Where does the chest pain radiate?")`
		`print(x)`
		`Where does the chest pain radiate?To my left shoulder, Doc!`
		`x = input("Where does the chest pain radiate?")`
		`print(x)`
		`Where does the chest pain radiate?To my left shoulder, Doc!`
		`To my left shoulder, Doc!`

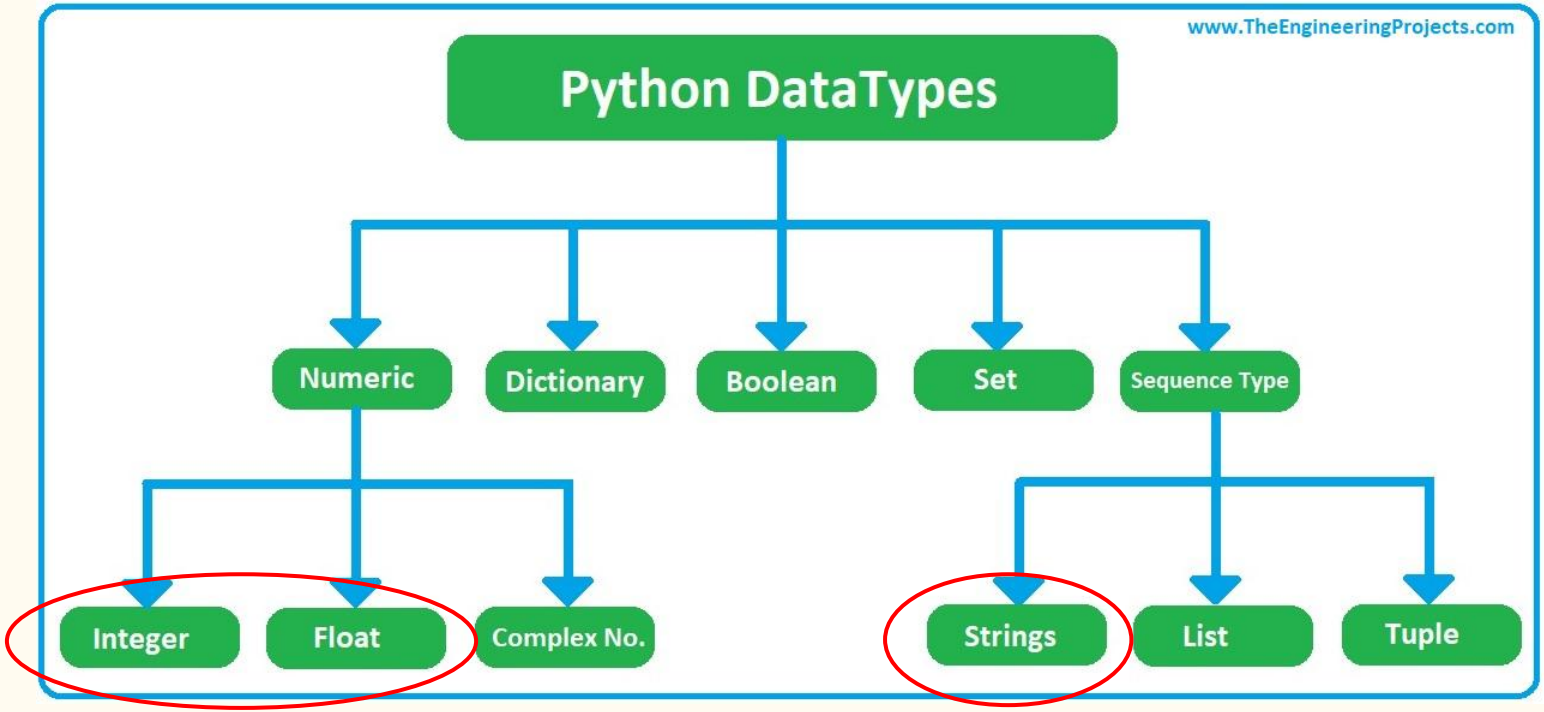
New Concepts

1. Data Types and the type() function.
2. Math calculations
3. Updating variables
4. Errors

Data Types


www.TheEngineeringProjects.com
Python DataTypes
Numeric
Dictionary
Boolean
Set
Sequence Type
Integer
Float
Complex No.
Strings
List
Tuple

What we'll cover today



Quick Maffs!
What’s an integer; What’s a float?

Data type	**Definition**	**Example**
String	Characters within “”	“Hello 123”
Integer	Whole number	29, 5, -74, 0
Float	Real number with decimal point.	5.5, 29.48, 1.3904, 0.00

Math calculations

Calculation	Python syntax	Definition + Example
Addition	+	Adding
Subtraction	-	Subtracting
Multiplication	*	Multiplying
Division	/	Divising
Floor division	//	How many times a number will fully go into another.
Modulo	%	The remainder.
Power	**	It's for the exponent.

```
#25 = 3*7 + 4

# 25/7 = 3 remainder 4
```

```
25//7
```

```
3
```

```
# 25/7 = 3 remainder 4
```

```
25%7
```

```
4
```

```
5**2
```

```
25
```

1+1 = 2?

Expression	Output	Explanation
1 + 1	2	This is math
“1” + “1”	“11”	This is string concatenation
1 + “1”	error	Python cannot handle addition of a string and number
“1” * 3 = “111”	“111”	String can be multiplied by * operator

type() function

Put the object in the type(), it returns the class of the object.

```
print(type("Brian"))
```

```
<class 'str'>
```

```
print(type(74))
```

```
<class 'int'>
```

```
print(type(2.3))
```

```
<class 'float'>
```

Data conversion

"74" is not the same as 74

Conversion Function	**str()**	**int()**	**float()**
Definition	Converts an argument to string type	Converts an argument to integer type	Converts an argument to float type

Example	`str(21)`
	`'21'`
	`y = 21`
	`print(type(y))`
	`y = str(21)`
	`print(type(y))`
	`<class 'int'>`
	`<class 'str'>`
		`int(2.73)`
		`2`
		`a = "21"`
		`print(type(a))`
		`a = int(a)`
		`print(type(a))`
		`<class 'str'>`
		`<class 'int'>`
			`float(21)`
			`21.0`
			`b = "21"`
			`print(type(b))`
			`b = float(b)`
			`print(type(b))`
			`print(b)`
			`<class 'str'>`
			`<class 'float'>`
			`21.0`

Updating Variables

Question:

Why do we want to update a variable? Why not just set it right the first time?

Answer:

Because data (which is stored in variables) often CHANGES!

Hence the name "variable"

EXAMPLE:

"We have 8 apples, we want number of apples to equal number of people, but if there are more than 8 fruits, we no longer need apples."

```
No_of_apples = 8
No_of_people = random.randint()
No_of_other_fruits = random.randint()

If No_of_people == No_of_apples:
    print("We have enough fruits")
If No_of_people != No_of_apples:
    print("We need more apples!")
If No_of_other_fruits >= No_of_apples:
    print("We have enough fruits")
```

Updating Variables

Consider this example:

The patient's diagnosis changes so we update the variable called diagnosis.

```
#One patient came to an emergency room and he diagnosed himself
diagnosis = "I have a pain in the lower right abdomen, I think I have cancer."
print("Patient said: ",diagnosis)
#Doctor opens to the possibility of typhlitis
diagnosis = "It could be typhlitis."
print("diagnosis from doctor: ",diagnosis)
```

```
Patient said:  I have a pain in the lower right abdomen, I think I have cancer.
diagnosis from doctor:  It could be typhlitis.
```

Notice how when we print the diagnosis again, it shows the updated value!

Updating Variables

Just some more examples ...

- The nurse might update the patient's hourly respiratory rate (RR variable) by entering the latest reading into the electronic medical record.
- We need to update a patient's medication list variable (covering the list data type in future workshops) of regular meds by adding or "appending" another medication (eg morphine).
- Or... a patient's length of stay in hospital increases by 1 day. We want to update this within a length_of_stay variable.
    - We want to **count** or **iterate** (in programming) their length_of_stay variable.

Updating Variables

Just some more examples ...

- Say we have a step counter, and for every step we want to add one to the counter.

- Therefore, we can use a variable called stepCount and update stepCount += 1 for every step that the user takes.

- stepCount = 0
  stepCount += 1

- Now stepCount == 1 will be True. If we encounter stepCount += 1 again, stepCount == 2 is True.

- To check total steps -> print(stepCount)

So how does iteration work?

Iteration is a particular way of updating variables. It helps you COUNT things 1 by 1.

Is defined as...

Plus 1.

$$x = x + 1$$

New value of x...

Old value of x ...

Old value of x

New, updated value of x

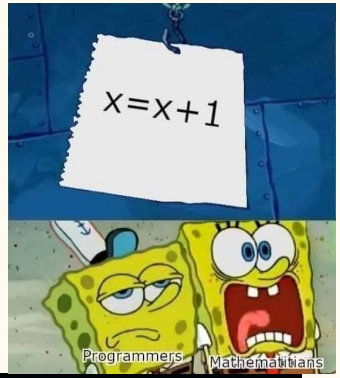
```
x = 0;
print(x)
x = x+1;
print(x)
```

```
0
1
```

x = x + 1

Mathematically, this is BOGUS.

Programatically...



“=” versus “==”

Assignment vs. comparison

Note: in relation to Comparison, Booleans will be covered later on

Other methods to update variables

x += 5 is shorthand for x = x + 5

x *= 5 is shorthand for x = x * 5

x /= 5 is shorthand for x = x / 5

```
x = 3
x = x + 5
print(x)
x = 3
x += 5
print(x)
```

```
8
8
```

Errors

Changing track, what exactly is an error?

- For all intents and purposes…
  - An error is a nothing more than a programming mishap that causes the code not to run as you expected.
  - Some synonyms:
    - Error = Bug = problem = “My code’s not working! :(” = “Why won’t it !@#$ing work!”

Error messages provide feedback to help us DIAGNOSE what’s gone wrong so we can correct our code.

Where are the 4 errors here?

```
symptom_1 = "fever"
symptom_2 = "cough"
symptom 3 = "runny nose"
symptom_4 = "sore throat

next_step = "covid swab"
print(next_Step
```

1.

2.

3.

4.

Syntax Error

- Very very common error!
  - Issue with programming “grammar”.
  - Whilst humans can communicate with less than perfect grammar, computers cannot ignore our mistakes!
  - It can be as simple a mistake as missing a bracket, forgetting a quotation mark or mispelling a variable.

Where is the error here?

```
weight = 75
height = 0
BMI = weight / (height**2)
print(BMI)
```

Arithmetic Error

- Example: ZeroDivisionError

```
weight = 75
height = 0
BMI = weight / (height**2)
print(BMI)
```

```
---------------------------------------------------------------------------
ZeroDivisionError                         Traceback (most recent call last)
<ipython-input-72-300f51e27a9b> in <module>()
      1 weight = 75
      2 height = 0
----> 3 BMI = weight / (height**2)
      4 print(BMI)

ZeroDivisionError: division by zero
```

What do I do about errors?

- Thankfully, most coding software (such as jupyterLab) has built-in features to minimise errors and write neat code!
  - Error messages
  - Squiggly underlining
  - Colour coding


```
print("coronavirus"

  File "<ipython-input-65-c59f9162fef6>", line 1
    print("coronavirus"
                       ^
SyntaxError: unexpected EOF while parsing
```

“All you need to know for programming is how to Google things.”

mini_Project: Lindsay Brown

mini_Project: Lindsay Brown (Coding a History)

mini_Project brief:

- Using data from a PCP1 role-play, we will create a function that uses basic Natural Language Processing to:
  - Extract the relevant history (as our input)!
  - Output a summary for the Consultant!

In future workshops, we'll hopefully build these mini-Projects together into a larger, integrated EMR assistant application!

Breakout Time!

FEEDBACK FORM:

https://forms.gle/6ntfWe9RL49z6xG49