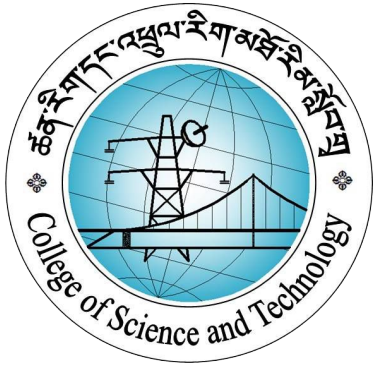


Royal University of Bhutan

Unit II:
Understanding Basic Data Types & Packages

Programming Methodology (CSF101)

Outline

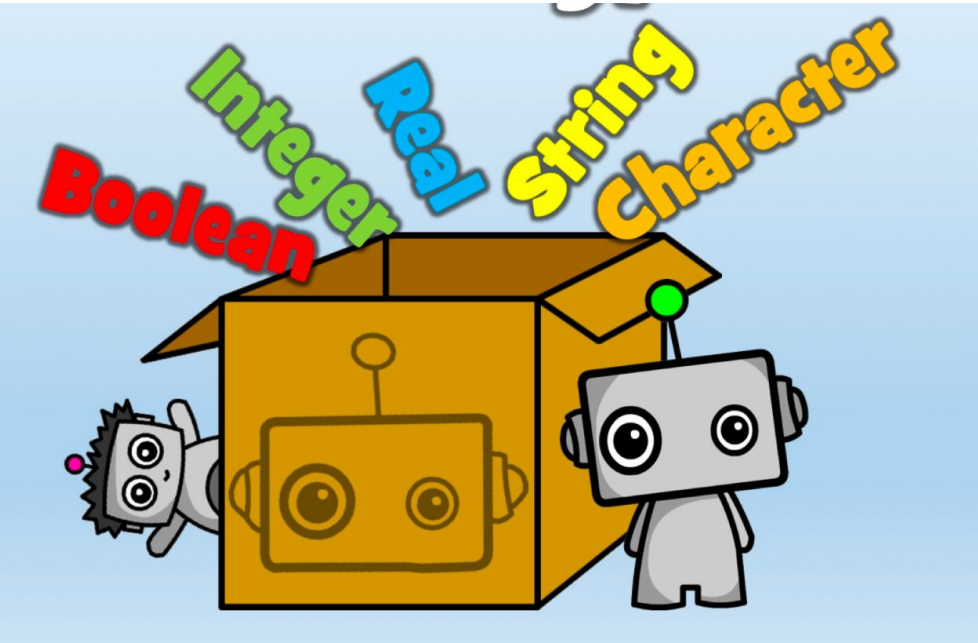
- Language Data Types & Abstract Data Types
- String Manipulation

Data types

- For an instance, you have packed your things in a following manner to move out the new resident

Cont…

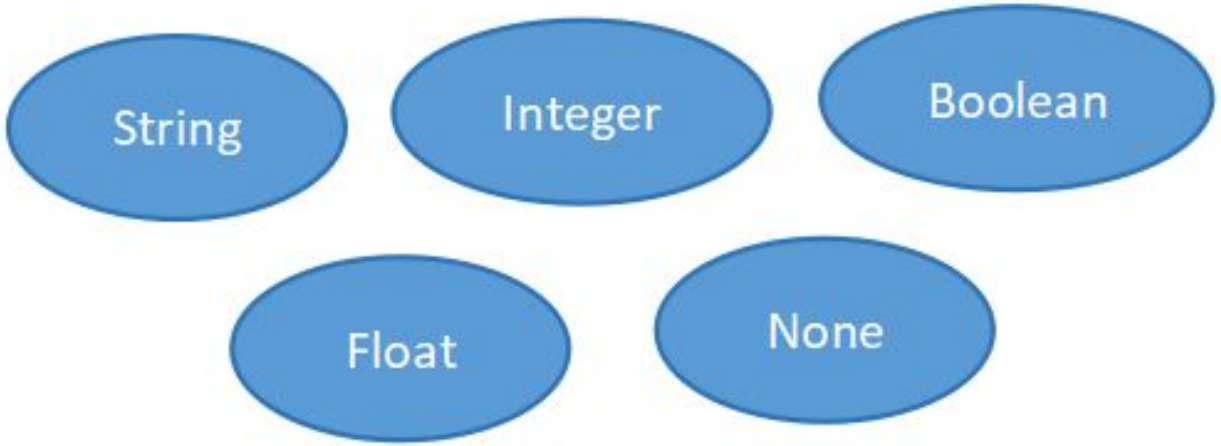
- Labels in computer variable is Data types



Types of Data Types

Primitive data type

- Basic inbuilt data type
- Represents a single data
- Used for simple tasks



String

- Text values are called strings

```
print("Hello")
print('Hello')
```

```
>>> a = 1
>>> b = 1
>>> c = a + b
>>> c = str(c)
>>> print(type(c))
<class 'str'>
```

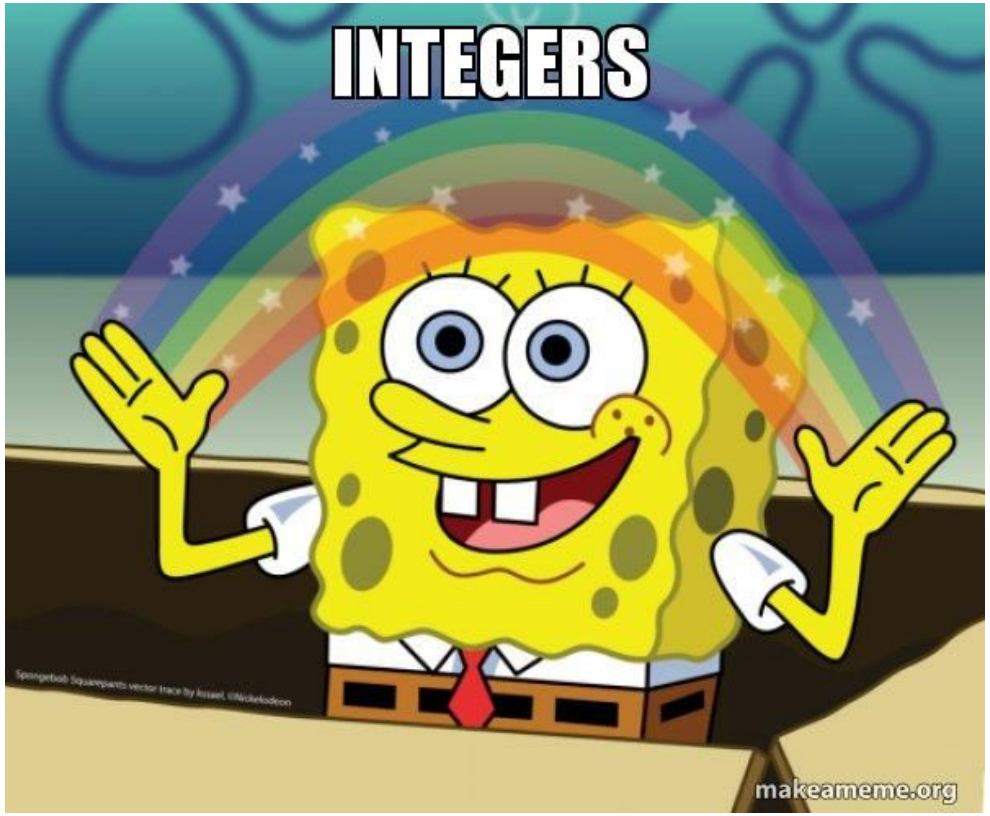
```
a = """Lorem ipsum dolor sit amet,
consectetur adipiscing elit,
sed do eiusmod tempor incididunt
ut labore et dolore magna aliqua."""
```

```
a = '''Lorem ipsum dolor sit amet,
consectetur adipiscing elit,
sed do eiusmod tempor incididunt
ut labore et dolore magna aliqua.'''
```

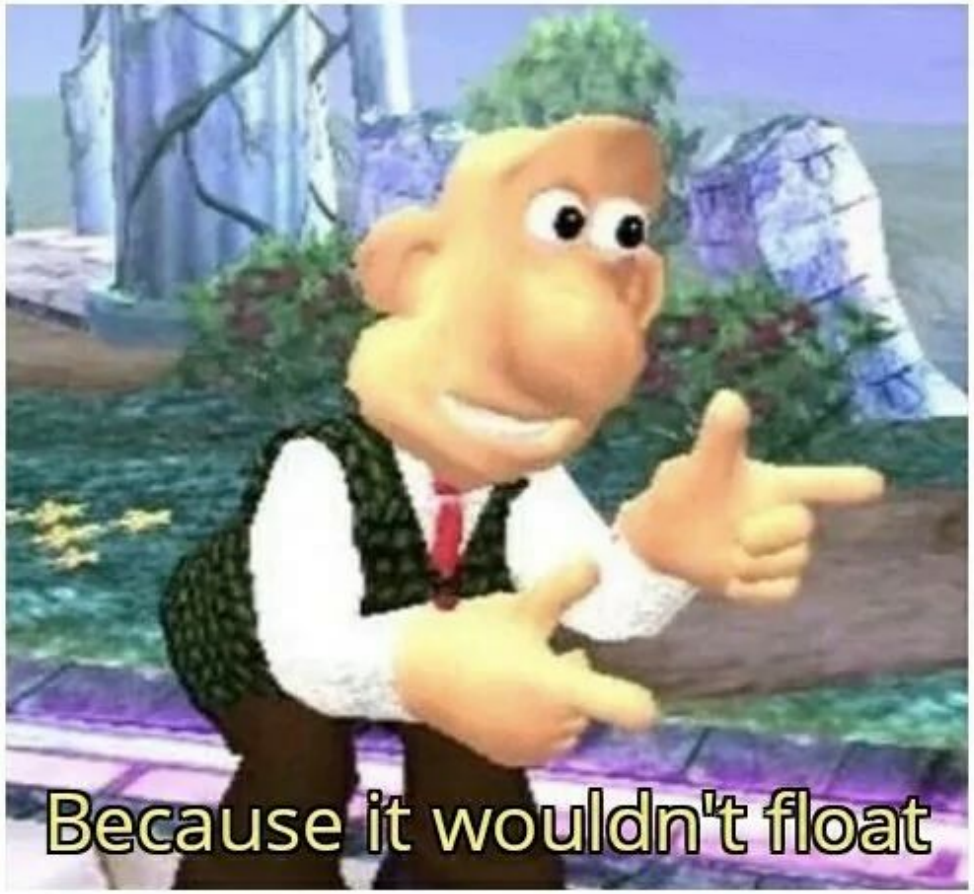
Integer

```
x = 1
y = 35656222554887711
z = -3255522
```

```
>>> pi = 3.14
>>> pi = int(pi)
>>> print(pi)
3
```



why did the integer drown?



Float

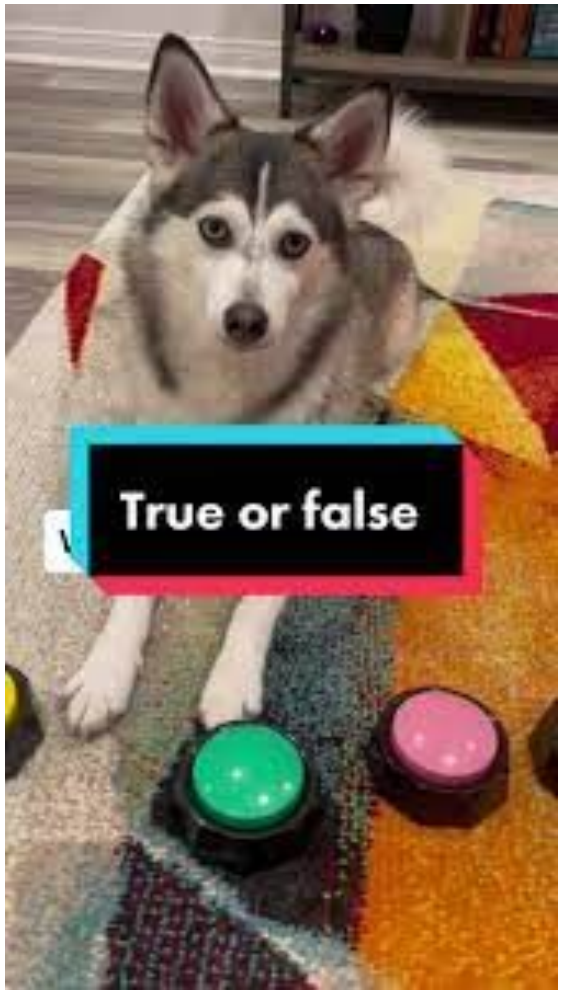
```
x = 1.10
y = 1.0
z = -35.59
```

```
>>> pi = "3.14"
>>> print(type(pi))
    <class 'str'>
>>> pi = float(pi)
>>> print(type(pi))
    <class 'float'>
```

Boolean

```
>>> print(0<1)
True
>>> print(0>1)
False
```

```
>>> bool(0)
False
>>> bool(1)
True
>>> bool("Hello")
True
>>> bool()
False
```



None

```
>>> x = None
>>> print(x)
None
```

```
>>> x = None
>>> print(type(x))
<class 'NoneType'>
```

Empty Parking Lot

Class Activity

Code a python program following the instructions given below.

The code begins by prompting the user to enter their name, storing it in the variable name.

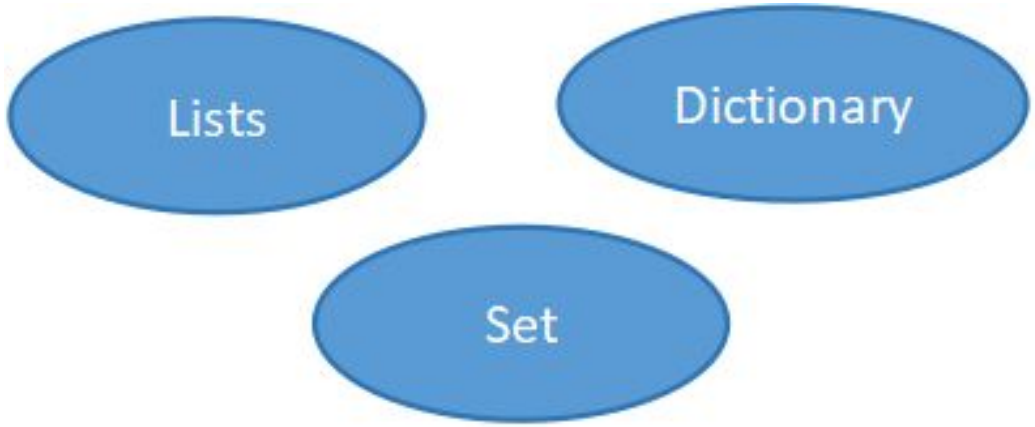
It then prompts the user to enter two numbers (num1 and num2) and stores them as floating-point numbers.

The entered numbers (num1 and num2) are used to perform basic arithmetic operations: addition, subtraction, multiplication, and division. And the results of these operations are stored in separate variables (addition_result, subtraction_result, etc.).

The program prints a personalized greeting along with the results using the print() function. Ensure that the print statements are clear and informative, providing a user-friendly output.

Abstract Data Type

- Data types that are derived from primitive data types
- Used to represent group of data



Lists

```
['hello', 3.1415, True, None, 42]
```

```
spam = ["cat", "bat", "rat", "elephant"]
       spam[0]  spam[1]  spam[2]  spam[3]
```

```
>>> spam[1:3]
['bat', 'rat']
```

```
>>> supplies = ['pens', 'staplers', 'flamethrowers', 'binders']
>>> for index, item in enumerate(supplies):
...     print('Index ' + str(index) + ' in supplies is: ' + item)
```

Operations in Lists

```
>>> spam = ['cat', 'dog', 'bat']
>>> spam.index('cat')
0
```

```
>>> spam = ['cat', 'dog', 'bat']

>>> spam.append('moose')

>>> spam

['cat', 'dog', 'bat', 'moose']
```

```
>>> spam.insert(1,1)
>>> spam
['cat', 1, 'dog', 'bat']
```

```
>>> spam = ['cat', 'dog', 'moose']

>>> spam.reverse()

>>> spam

['moose', 'dog', 'cat']
```

```
>>> spam.remove('cat')
>>> spam
[1, 'dog', 'bat']
```

```
>>> spam = ['ants', 'cats', 'dogs', 'badgers', 'elephants'

>>> spam.sort()

>>> spam

['ants', 'badgers', 'cats', 'dogs', 'elephants']
```

Set

```
>>> spam = {'cat', 'dog', 1, 'monkey'}
>>> spam
{1, 'cat', 'dog', 'monkey'}
```

Operations in set

```
>>> spam.add(2)
>>> spam
{1, 2, 'cat', 'monkey', 'dog'}
```

```
>>> spam.remove(1)
>>> spam
{2, 'cat', 'monkey', 'dog'}
```

```
>>> spam
{2, 'cat', 'monkey', 'dog'}
>>> spam1 = {2, True}
>>> unionSpam = spam.union(spam1)
>>> unionSpam
{True, 2, 'cat', 'monkey', 'dog'}
```

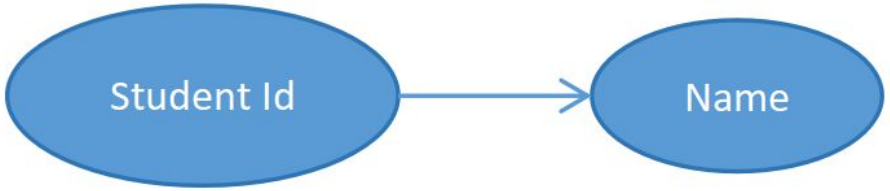
```
>>> intersecSpam = spam.intersection(spam1)
>>> intersecSpam
{2}
```

```
>>> differSpam = spam.difference(spam1)
>>> differSpam
{'cat', 'dog', 'monkey'}
```

Dictionary

```
>>> myCat = {'size': 'fat', 'color': 'gray', 'disposition': 'loud'}
```

```
>>> spam = ['cats', 'dogs', 'moose']
>>> bacon = ['dogs', 'moose', 'cats']
>>> spam == bacon
False
>>> eggs = {'name': 'Zophie', 'species': 'cat', 'age': '8'}
>>> ham = {'species': 'cat', 'age': '8', 'name': 'Zophie'}
>>> eggs == ham
True
```



Operations in Dictionary

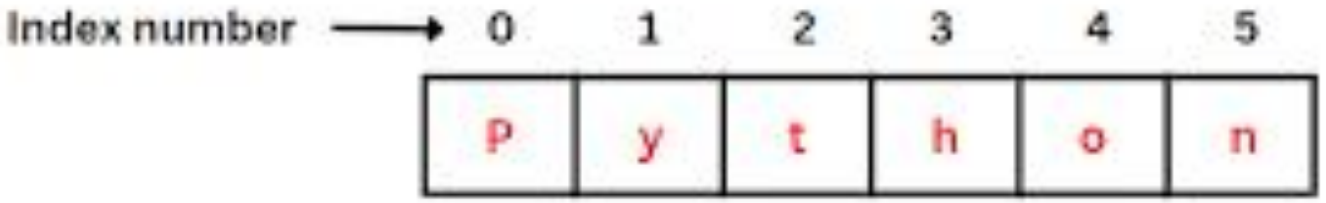
```
>>> spam = {'name' : 'Alice', 'age' : 20}
>>> spam.keys()
dict_keys(['name', 'age'])
>>> spam.values()
dict_values(['Alice', 20])
```

```
>>> spam.items()
dict_items([('name', 'Alice'), ('age', 20)])
```

```
>>> spam.get('name')
'Alice'
```

```
>>> spam.setdefault('height', 180)
180
>>> spam.items()
dict_items([('name', 'Alice'), ('age', 20), ('height', 180)])
```

String Manipulation



With Strings !

len()

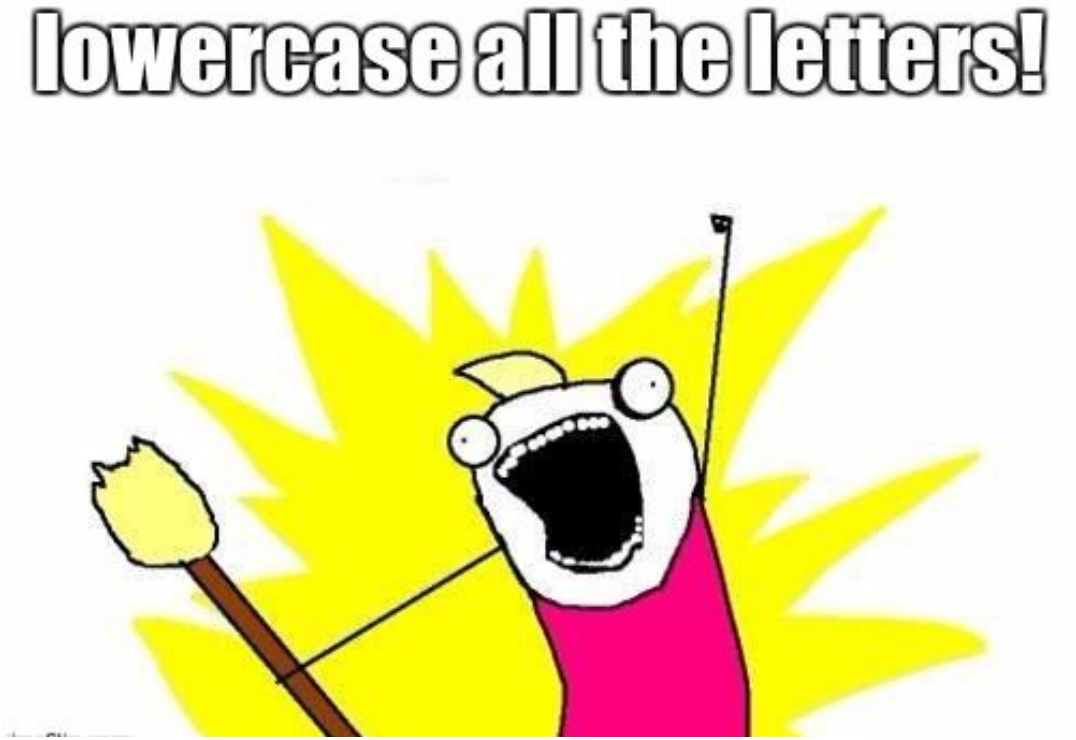
```
>>> spam = 'Hello, world!'
>>> print(len(spam))
13
```

upper()

```
>>> spam = 'Hello, world!'
>>> spam = spam.upper()
>>> spam
'HELLO, WORLD!'
```

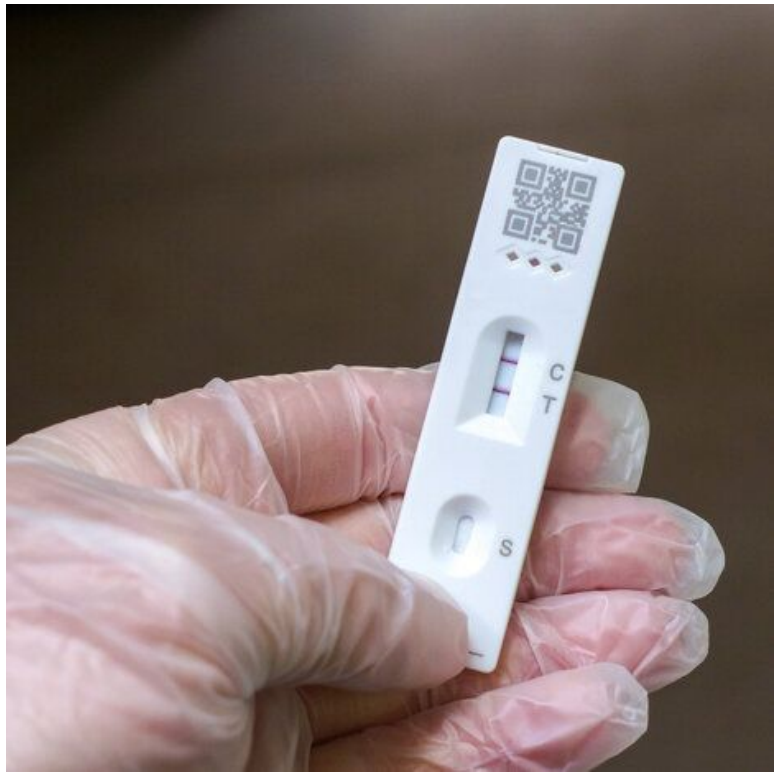
lower()

```
>>> spam = spam.lower()

>>> spam

'hello, world!'
```



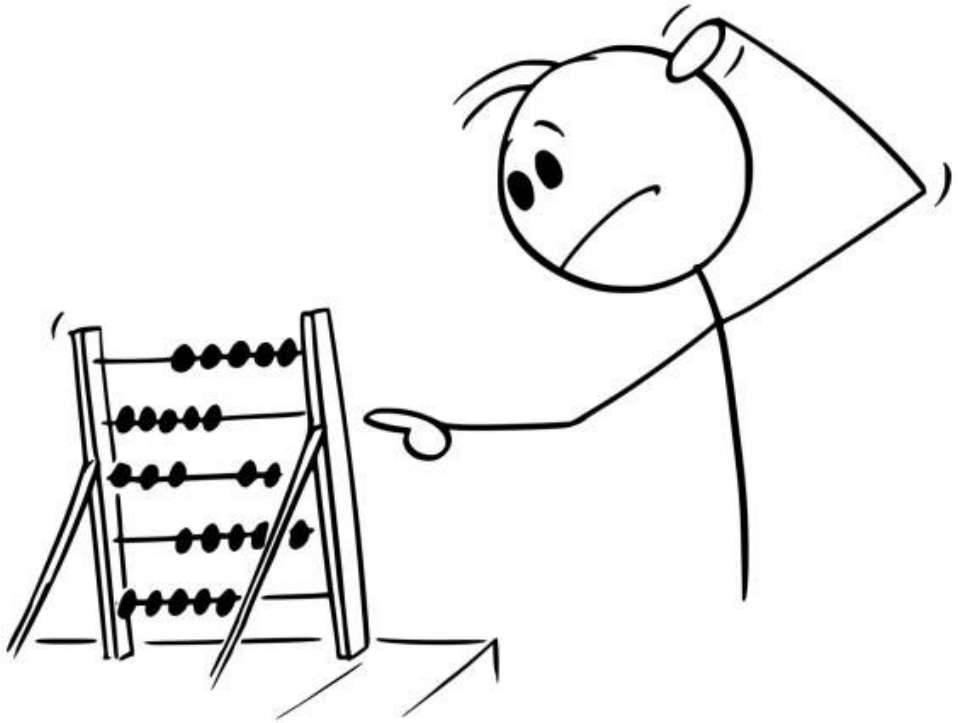
isX()

```
>>> 'HELLO'.isupper()

True

>>> 'abc12345'.islower()

True
```



count()

```
>>> a = "Hello"
>>> print(a.count('l'))
2
>>> print(a.count('H'))
1
```



find()

```
>>> a = "Hello"
...
>>> print(a.find('e'))
...
1
>>>
```

strip()

```
>>> a = "    ,  Hello  "
>>> print(a)
    ,  Hello
>>> print(a.strip())
,  Hello
>>> a = a.strip(" ,")
>>> print(a)
Hello
```

Split()

```
>>> spam = "Hello, Welcome to whereever"
>>> print(spam)
Hello, Welcome to whereever
>>> spam.split()
['Hello,', 'Welcome', 'to', 'whereever']
>>> spam.split(",")
['Hello', ' Welcome to whereever']
```

Escape Characters

Escape character	Prints as
\'	Single quote
\"	Double quote
\t	Tab
\n	Newline (line break)
\\	Backslash

```
>>> print("Hello there!\nHow are you?\nI\'m doing fine.")
Hello there!
How are you?
I'm doing fine.
```

```
>>> print(r'That is Carol\'s cat.')
That is Carol\'s cat.
```

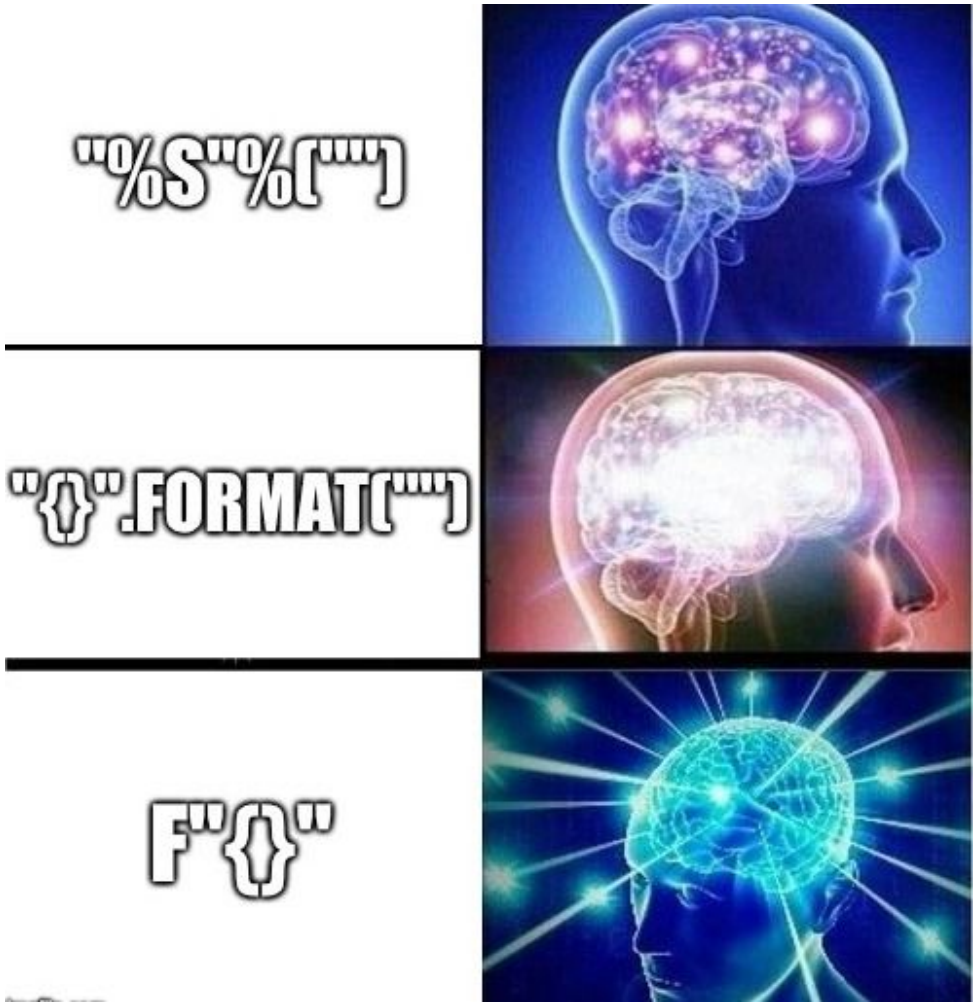
Formatted String

```
>>> name = "Karma"
>>> print(f"Hello {name}")
Hello Karma
```

```
>>> print("Greeting to you, {}".format(name))
Greeting to you, Karma
```

```
>>> name = "Karma"
>>> Number = 2
>>> print("There are %d %s in the class" %(Number, name))
There are 2 Karma in the class
```

Cont…



Concatenation

```
a = "Hello"
b = "World"
c = a + " " + b + "!"
print(c)

#Output: Hello World!
```

```
print("Hello" + 2)
#Output: TypeError: can only concatenate str (not "int") to str
```

```
>>> ', '.join(['cats', 'rats', 'bats'])

'cats, rats, bats'

>>> ' '.join(['My', 'name', 'is', 'Simon'])

'My name is Simon'
```

String Replication

```
>>> 'Alice' * 5

'AliceAliceAliceAliceAlice'
```

4X

Cont…

```
>>> 'Python goes b' + 'r'*10
'Python goes brrrrrrrrrr'
```

Comments

```
print("This line will be executed!")
#This is commented line
```

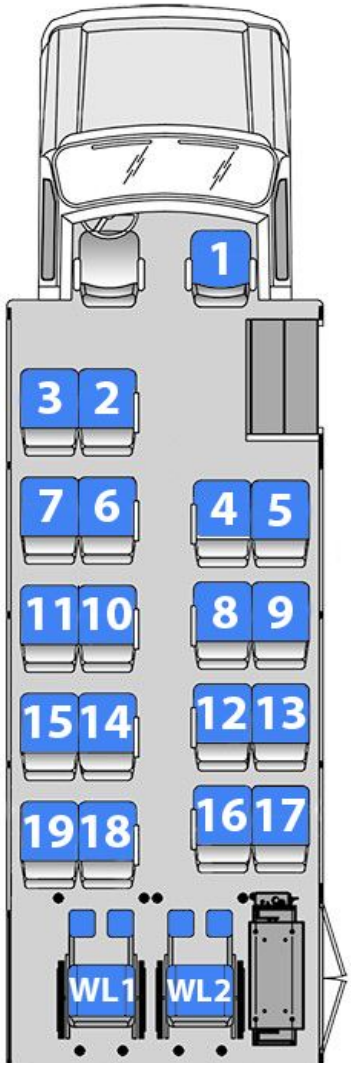
```
/comment.py
This line will be executed!
>>>
```

```
"""

This  code is under comment.

"""
```

Indexing

'	H	e	l	l	o	,		w	o	r	l	d	!	'
	0	1	2	3	4	5	6	7	8	9	10	11	12	

```
>>> spam[4]
'o'
>>> spam[-1]
'!'
```



Slicing

'	H	e	l	l	o	,		w	o	r	l	d	!	'
	0	1	2	3	4	5	6	7	8	9	10	11	12	

```
>>> spam[:5]
'Hello'
>>> spam[7:]
'world!'
```

Class Acitivity

Go to https://www.slido.com/.

Then enter the code: csf101

Reference

Automate the boring stuff with python. Automate the Boring Stuff with Python book cover thumbnail. (n.d.). https://automatetheboringstuff.com/

Python tutorial. (n.d.). https://www.w3schools.com/python/


Thanks