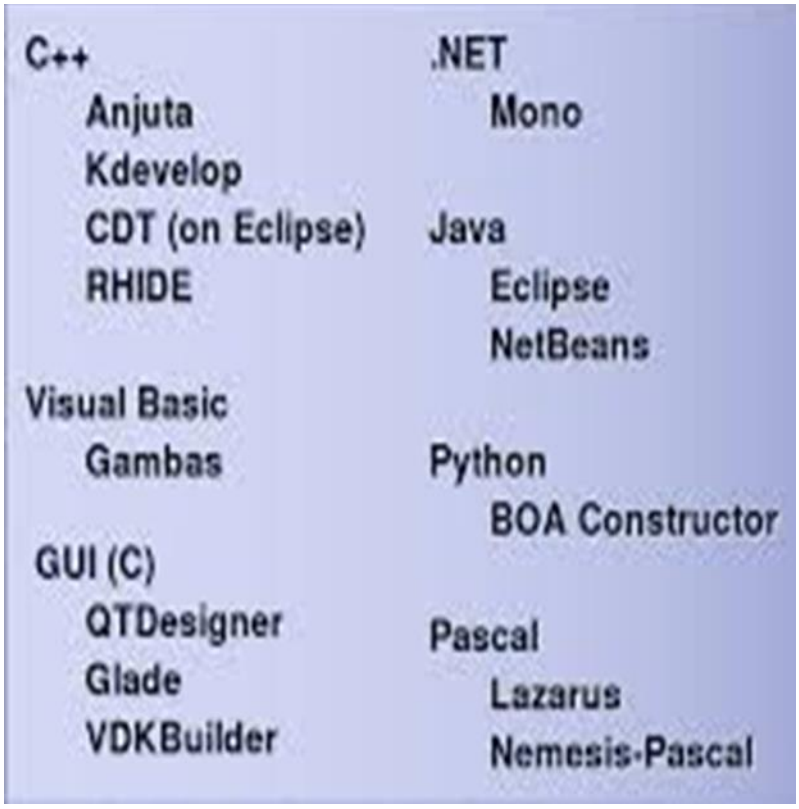
Explores integrated development environment (IDE) to identify their basic features

- Basic Features of IDE
- Instructions to Use
    - Opening and Saving Files
    - Compiling, Executing Programs
- Debugging Facilities

Basic Features of IDE

What is an Integrated Development Environment (IDE)?

- An Integrated Development Environment (IDE) is an application that facilitates application development.
- In general, an IDE is a graphical user interface (GUI)-based workbench designed to aid a developer in building software applications with an integrated environment combined with all the required tools at hand.



Types of Integrated Development Environment (IDE)

- Multi-Language IDEs. Multi-language IDEs, such as Eclipse, NetBeans, Komodo, Aptana and Geany, support multiple programming languages
- IDEs for Mobile Development
- HTML IDEs
- Cloud-Based IDEs
- IDEs Specific to Microsoft or Apple
- IDEs for Specific Languages.

Examples of Integrated Development Environment (IDE)

- **Microsoft Visual Studio –** 
    - premium IDE ranging in price from$699 - $2,900 depending on the edition and licensing
    - Languages Supported: ASP.NET, DHTML, JavaScript, JScript, Visual Basic, Visual C#, Visual C++, Visual F#, XAML and more

- **NetBeans -** 
    - free and open-source IDE
    - **Languages Supported:** C, C++, C++11, Fortan, HTML 5, Java, PHP and more

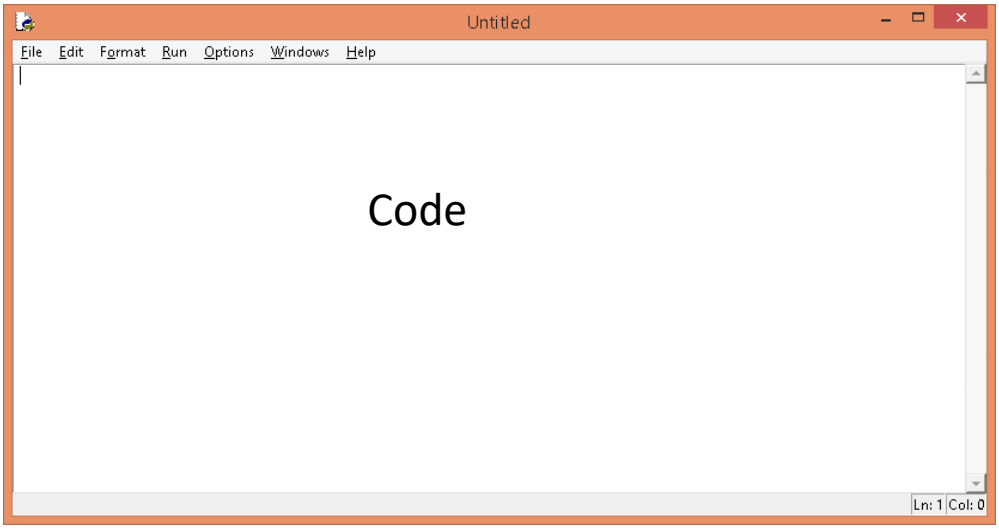
Common Features of Integrated Development Environment (IDE)

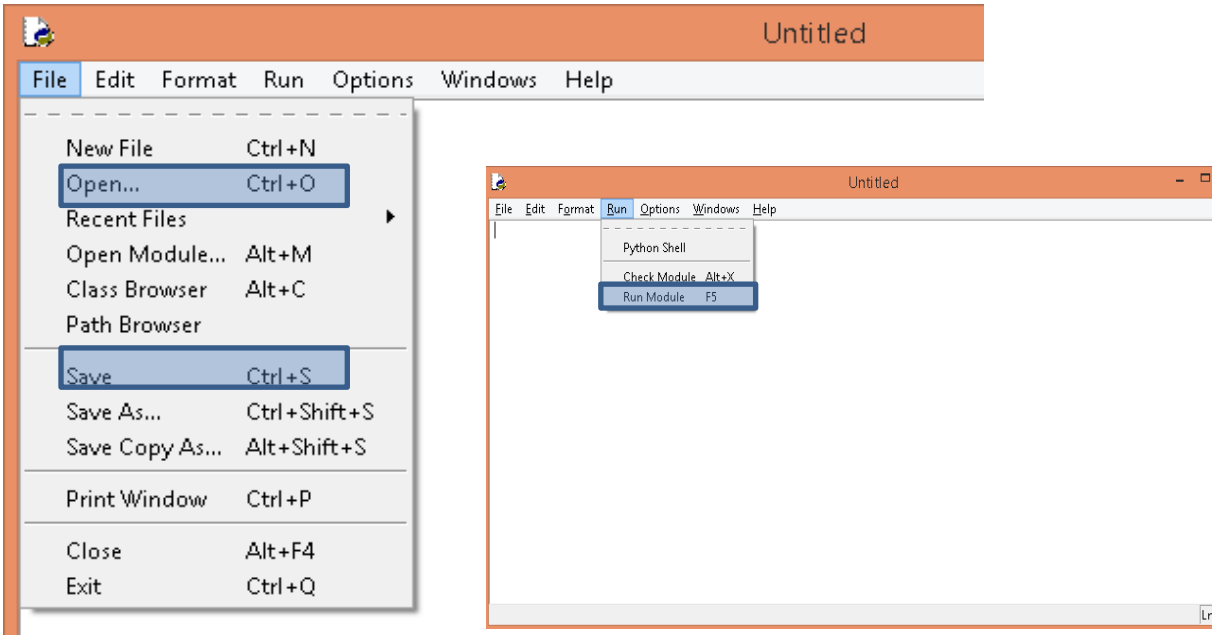
- An IDE typically contains a code editor, a compiler or interpreter, and a debugger, accessed through a single graphical user interface (GUI).
- The user writes and edits source code in the code editor.
- The compiler translates the source code into a readable language that is executable for a computer.
- And the debugger tests the software to solve any issues or bugs.
- An IDE can also contain features such as programmable editors, object and data modeling, unit testing, a source code library and build automation tools.

Benefits of Using Integrated Development Environment (IDE)

- Most common features, such as debugging, version control and data structure browsing, help a developer quickly execute actions without switching to other applications. Thus, it helps maximize productivity by providing similar user interfaces (UI) for related components
- Without an IDE, developers spend time deciding what tools to use for various tasks, configuring the tools and learning how to use them.
- Reduces the time taken to learn the language
- IDEs are also designed with all their tools under one user interface.
- An IDE supports single or multiple languages.

IDE for Python





What is Debugging?

Debugging, in computer programming and engineering, is a multistep process that involves **identifying a problem, isolating the source of the problem, and then either correcting the problem or determining a way to work around it.** The final step of debugging is to test the correction or workaround and make sure it works.

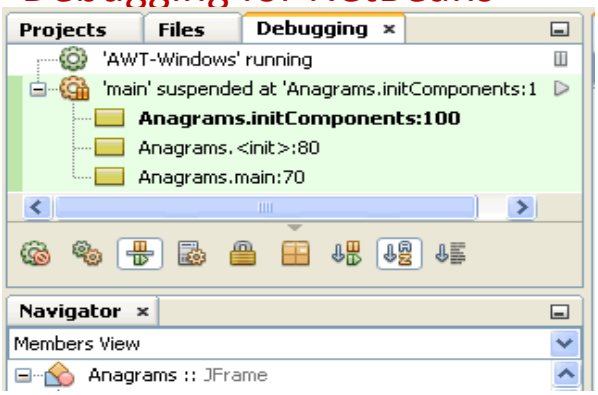
The standard practice is to set up a "breakpoint" and run the program until that breakpoint, at which time program execution stops. The debugging

component of an IDE typically provides the programmer with the capability to view memory and see variables, run the program to the next breakpoint, execute just the next line of code, and, in some cases, change the value of variables or even change the contents of the line of code about to be executed.

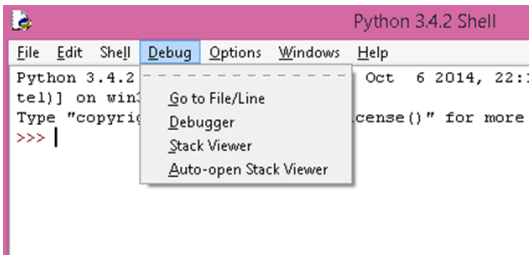
Bugs = Errors

Debugging = Troubleshooting

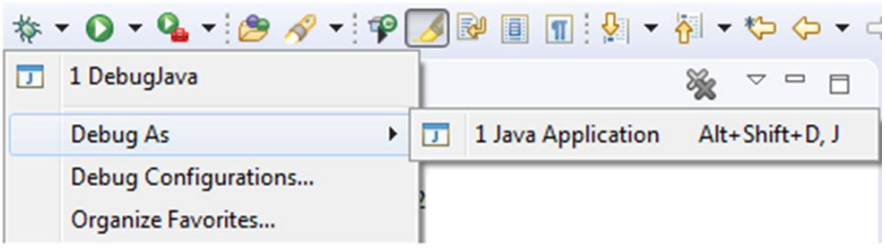
Debugging for NetBeans



Debugging for Python IDE



Debugging the Eclipse IDE for Java



Uses an imperative programming language to encode algorithms

- **Structure of a Program**
- **Comments**
- **Constants and Variables**
- **Primitive Data Types**

What is a structure of a program?

- When we create a computer program, there is a style to insert coding into the program.
- This style is different from one computer programming language to another.
- The way we develop a program with this style called structure of a program.
- In this lesson we will discuss about the structure of python programming language.

Comments

- Comment is a text in a computer program that is a programmer-readable explanation or annotation in the source code.
- It is ignored by compiler/interpreter.

How do we provide comments in Python?

- **Single line comment**
  - In Python script, the **symbol #** indicates start of comment line.
  - It is effective till the end of line in the editor.
  - If # is the first character of line, then entire line is a comment.
  - It can be used in the middle of a line.

- **Example for a single line comment**

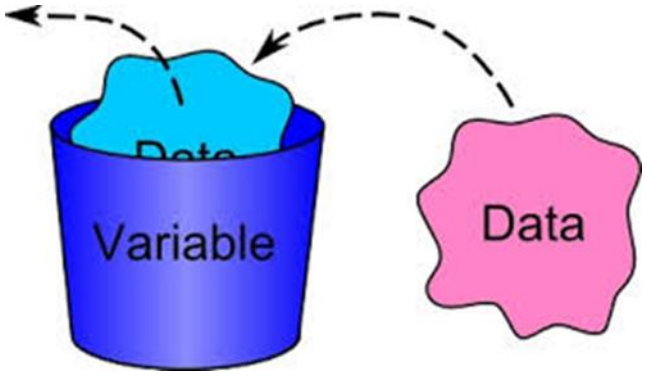
```
#this is a comment
print ("Hello World")
print ("Welcome to Python Programming")#this is also a comment but after a statement.
```

- **Multi – line comment or a block comment in python?**
  - Many Python IDEs have shortcuts to mark a block of statements as comment.
  - A triple quoted multi-line string is also treated as comment.
  - **Example for a Multi – line comment or a block comment**

```
'''
comment1
comment2
comment3
'''
print ("Hello World")
```

Constants and Variables

- A program is a sequence of instructions, which process on input data and produces the desired output.
- During the running of a computer program, there will be times when the program needs to remember/store a value so it can be read and used later on.
- In order to store a value, the program needs to establish a memorable (named) place (data structure) that it can use to hold the data. These locations are called variables and constants.



- A program can contain many variables and constants, so it is important to give them sensible names that try to describe the item of data that they hold.
- **Constant, variables and data types** are the **basic building blocks** of any computer program.
- So, it is **very essential** to know about these terms and this is from where we start programming.

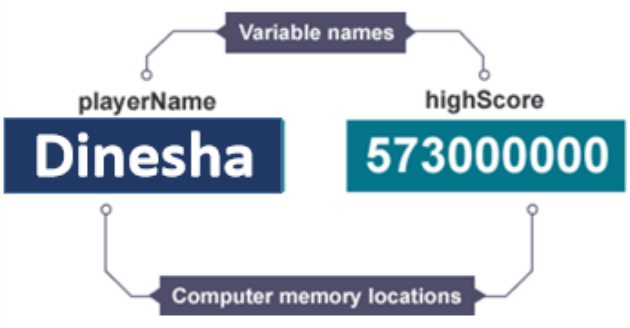
Variables

- In most of the programming languages a **variable is a named location used to store data in the memory**.

- Each variable **must have a unique name called identifier**. It is helpful to think of variables as **container that hold data which can be changed later throughout programming.**

- Non technically, you can suppose variable as a bag to store books in it and those books can be replaced at any time.

- Note: In Python we don't assign values to the variables, whereas Python gives the reference of the object (value) to the variable.



Declaring Variables in Python

- In Python, variables do not need declaration to reserve memory space.
- The "variable declaration" or "variable initialization" happens automatically when we assign a value to a variable.

Ex 01: Assigning Value to Variables in Python

- You can use the **assignment operator (=)** to assign the variable to a variable

```
school="WP/GM/Bandaranayake Vidyalaya"
print(school)

>>>
WP/GM/Bandaranayake Vidyalaya
>>>
```

Declaring Variables in Java

```
int score;

    score = 0;
```

Declaring Variables in C

```
int data;

data = value*value;
```

Ex 02 : Changing value of a variable

```
school="WP/GM/Bandaranayake Vidyalaya"
school="WP/GM/Rathnawali Balika Vidyalaya"
print(school)
```

Ex 03 : Assigning multiple values to multiple variables

```
a, b, c = 5, 3.2, "Hello"

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

Ex 04 : Assign the same value to multiple variables at once

```
x = y = z = "same"

print (x)
print (y)
print (z)
```

Constants

- Constants are fixed values that do not change during the execution of a program.
- It is helpful to think of constants as containers that hold information which cannot be changed later.
- Non technically, you can think of constant as a bag to store some books and those books cannot be replaced once placed inside the bag.

Python supports two types of constants that are listed below:

1. Numeric constants: These are numbers, may it be **positive number, negative number, complex number, floating point number or an exponential number**. Thus, in Python, following numbers are valid numeric constants: 127, -128, 98.67, 4 + 3i, etc.

2. String constants: These are the sequence of alphanumeric characters, special characters and underscores enclosed **in a single or double quote**. This makes following valid string constants: "Your Own Linux!", "myString", 'I<3Python!', '', etc.

Assigning value to a constant in Python

- In Python, constants are usually declared and assigned on a **module**.
- Here, the module means a new file containing variables, functions etc which is imported to main file.
- Inside the module, constants are written in all capital letters and underscores separating the words.

Ex: Declaring and assigning value to a constant

Create a constant.py

```
RADIUS = 3.14
GRAVITY = 9.8
```

Create a main.py

```
import constant
print(constant. RADIUS )
print(constant. GRAVITY)
```

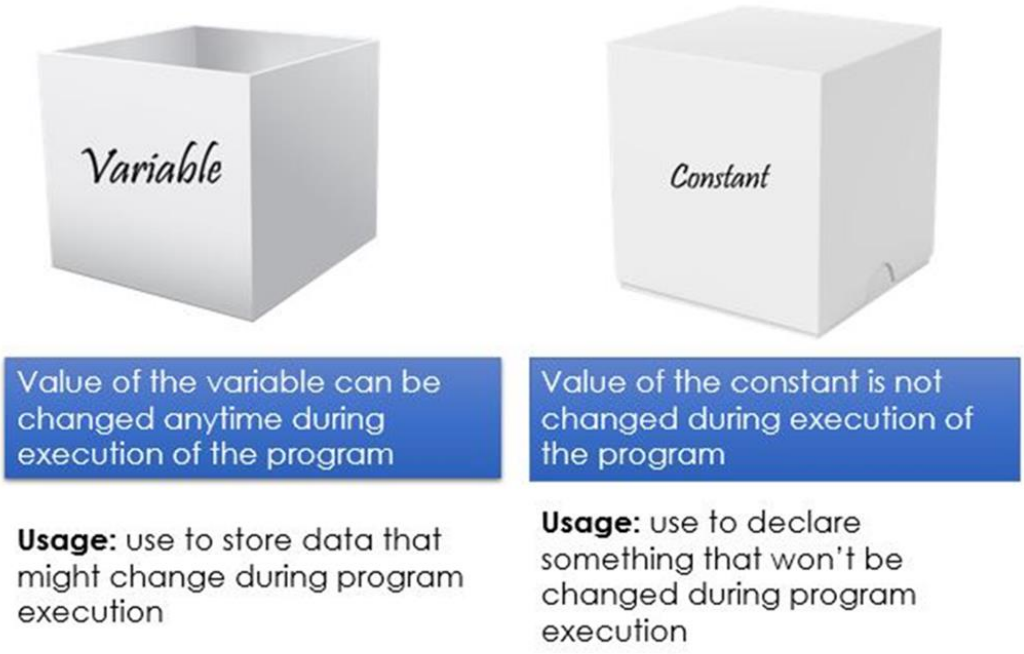
When you run the program, the output will be:

```
3.14
9.8
```

- In the above program, we create a constant.py module file. Then, we assign the constant value to RADIUS and GRAVITY.

- After that, we create a main.py file and import the constant module. Finally, we print the constant value.

Constants Vs Variables



- Both constants and variables can only hold one piece of data at a time.
- When a new value is assigned to variable (using an assignment statement) it automatically replaces whatever was stored previously.

- **Examples:**

Variable Name	Value
level	4
highscore	1202
surname	Smith

Constant Name	Value
VAT	20
days	365
bonus	100

Rules for naming variables:

- All variable names must begin with a letter of the alphabet (a-z, A-Z), an underscore (_). The convention is to always use a letter of the alphabet.
- After the first initial letter, variable names may also contain letters and the digits 0 to 9. No spaces or special characters are allowed.
- The name can be of any length, but don't get carried away. Remember that you will have to type this name.
- Uppercase characters are distinct from lowercase characters.
- Using ALL uppercase letters are primarily used to identify constant variables.
- Remember that variable names are case-sensitive.

Ex: case_sensitive, CASE_SENSITIVE, and Case_Sensitive are each a different variable.

- You cannot use a python keyword (reserved word) for a variable name.

Good Variable Name:

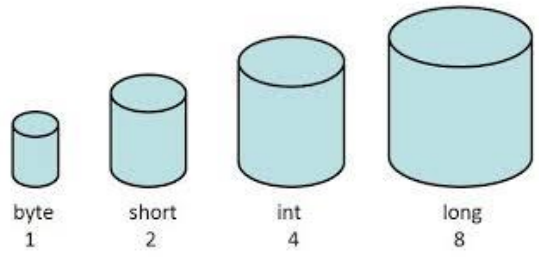
- Choose meaningful name instead of short name. **roll_no is better than rn**.
- Maintain the length of a variable name. **Roll_no_of_a-student is too long**?
- **Be consistent; roll_no or RollNo**
- Begin a variable name with an underscore (_) character for a special case.

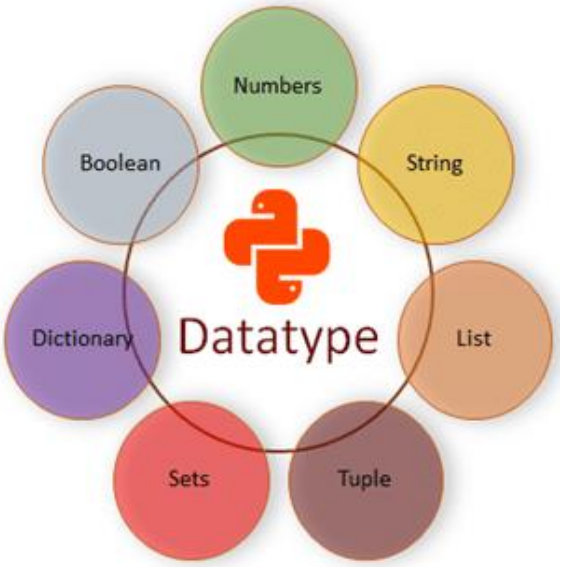
Samples of acceptable variable names: **YES**	Samples of unacceptable variable names: **NO**
Grade	Grade(Test)
GradeOnTest	GradeTest#1
Grade_On_Test	3rd_Test_Grade
GradeTest	Grade Test (has a space)
_Grade	#Grade

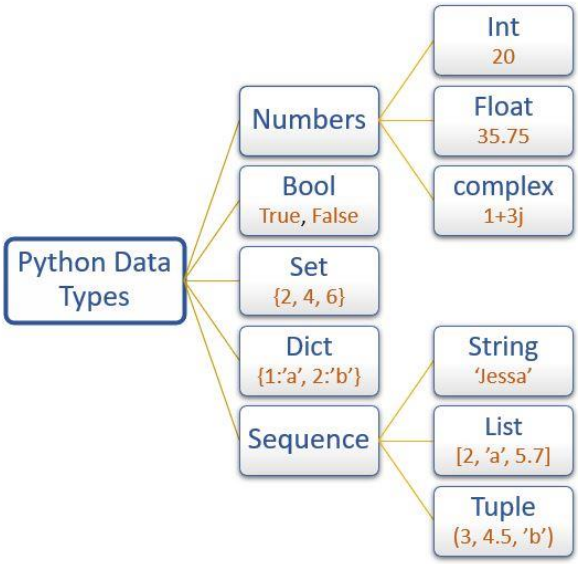
Primitive Data Types

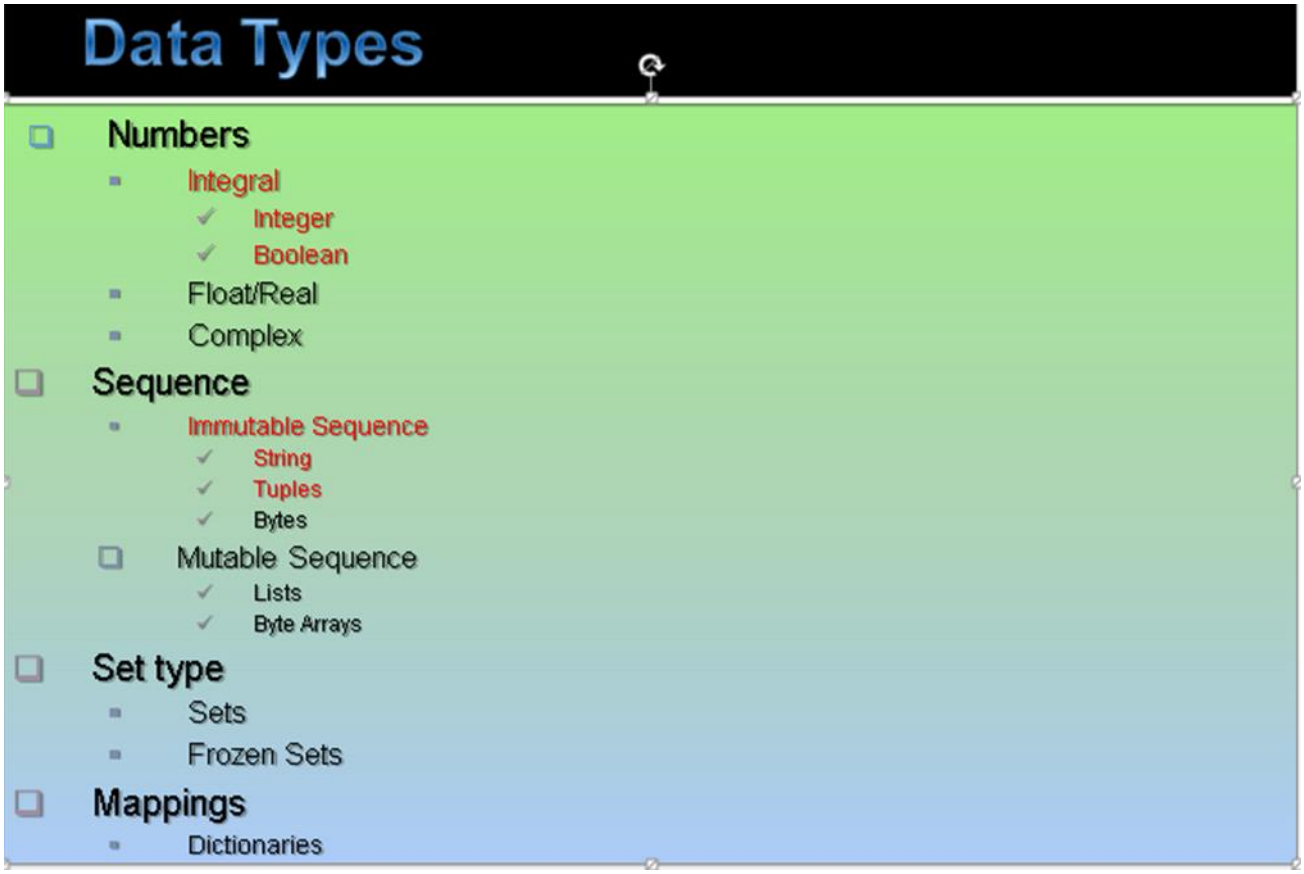
What is a Primitive Data Type?

Primitive data types are the most basic data types available in a programming language.








Data Types
Numbers
Integral
Integer
Boolean
Float/Real
Complex
Sequence
Immutable Sequence
String
Tuples
Bytes
Mutable Sequence
Lists
Byte Arrays
Set type
Sets
Frozen Sets
Mappings
Dictionaries