Introduction to Python

```
In [7]: # Example 1
        print('Welcome to AI class')
```

```
Welcome to AI class
```

Python Comments

- Comments can be used to explain Python code.
- Comments can be used to make the code more readable.
- Comments can be used to prevent execution when testing code.

Creating a Comment

Comments starts with a #, and Python will ignore them:

```
In [4]: #Example of Comments
        print("Hello, World!")
```

```
Hello, World!
```

```
In [5]: print("Hello, World!") #This is a comment
```

```
Hello, World!
```

Multi Line Comments

```
In [6]: """
        This is a comment
        written in
        more than just one line
        """
        print("Hello, World!")
```

```
Hello, World!
```

What is a Variable in Python?

A Python variable is a reserved memory location to store values. A variable is created the moment you first assign a value to it.

```
In [16]: x=5 # x is type of int
         x='Artificial intelligence' # y is type of str
         print('The value of x is :',x)
         print('The value of y is :',y)
```

```
The value of x is : 5
The value of y is : Artificial intelligence
```

Casting

If you want to specify the data type of a variable, this can be done with casting.

```
In [21]: x = str(3)    # x will be '3'
         y = int(3)    # y will be 3
         z = float(3)  #z will be 3.0
         print('The value of x: {}, y: {}, and z: {}'.format(x,y,z))

         #Another Method of printing multi variable in single line is
         print('The value of x:',x,'y:',y ,'and z:',z)
```

```
The value of x: 3, y: 3, and z: 3.0
The value of x: 3 y: 3 and z: 3.0
```

Get the Type

You can get the data type of a variable with the type() function.

```
In [22]: x = 5
         y = "AI"
         print(type(x))
         print(type(y))

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

Variable Names

- A variable can have a short name (like x and y) or a more descriptive name (age, carname, total_volume). Rules for Python variables:
- A variable name must start with a letter or the underscore character
- A variable name cannot start with a number
- A variable name can only contain alpha-numeric characters and underscores (A-z, 0-9, and _)
- Variable names are case-sensitive (age, Age and AGE are three different variables)

```
In [23]: myvar='AI'
         my_var='AI'
         _my_var='AI'
         MYVAR='AI'
```

```
In [24]: #Illegal variable names:
         2myvar = "AI"
         my-var = "AI"
         my var = "AI"
```

```
  File "<ipython-input-24-d59a5bab4a43>", line 2
    2myvar = "AI"
    ^
SyntaxError: invalid syntax
```

Assign Multiple Values

```
In [33]: x, y, z = "Data science", "Machine learning", "Deep learning"

         print("Welcome to",x, end = ' ')
         print(",",y, end = ' ')
         print("and ", z, end = ' ')
```

```
Welcome to Data science, Machine learning and Deep learning
```

```
In [35]: #One Value to Multiple Variables
         x=y=z='AI'
         x,y,z
```

```
Out[35]: ('AI', 'AI', 'AI')
```

Taking input in Python

```
In [49]: val= input('Enter your marks: ')

         Enter your marks: 5
```

```
In [50]: print('Your marks are :',val)

         Your marks are : 5
```

```
In [54]: num = int(input ("Enter number :"))
         print(num)
         names1 = str(input("Enter name : "))
         print(names1)

         Enter number :10
         10
         Enter name : sikandar
         sikandar
```

```
In [68]: #Taking multiple input fromthe user
         x, y = input("Enter two values: ").split(',')
         print('The first value is {} and seconnd value is {}'.format(x,y))

         Enter two values: 7,5
         The first value is 7 and seconnd value is 5
```

Python Data Types

Variables can store data of different types, and different types can do different things.
Python has the following data types built-in by default

Text Type: **str**

Numeric Types: **int, float, complex**

Sequence Types: **list, tuple, range**

Mapping Type: **dict**

Set Types: **set, frozenset**

Boolean Type: **bool**

Python Numbers

There are three numeric types in Python:

- int
- float
- complex

Variables of numeric types are created when you assign a value to them:

```
In [37]: #int
         x=10
         y=-100
         z=111111221215613

         print(type(x))

         <class 'int'>
```

```
In [42]: # float
         #x=3.14
         y = 124.4 # E indicate the power of 10.
         z = -87.7e100
         print(type(z))
         print(z)

         <class 'float'>
         -8.77e+102
```

```
In [43]: ''' Complex
         Complex numbers are written with a "j" as the imaginary part:'''
         x = 3+5j
         y = 5j
         z = -5j
         print(type(x))

         <class 'complex'>
```

Type Conversion

```
In [47]: x=10
         y=12
         z=12j

         #convert from int to float:
         a = float(x)

         #convert from float to int:
         b = int(y)

         #convert from int to complex:
         c = complex(x)

         print(a)
         print(b)
         print(c)

         print(type(a))
         print(type(b))
         print(type(c))

         10.0
         12
         (10+0j)
         <class 'float'>
         <class 'int'>
         <class 'complex'>
```

Python Strings

Strings in python are surrounded by either single quotation marks, or double quotation marks. 'Al' is the same as "Al".

```
In [1]: print("Hello")
        print('Hello')

        Hello
        Hello
```

```
In [9]: # Assign String to a Variable
        course_Name='Artificial intelligence'
        print(course_Name)

        Artificial intelligence
```

Strings are Arrays

An array is a data structure that stores values of same data type

```
In [12]: # Get the character at position 1 (remember that the first character has the position 0):
         print(course_Name[0])
         print(course_Name[-1])

         A
         e
```

Slicing Strings

Specify the start index and the end index, separated by a colon, to return a part of the string.

```
In [17]: print(course_Name[0:10])
         print(course_Name[10:])

         Artificial
          intelligence
```

```
In [18]: print(course_Name[:10])

         Artificial
```

```
In [30]: print(course_Name[-12:])

         intelligence
```

```
In [31]: print(course_Name[-5:-2])

         gen
```

Modify Strings

Python has a set of built-in methods that you can use on strings.

```
In [35]: #Upper Case: The upper() method returns the string in upper case:
         upper_Case=course_Name.upper()

         # print(course_Name.upper())
         print(upper_Case)

         ARTIFICIAL INTELLIGENCE
```

```
In [36]: lower_case=upper_Case.lower()
         print(lower_case)

         artificial intelligence
```

```
In [40]: #Remove Whitespace
         course='Machine learning '
         print(course)
         print(course.strip())

          Machine learning
         Machine learning
```

```
In [42]: #Replace string
         re_Course=course.replace(' ','')
         print(re_Course)

         Machinelearning
```

```
In [50]: Course='Machine learning / Deeplearning'
         re_Course=Course.replace('/','&')
         print(Course)
         print(re_Course)

         Machine learning / Deeplearning
         Machine Learning & Deeplearning
```

```
In [57]: #Split string
         f_Course,l_Course = Course.split('/')
         print(f_Course)
         print(l_Course)

         Machine Learning
          Deeplearning
```

```
In [59]: #String Concatenation
         full_Course=f_Course+l_Course
         print(full_Course)

         Machine Learning  Deeplearning
```

```
In [63]: full_Course=f_Course+ '&' +l_Course
         print(full_Course)

         Machine Learning & Deeplearning
```

```
In [64]: #Assignments practice all the String Methods
         # https://www.programiz.com/python-programming/methods/string
```

Booleans

Booleans represent one of two values: True or False.

```
In [66]: print(10 > 9)
         print(10 == 9)
         print(10 < 9)

         True
         False
         False
```

```
In [69]: print(bool("Hello"))
         print(bool(0))
         print(bool(-1))

         True
         False
         True
```

```
In [77]: x='True'
         x=bool(x)
         y=bool(0)
         z=bool(z)
         Z
```

```
Out[77]: False
```

```
In [ ]:
```