# ****Built in Methods in Java****

#### ****Categories of Built in Methods****

i) String Methods

ii) Number Methods

iii) Character Methods

iv) Array Methods etc…

##### ****i) Java String Methods****

****1) compareTo()**** Method (It compares two strings, supports 3-way comparison)  
Result Criteria for 3-way comparison

If str1 = str2 then 0  
If str1 > str2 then positive value  
If str1 < str2 then negative value  
——————–  
Result Criteria for 2-way comparison

If str1 = str2 then true  
If str1 (Greater than or Less than) str2 then false  
————————–  
Example:

public static void main (String [] args){  
String str1 = “selenium”;  
String str2 = “SELENIUM”;  
String str3 = “seleniuma”;  
String str4 = “selenium”;

System.out.println(str1.compareTo(str2));//Positive value  
System.out.println(str1.compareTo(str3));//Negative value  
System.out.println(str1.compareTo(str4));//0  
}  
}  
———————————  
****2) equals ()**** Method (It compares two strings and supports 2-way comparison)

Example:

public static void main (String [] args){  
String str1 = “selenium”;  
String str2 = “SELENIUM”;  
String str3 = “selenium”;

System.out.println(str1.equals(str2));//false  
System.out.println(str1.equals(str3));//true  
}  
————————————-  
****3) concat()**** Method (It concatenates two strings /Joins two strings)

public static void main (String [] args){  
String str1 = “Selenium”;  
String str2 = “Testing”;

System.out.println(str1.concat(str2));//SeleniumTesting  
System.out.println(str1 + str2);//SeleniumTesting  
}  
——————————  
****4) charAt()**** Method (Returns a character by index position)

public static void main (String [] args){  
String str1 = “Selenium”;

System.out.println(str1.charAt(1));//e  
}  
———————————-  
****5) equalsIgorecase()**** Method

public static void main (String [] args){  
String str1 = “SELENIUM”;  
String str2 = “selenium”;  
String str3 = “UFT”;  
System.out.println(str1.equalsIgnoreCase(str2));//true  
System.out.println(str1.equalsIgnoreCase(str3));//false  
}  
—————————  
****6) toUpperCase ()**** – Converts values to Upper case)

public static void main (String [] args){  
String str1 = “SELENIUM”;  
String str2 = “selenium”;  
String str3 = “SELEnium”;  
String str4 = “selenium123”;

System.out.println(str1.toUpperCase());//SELENIUM  
System.out.println(str2.toUpperCase());//SELENIUM  
System.out.println(str3.toUpperCase());//SELENIUM  
System.out.println(str4.toUpperCase());//SELENIUM123  
}  
————————–  
****7) toLowerCase()**** -Converts values to Lower case

public static void main (String [] args){  
String str1 = “SELENIUM”;  
String str2 = “selenium”;  
String str3 = “SELEnium”;  
String str4 = “selenium123”;

System.out.println(str1.toLowerCase());//selenium  
System.out.println(str2.toLowerCase());//selenium  
System.out.println(str3.toLowerCase());//selenium  
System.out.println(str4.toLowerCase());//selenium123  
}  
—————————————-  
****8) trim()**** Method (Removes spaces from both sides of a String)

public static void main (String [] args){  
String str1 = ” Selenium “;

System.out.println(str1);  
System.out.println(str1.trim());  
}  
———————————-  
****9) substring ()**** Method

public static void main (String [] args){  
String str = “Welecome to Selenium Testing”;

System.out.println(str.substring(12));//Selenium Testing  
System.out.println(str.substring(21));//Testing  
System.out.println(str.substring(12, 20));//Selenium  
System.out.println(str.substring(9, 11));//to  
}  
—————————————  
****10) endsWith()**** -Ends with specified suffix

public static void main (String [] args){  
String str = “Welcome to Selenium Testing”;

System.out.println(str.endsWith(“Selenium Testing”));//true  
System.out.println(str.endsWith(“Testing”));//true  
System.out.println(str.endsWith(“Selenium”));//false  
}  
——————————–  
****11) length()**** (returns string length)

public static void main (String [] args){  
String str = “Selenium Testing”;  
String str2 = “Selenium”;  
System.out.println(str.length());//16  
System.out.println(str2.length());//8  
}

##### ****ii) Java Number Methods****

****1) compareTo()**** Method (Number, 3-way comparison)

public static void main (String [] args){  
// Integer class wraps a value of the primitive type int in an object  
//An object of type Integer contains a single field whose type is int.

int x = 5;  
Integer a =x;  
System.out.println(a.compareTo(5));//0  
System.out.println(a.compareTo(6));//-1  
System.out.println(a.compareTo(4));//1  
}  
———————————  
****2) equals()**** Method (Number, 2-way comparison)

public static void main (String [] args){  
// Integer class wraps a value of the primitive type int in an object  
//An object of type Integer contains a single field whose type is int.

int x = 5;  
Integer a =x;  
System.out.println(a.equals(5));//true  
System.out.println(a.equals(6));//false  
System.out.println(a.equals(4));//false  
}  
——————————–  
****3) abs()**** -Returns absolute value

public static void main (String [] args){  
double a =10.234;  
double b =-10.784;  
System.out.println(Math.abs(a));//10.234  
System.out.println(Math.abs(b));//10.784  
}  
————————————  
****4) round()**** -It rounds the value to nearest integer

public static void main (String [] args){  
double a =10.234;  
double b =-10.784;  
double c =10.51;  
System.out.println(Math.round(a));//10  
System.out.println(Math.round(b));//-11  
System.out.println(Math.round(c));//11  
}  
———————————–  
****5) min()**** – Returns minimum value between two numbers

public static void main (String [] args){  
int a=10, b=20;  
double c =10.234, d =10.345;  
System.out.println(Math.min(a, b));//10  
System.out.println(Math.min(c, d));//10.234  
System.out.println(Math.min(7, 9));//7  
System.out.println(Math.min(1.23, 1.234));//1.23  
}  
———————————  
****6) max()****-Returns maximum value between two numbers

public static void main (String [] args){  
int a=10, b=20;  
double c =10.234, d =10.345;  
System.out.println(Math.max(a, b));//20  
System.out.println(Math.max(c, d));//10.345  
System.out.println(Math.max(7, 9));//9  
System.out.println(Math.max(1.23, 1.234));//1.234  
}  
————————————-  
****7) random()**** – Generates a random number

public static void main (String [] args){  
System.out.println(Math.random());//  
}

##### ****iii) Java Character Methods****

****1) isLetter()**** – Checks weather the value is Alphabetic or not?

public static void main (String [] args){  
//The Character class wraps a value of primitive data type char is an object

char a =’A’;  
char b =’1′;

System.out.println(Character.isLetter(a));//true  
System.out.println(Character.isLetter(b));//false  
System.out.println(Character.isLetter(‘Z’));//true  
System.out.println(Character.isLetter(‘1’));//false  
System.out.println(Character.isLetter(‘\*’));//false  
}  
———————  
public static void main (String [] args){  
//The Character class wraps a value of primitive data type char is an object  
char a =’A’;  
char b =’1′;  
System.out.println(Character.isAlphabetic(a));//true  
System.out.println(Character.isAlphabetic(b));//false  
System.out.println(Character.isAlphabetic(‘Z’));//true  
System.out.println(Character.isAlphabetic(‘1’));//false  
System.out.println(Character.isAlphabetic(‘\*’));//false  
}  
——————————  
Assignment:

What is the difference between isLetter() and isAlphabetic()  
————————————-  
****2) isDigit()**** -Checks weather the value is Number or not?

public static void main (String [] args){  
//The Character class wraps a value of primitive data type char is an object

char a =’A’;  
char b =’1′;  
System.out.println(Character.isDigit(a));//false  
System.out.println(Character.isDigit(b));//true  
System.out.println(Character.isDigit(‘Z’));//false  
System.out.println(Character.isDigit(‘1’));//true  
System.out.println(Character.isDigit(‘\*’));//false  
}  
——————————-  
****3) isUpperCase()**** – Checks weather the value is Upper case or not?

****4) isLowerCase()****-Checks weather the value is Lower case or not?

Examples:

public static void main (String [] args){  
//The Character class wraps a value of primitive data type char is an object

char a =’A’;  
char b =’z’;  
char c =’1′;

System.out.println(Character.isUpperCase(a));//true  
System.out.println(Character.isUpperCase(b));//false  
System.out.println(Character.isUpperCase(c));//false

System.out.println(Character.isLowerCase(a));//false  
System.out.println(Character.isLowerCase(b));//true  
System.out.println(Character.isUpperCase(c));//false  
}

##### ****iv) Java Array Methods****

****1) length**** -It returns length of the Array.

public class Sample1 {

public static void main (String [] args){  
int [] array1 = {10, 20, 30, 40};  
System.out.println(array1.length);//4  
}  
}  
————————-  
****2) toString()**** -It prints an Array.

public static void main (String [] args){  
String [] array1 = {“Selenium”, “UFT”, “LoadRunner”, “RFT”};  
String str = Arrays.toString(array1);  
System.out.println(str);  
}  
————————————-  
****3) contains()**** – Checks if the Array contains certain value or not?

public static void main (String [] args){  
String [] array1 = {“Selenium”, “UFT”, “LoadRunner”, “RFT”};  
boolean a = Arrays.asList(array1).contains(“UFT”);  
boolean b = Arrays.asList(array1).contains(“Java”);

System.out.println(a);//true  
System.out.println(b);//false  
}

## Java Constructors

A constructor in Java is a ****special method**** that is used to initialize objects. The constructor is called when an object of a class is created. It can be used to set initial values for object attributes:

### Example

Create a constructor:

// Create a MyClass classpublic class MyClass {

int x; // Create a class attribute

// Create a ****class constructor**** for the MyClass class

public MyClass() {

x = 5; // Set the initial value for the class attribute x

}

public static void main(String[] args) {

MyClass myObj = new MyClass(); // Create an object of class MyClass (This will ****call the constructor****)

System.out.println(myObj.x); // Print the value of x

}}

// Outputs 5

Note that the constructor name must ****match the class name****, and it cannot have a ****return type**** (like void).

Also note that the constructor is called when the object is created.

All classes have constructors by default: if you do not create a class constructor yourself, Java creates one for you. However, then you are not able to set initial values for object attributes.

## Constructor Parameters

Constructors can also take parameters, which is used to initialize attributes.

The following example adds an int y parameter to the constructor. Inside the constructor we set x to y (x=y). When we call the constructor, we pass a parameter to the constructor (5), which will set the value of x to 5:

### Example

public class MyClass {

int x;

public MyClass(int y) {

x = y;

}

public static void main(String[] args) {

MyClass myObj = new MyClass(5);

System.out.println(myObj.x);

}}

// Outputs 5

You can have as many parameters as you want:

### Example

public class Car {

int modelYear;

String modelName;

public Car(int year, String name) {

modelYear = year;

modelName = name;

}

public static void main(String[] args) {

Car myCar = new Car(1969, "Mustang");

System.out.println(myCar.modelYear + " " + myCar.modelName);

}}