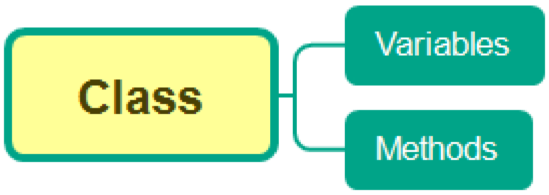
Last updated: 12 Oct 2021

Selenium Java Training - Session 7 - Java (Part 5) - Classes, Objects, Strings and Wrapper Classes

Java (Part 5) - Classes, Objects, Strings and Wrapper Classes

Classes and Objects

- Class encloses variables and methods - Demonstrate here



- **Class** is a template used for creating **Objects**
- Demonstrate creating a Class and use the Class as a template for creating Objects - Demonstrate here
    - Create a Class having a main method say Demo
    - Inside the same Java file create another Class named Car
    - Create any variables inside the Car class say model, cost, color
    - Create any methods inside the Car class say startCar(), stopCar(), carDetails()
    - Create any objects using Car class
    - Initialize and Access the variables & methods of Objects
- Object Creation Statement
    - **Car benz = new Car();** - View here

String

String is not a data type, instead it is a predefined class in Java Class Library

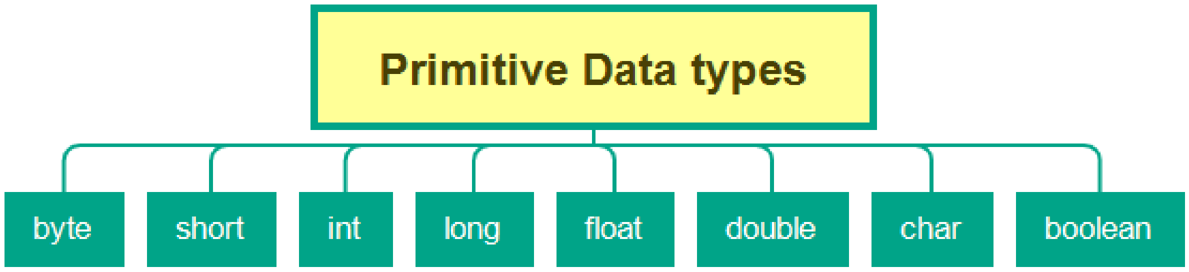
- Google "Java 11 API" and find the **String** class in the Java Class Library
- Actual Representation of String - String s = new String("Sample Text"); - Demonstrate here
- Shortcut Representation of String - String s = "Sample Text"; - Demonstrate here
- Concatenate two strings
    - using '+' operator - Demonstrate here
- Predefined methods of String class - Out of all the predefined methods of Strings, the below are the methods which are useful as part of Selenium Automation:
    - Using **equals()** method to compare two strings - Demonstrate here

- Using length() method to find the length of String literal text - Demonstrate here
- Using substring() method to retrieve the portion from the actual Str Save Copy to Evernote
  here
- Using **trim()** to remove the spaces before and after the string text - Demonstrate here
- Using **indexOf()** to check whether the provided text is in the provided paragraph. - Demonstrate here
  - Returns -1 in case the provided text is not available
- Using **split()** method to split the text into different parts based on the provided text, symbol or space.

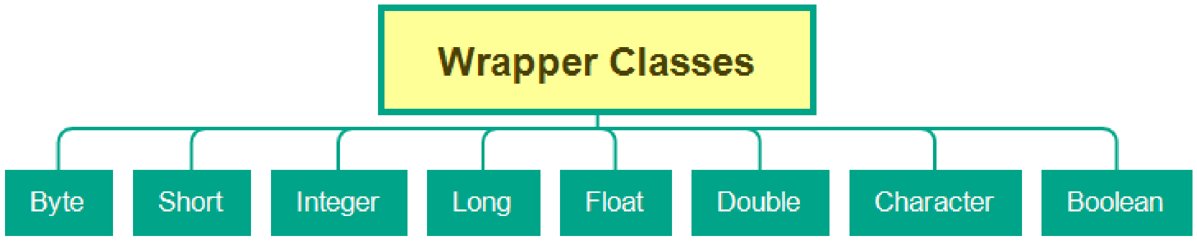
Wrapper Classes and Primitive Data types

In order to use primitive data types as Objects, we have to use Wrapper Classes which help us in converting the primitive data types into objects.

- The below are the different primitive data types:



- For, all the above data types, there are corresponding Wrapper Classes as shown below:



- Demonstrate a program which uses Wrapper Classes for converting the primitive data types into Objects and Objects into primitive data types - Demonstrate here
- String and using Wrapper classes for conversion to appropriate data type

By,
Arun Motoori