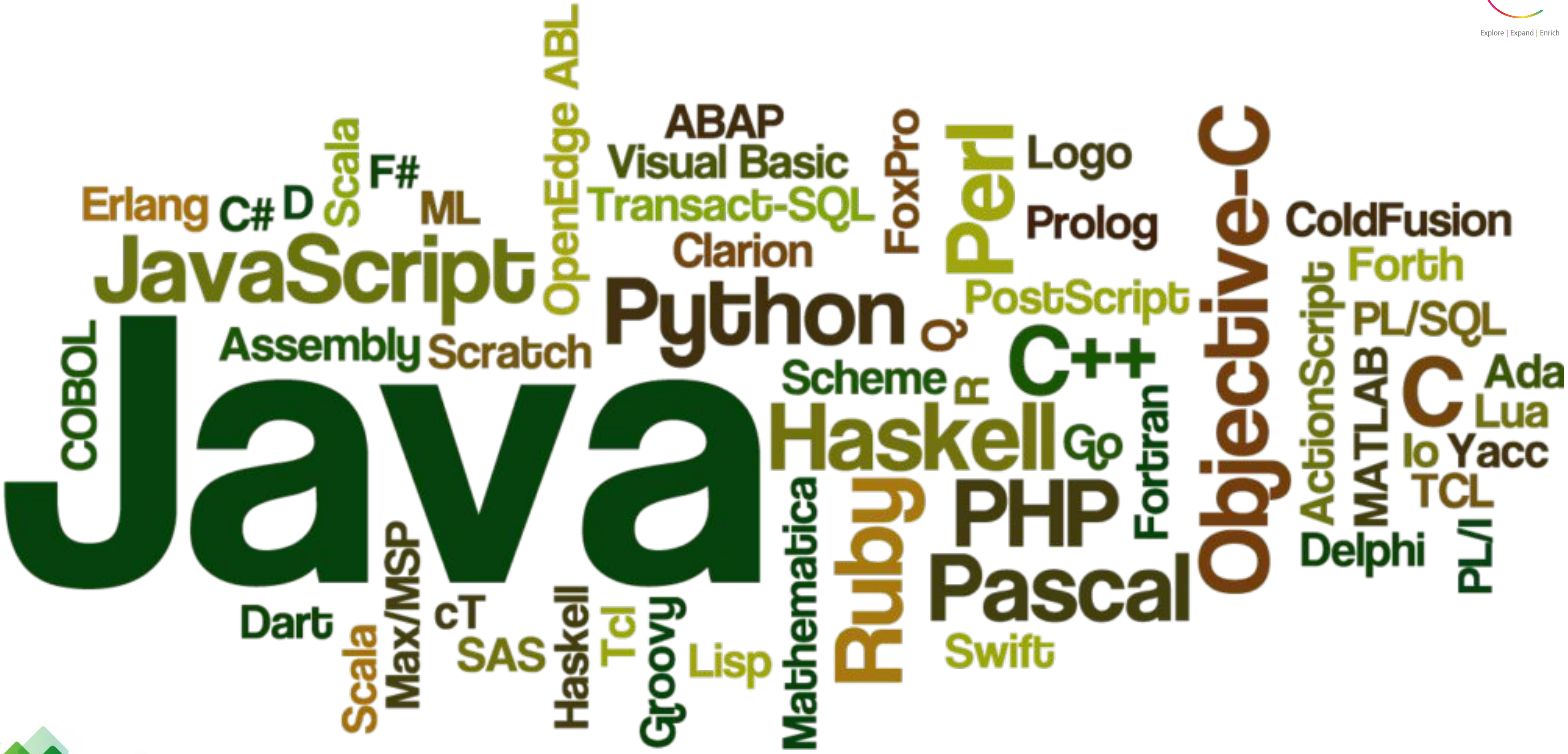
ETHNUS™

Explore | Expand | Enrich

Explore | Expand | Enrich



Date and Time
IN
Java

.toString() : java.util.Date.tostring()

- method is a java.util.Date class method.It displays the Current date and time
- Here Date object is converted to a string and represented as

```
day mon dd hh:mm:ss zz yyyy
```

Syntax:

```
public String toString()
```

Return:

a string representation of the given date.

day	day of the week
mon	month
dd	day of the month
hh	hour
mm	minute
ss	second
zz	time zone
yyyy	year upto 4 decimal places

.setTime() : java.util.Date.setTime()

- method is a java.util.Date class method. Sets this Date object to represent a point in time that is time milliseconds after January 1, 1970 00:00:00 GMT

Syntax:

```
        public void setTime(long time)
```

Parameters:

time : the number of milliseconds.

.hashCode() : java.util.Date.hashCode()

- method is a java.util.Date class method. Returns a hash code value for the Date object. The result is exclusive OR of the two halves of the primitive long value returned by the getTime() method

Syntax:

```
      public int hashCode()
```

Return:

a hash code value for the Date object.

use of .toString(), setTime(), hashCode() method

```
import java.util.*;
public class NewClass {
     public static void main(String[] args) {
     Date mydate = new Date();
     System.out.println("System date : " + mydate.toString());
     mydate.setTime(15680);
     System.out.println("Time after setting:  " + mydate.toString());
     int d = mydate.hashCode();
     System.out.println("Amount (in ms) by which time" + " is shifted :  " + d);
     }
}
```

.after() : java.util.Date.after()

- method tests if current date is after the given date

Syntax:

```
public boolean after(Date d)
```

Parameters:

d : date

Return:

true if and only if the instant represented by this Date object is strictly later than the instant represented by 'when'; else false

Exception:

NullPointerException - if Date object is null.

.clone() : java.util.Date.clone()

- method returns the duplicate of passed Date object

Syntax:

```
public Object clone()
```

Return:

a clone of this instance.

.before() : java.util.Date.after()

- method tests if current date is before the given date

Syntax:

```
public boolean before(Date d)
```

Parameters:

d : date

Return:

true if and only if the instant represented by this Date object is strictly earlier than the instant represented by 'when'; else false

Exception:

NullPointerException - if when is null.

use of .toString(), setTime(), hashCode() methods

```
import java.util.Date;
public class NewClass {
     public static void main(String[] args){
          Date date1 = new Date(2016, 11, 18);
          Date date2 = new Date(1997, 10, 27);
          boolean a = date2.after(date1);
          System.out.println("Is date2 is after date1 : " + a);
          date1.after(date2);
          System.out.println("Is date1 is after date2 : " + a);
          System.out.println("");
          Object date3 = date1.clone();
          System.out.println("Cloned date3 :" + date3.toString());
          System.out.println("");
          boolean b = date2.before(date1);
          System.out.println("Is date2 is before date1 : " + a);
     }
}
```

.compareTo() : java.util.Date.compareTo()

- method compares two dates and results in -1, 0 or 1 based on the comparison

Syntax:
```
public int compareTo(Date argDate)
```
Parameters:
argDate : another date to compare with
Result:
0 : if the argumented date = given date.
-1 : if the argumented date > given date.
1 : if the argumented date < given date.

.equals() : java.util.Date.equals()

- method checks whether two dates are equal or not based on their millisecond difference

Syntax:
```
public boolean equals(Object argDate)
```
Parameters:
argDate : another date to compare with
Result:
true if both the date are equal; else false.

.getTime() : java.util.Date.getTime()

- method results in count of milliseconds of the argumented date, referencing January 1, 1970, 00:00:00 GMT

Syntax:

```
public long getTime()
```

Result:

milliseconds of the argumented date, referencing January 1, 1970, 00:00:00 GMT.

use of .toString(), setTime(), hashCode() methods

```
import java.util.*;
public class NewClass {
    public static void main(String[]
args) {
        Date d1 = new Date(97, 10, 27);
        Date d2 = new Date(97, 6, 12);
        int comparison =
d1.compareTo(d2);
        int comparison2 =
d2.compareTo(d1);
        int comparison3 =
d1.compareTo(d1);
        System.out.println("d1 > d2 : "
+ comparison);
        System.out.println("d1 < d2 : "
+ comparison2);
        System.out.println("d1 = d1 : " +
comparison3);
        System.out.println(""); boolean r1 =
d1.equals(d2);
        System.out.println("Result of equal()
r1 : " + r1);
        boolean r2 = d1.equals(d1);
        System.out.println("Result of equal()
r2 : " + r2);
        System.out.println("");
        long count1 = d1.getTime();
        long count2 = d1.getTime();
System.out.println("Milliseconds of d1 : " +
count1);
        System.out.println("Milliseconds of
d2 : " + count2);
    }
}
```

THANK YOU