

JAVA - NUMBERS CLASS

http://www.tutorialspoint.com/java/java_numbers.htm

Copyright © tutorialspoint.com

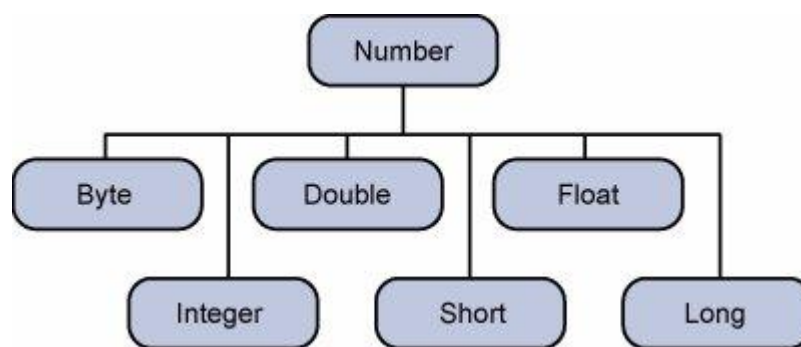
Normally, when we work with Numbers, we use primitive data types such as byte, int, long, double, etc.

Example:

```
int i = 5000;  
float gpa = 13.65;  
byte mask = 0xaf;
```

However, in development, we come across situations where we need to use objects instead of primitive data types. In-order to achieve this Java provides **wrapper classes**

All the wrapper classes *Integer*, *Long*, *Byte*, *Double*, *Float*, *Short* are subclasses of the abstract class *Number*.



The object of the wrapper class contains or wraps its respective primitive data type. converting primitive data types into object is called boxing, and this is taken care by the compiler. therefore while using a wrapper class you just need to pass the value of the primitive data type to the constructor of the Wrapper class.

And the Wrapper object will be converted back to a primitive data type, and this process is called unboxing. The **Number** class is part of the java.lang package.

Here is an example of boxing and unboxing:

```
public class Test{  
    public static void main(String args[]){  
        Integer x = 5; // boxes int to an Integer object  
        x = x + 10;    // unboxes the Integer to a int  
        System.out.println(x);  
    }  
}
```

This would produce the following result:

```
15
```

When x is assigned integer value, the compiler boxes the integer because x is integer object. Later, x is unboxed so that they can be added as integer.

Number Methods:

Here is the list of the instance methods that all the subclasses of the Number class implement:

SN Methods with Description

- 1 [xxxValue](#)
Converts the value of *this* Number object to the xxx data type and returned it.
- 2 [compareTo](#)
Compares *this* Number object to the argument.
- 3 [equals](#)
Determines whether *this* number object is equal to the argument.
- 4 [valueOf](#)
Returns an Integer object holding the value of the specified primitive.
- 5 [toString](#)
Returns a String object representing the value of specified int or Integer.
- 6 [parseInt](#)
This method is used to get the primitive data type of a certain String.
- 7 [abs](#)
Returns the absolute value of the argument.
- 8 [ceil](#)
Returns the smallest integer that is greater than or equal to the argument. Returned as a double.
- 9 [floor](#)
Returns the largest integer that is less than or equal to the argument. Returned as a double.
- 10 [rint](#)
Returns the integer that is closest in value to the argument. Returned as a double.
- 11 [round](#)
Returns the closest long or int, as indicated by the method's return type, to the argument.
- 12 [min](#)

Returns the smaller of the two arguments.

13

[max](#)

Returns the larger of the two arguments.

14

[exp](#)

Returns the base of the natural logarithms, e, to the power of the argument.

15

[log](#)

Returns the natural logarithm of the argument.

16

[pow](#)

Returns the value of the first argument raised to the power of the second argument.

17

[sqrt](#)

Returns the square root of the argument.

18

[sin](#)

Returns the sine of the specified double value.

19

[cos](#)

Returns the cosine of the specified double value.

20

[tan](#)

Returns the tangent of the specified double value.

21

[asin](#)

Returns the arcsine of the specified double value.

22

[acos](#)

Returns the arccosine of the specified double value.

23

[atan](#)

Returns the arctangent of the specified double value.

24

[atan2](#)

Converts rectangular coordinates x, y to polar coordinate r, θ and returns θ .

25

[toDegrees](#)

Converts the argument to degrees

26

[toRadians](#)

Converts the argument to radians.

27

[random](#)

Returns a random number.

What is Next?

In the next section, we will be going through the Character class in Java. You will be learning how to use object Characters and primitive data type char in Java.

Loading [MathJax]/jax/output/HTML-CSS/jax.js