

# آموزش زبان برنامه نویسی جاوا

## کلاس String

جلسه چهاردهم

نویسنده: رحمان زارعی

جاوا را ساده، آسان و شیرین بنوشید!!!!



عبارت String خیلی آشناست نه؟! درست حدس زدید ما در جلسات قبل ازش یک استفاده هایی بردیم اما فقط سطحی و وارد عمقش نشدیم!!!! String یا رشته که در جلسات قبل برای ذخیره متن یا جمله مورد نظر یا به عبارتی متغیری بود که بیش از یک کاراکتر میشد داخلش ریخت و ذخیره کرد! حالا قصد داریم بصورت مفصل به جناب String بزرگ پردازیم!!! بعد از مطالعه این بخش ، از این که یک جلسه را به String اختصاص دادیم بهش حق خواهید داد که چقدر بزرگه!!! خب بریم سراغ اصل مطلب!!!!!!

رشته ها یا Strings به طور گسترده در زبان برنامه نویسی استفاده می شود، رشته ها از مجموعه ای از کاراکتر های متوالی تشکیل شده، رشته یا String همون جمله است دیگه!!! چون جمله هم از مجموعه ای از کاراکتر تشکیل شده است!!

در زبان برنامه نویسی جاوا رشته (String) یک کلاس می باشد که می توان از آن شی ساخت.

پلتفرم جاوا کلاس String را برای ایجاد و دستکاری رشته ها فراهم میکند.

## ایجاد یک String (رشته):

دو روش برای ایجاد رشته یا String وجود دارد یکی راه مستقیم و دیگری راه غیر مستقیم ، که متداول ترین راه که بیشتر استفاده می شود راه مستقیم می باشد:

راه مستقیم برای ایجاد رشته (String) :

```
String greeting = "Hello Iran!";
```

- ابتدا نوع متغیر را String تعیین می کنیم، بعد نام متغیر `greeting` انتخاب کرده، بعد رشته یا جمله (مجموعه ای از کاراکترها) را بهش نسبت می دهیم ("`Hello Iran!`")
- رشته ای که قرار هست داخل متغیر String قرار دهیم باید درون دو " " قرار بگیرد. شبیه مثال

`"Hello Iran!"`

📍 راه غیرمستقیم برای ایجاد رشته (String): البته این راه هم مستقیم هست اما راه اولی مستقیم ترین بود حالا من اینجا گفتم غیر مستقیم که یک تفاوتی قائل شویم 😊

```
String greeting =new String("Hello Iran!");
```

در اینجا یک شی از کلاس String ساخته ایم و درون سازنده کلاس String، پارامتر رشته "`Hello Iran!`" به عنوان ورودی بهش داده ایم.

- برای اینکار ابتدا نوع شی را کلاس String تعیین می کنیم، بعد نام شی را `greeting` انتخاب کرده، بعد کلمه کلیدی `new` در پایان از سازنده کلاس String استفاده میکنیم و درون آن رشته یا جمله (مجموعه ای از کاراکترها) را قرار می دهیم ("`Hello Iran!`")
  - String ساخته شده از هر دو راه حاوی مقدار رشته مورد نظر می باشد و این را در نظر بگیرید که هر دو راه `greeting`، یک شی است که حاوی مقدار رشته می باشد. و با چاپ `greeting` مقدار `Hello Iran` در خروجی نمایش داده می شود:
- ❖ مثال راه مستقیم ایجاد رشته :

```
package javalike;
//https://telegram.me/javalike
//author:Rahman Zahrai
public class SampleString {
    public static void main(String[] args) {
        String greeting = "Hello Iran!";
        System.out.println(greeting);
    }
}
```

خروجی:

```
Hello Iran!
```

❖ مثال راه غیرمستقیم ایجاد رشته :

```
package javalike;

//https://telegram.me/javalike
//author:Rahman Zahrai
public class SampleString {

    public static void main(String[] args) {
        String greeting = new String("Hello Iran!");
        System.out.println(greeting);
    }
}
```

خروجی:

```
Hello Iran!
```

- همان طور که از هر دو راه مشاهده میکنیم خروجی یکی هست ، فقط روش ایجاد رشته (String) فرق می کند.

## به دست آوردن طول رشته (String Length) :

طول رشته یا String به تعداد کاراکتر های موجود درون رشته می گویند.

مثلا طول رشته "Hello Iran!" عدد ۱۱ می باشد. یعنی این رشته از ۱۱ حرف یا کاراکتر تشکیل شده است.

برای به دست آوردن طول رشته در زبان برنامه نویسی جاوا کلاس String متد `length()` را ارائه کرده است.

همان طور که گفتیم String یک کلاس می باشد که حاوی سری ویژگی ها و رفتار هاست. یکی از این رفتار ها ،متد `length()` می باشد. وقتی ما یک شی از این کلاس می سازیم ،از طریق شی ساخته شده از آن می توان به این متد و سایر متدهای آن دسترسی پیدا کنیم.

برای به دست آوردن طول یک رشته با استفاده از متد `length()` بصورت زیر عمل می کنیم:

(۱) نام شی ساخته شده از کلاس String

(۲) استفاده از نقطه (.) برای اشاره کردن به ویژگی ها و متدهای کلاس String

(۳) از بین ویژگی ها و متدهای موجود انتخاب متد `length()`

فرآیند ۱ تا ۳ را بصورت مثال کلی زیر مشاهده می فرمایید:

```
String StringObjectName=" String value";
```

• تعریف و مقدار دهی مستقیم به شی ساخته شده از کلاس String

```
StringObjectName.length();
```

• به دست آوردن طول رشته که یک عدد صحیح را به ما پس می دهد (برمی گرداند)، که این مقدار را می توان درون یک متغیر عدد صحیح قرار داد.

مثال زیر طریقه به دست آوردن طول رشته با استفاده از متد length() می باشد:

```
package javalike;
//https://telegram.me/javalike
//author:Rahman Zahrai
public class SampleString {
    public static void main(String[] args) {
        String greeting = "Hello Iran!";
        int len = greeting.length();
        System.out.println("String Length is : " + len);
    }
}
```

خروجی:

```
String Length is : 11
```

• یک رشته (String) تعریف کرده و به آن مقداری مستقیم کرده ایم.

```
String greeting = "Hello Iran!";
```


• برای به دست آوردن طول یک رشته ابتدا نام رشته (شی ساخته شده از کلاس String) سپس با نقطه (.) به متد length() اشاره می کنیم و مقدار صحیح که به ما برمی گرداند را درون یک متغیر صحیح قرار داده ایم.

```
int len = greeting.length();
```

• در پایان طول رشته را در خروجی چاپ می کنیم.


```
System.out.println("String Length is : " + len);
```

❖ برای درک بهتر این مثال، بصورت تصویری این فرآیند را در زیر مشاهده کنید:

نوشتن نام کلاس String (خط ۸) در بدنه کلاس SampleString درون متد main. تصویر (۱) 

```
1 package javalike;
2
3 //https://telegram.me/javalike
4 //author:Rahman Zarei
5 public class SampleString {
6
7     public static void main(String[] args) {
8         String
9
10    }
11
12 }
13
```

تصویر (۱)

تعیین یک نام (greeting) برای شی از نوع کلاس String (خط ۸). تصویر (۲) 

```
1 package javalike;
2
3 //https://telegram.me/javalike
4 //author:Rahman Zarei
5 public class SampleString {
6
7     public static void main(String[] args) {
8         String greeting |
9
10    }
11
12 }
13
```

تصویر (۲)

مقداردهی کردن شی ساخته شده از کلاس String (خط ۸). تصویر (۳)

```
1 package javalike;
2
3 //https://telegram.me/javalike
4 //author:Rahman Zarei
5 public class SampleString {
6
7     public static void main(String[] args) {
8         String greeting = "Hello Iran!";
9
10    }
11
12 }
13
```

تصویر (۳)

تعریف یک متغیر و تعیین نام آن (len) از نوع عدد صحیح (خط ۹). تصویر (۴)

```
1 package javalike;
2
3 //https://telegram.me/javalike
4 //author:Rahman Zarei
5 public class SampleString {
6
7     public static void main(String[] args) {
8         String greeting = "Hello Iran!";
9         int len
10
11     }
12
13 }
14
```

تصویر (۴)

برای به دست آوردن طول رشته greeting و ریختن آن درون متغیر len کفایت ابتدا سمت راست مساوی نام شی greeting را تایپ کنیم (خط ۹). تصویر (۵)

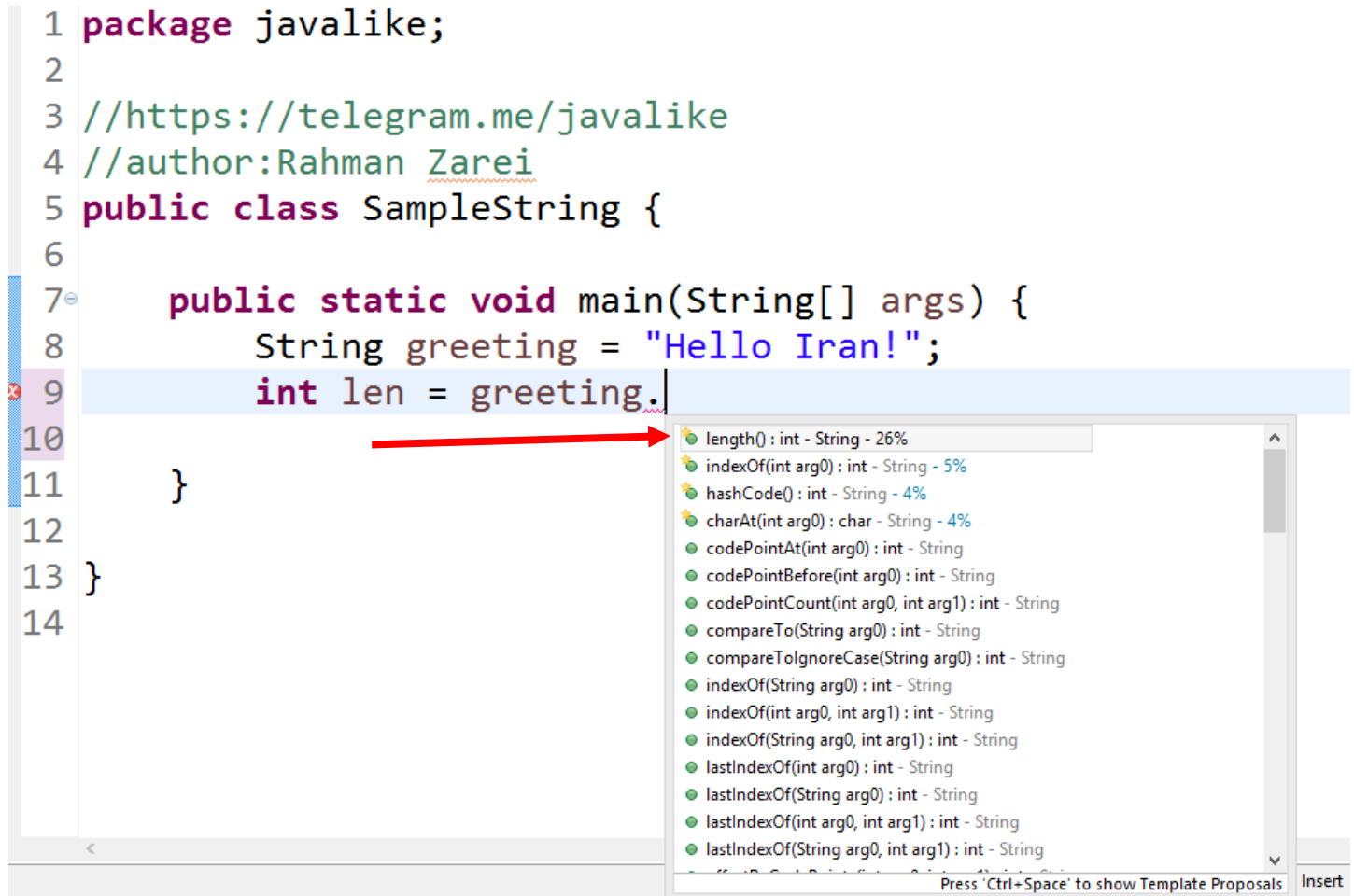


```
1 package javalike;
2
3 //https://telegram.me/javalike
4 //author:Rahman Zarei
5 public class SampleString {
6
7     public static void main(String[] args) {
8         String greeting = "Hello Iran!";
9         int len = greeting
10
11     }
12
13 }
14
```

تصویر (۵)

با استفاده از شی `greeting` و نقطه (.) به متدهای کلاس `String` دست پیدا میکنیم (برای این کار بعد از نام شی و نقطه دکمه ترکیبی `ctrl+space` را بزنید لیست متدها بالا می آید) که از میان این متدها ،متد `length()` را انتخاب میکنیم (خط ۹). تصویر (۶) و (۷)

```
1 package javalike;
2
3 //https://telegram.me/javalike
4 //author:Rahman Zarei
5 public class SampleString {
6
7     public static void main(String[] args) {
8         String greeting = "Hello Iran!";
9         int len = greeting.
10
11     }
12
13 }
14
```



The image shows a screenshot of a Java IDE. The code editor displays a Java class named `SampleString` with a `main` method. The code is as follows:

```
1 package javalike;
2
3 //https://telegram.me/javalike
4 //author:Rahman Zarei
5 public class SampleString {
6
7     public static void main(String[] args) {
8         String greeting = "Hello Iran!";
9         int len = greeting.
10
11     }
12
13 }
14
```

A dropdown menu is open at line 9, showing a list of methods for the `String` class. The first method, `length() : int - String - 26%`, is highlighted. A red arrow points from this method in the dropdown to the `length()` method in the code on line 9. The dropdown menu also includes other methods like `indexOf(int arg0) : int - String - 5%`, `hashCode() : int - String - 4%`, `charAt(int arg0) : char - String - 4%`, `codePointAt(int arg0) : int - String`, `codePointBefore(int arg0) : int - String`, `codePointCount(int arg0, int arg1) : int - String`, `compareTo(String arg0) : int - String`, `compareToIgnoreCase(String arg0) : int - String`, `indexOf(String arg0) : int - String`, `indexOf(int arg0, int arg1) : int - String`, `indexOf(String arg0, int arg1) : int - String`, `lastIndexOf(int arg0) : int - String`, `lastIndexOf(String arg0) : int - String`, `lastIndexOf(int arg0, int arg1) : int - String`, and `lastIndexOf(String arg0, int arg1) : int - String`. The bottom of the dropdown menu has the text "Press 'Ctrl+Space' to show Template Proposals" and an "Insert" button.

تصویر (۶)

```
1 package javalike;
2
3 //https://telegram.me/javalike
4 //author:Rahman Zarei
5 public class SampleString {
6
7     public static void main(String[] args) {
8         String greeting = "Hello Iran!";
9         int len = greeting.length();
10
11     }
12
13 }
14
```

تصویر (۷)

- با استفاده از شی greeting و نقطه (.) به متدهای کلاس String دست پیدا میکنیم (برای این کار بعد از نام شی و نقطه دکمه ترکیبی ctrl+space را بزنید لیست متدها بالا می آید) که از میان این متدها ،متد length() را انتخاب میکنیم (خط ۹). تصویر (۶) و (۷)
- چاپ مقدار طول رشته (len) در خروجی تصویر (۸)

```

package javalike;

//https://telegram.me/javalike
//author:Rahman Zarei
public class SampleString {

    public static void main(String[] args) {
        String greeting = "Hello Iran!";
        int len = greeting.length();
        System.out.println("String Length is : " + len);
    }
}

```

تصویر (۸)

پایان 

## الحاق (اتصال) دو رشته به یکدیگر (Concatenating Strings):

گاهی نیاز داریم که دو رشته جدا از هم را به هم وصل یا الحاق کنیم.

مثلا فرض کنید رشته s1 مقدارش "Hello" و رشته s2 مقدارش "Iran" باشد و قصد داریم یک رشته جدید به نام s3 تعریف کنیم که مقدارش برابر الحاق رشته های s1,s2 یعنی "Hello Iran" باشد. برای این کار از متد `concat(string arg0)` بصورت زیر استفاده می کنیم:

```
s3=s1.concat(s2);
```

❖ مقدار رشته s2 را به s1 وصل و در نهایت رشته جدید که حاصل الحاق رشته s1,s2 می باشد را درون رشته s3 قرار می دهد.

• مثال:

```
"My name is ".concat("Zahra");
```

خروجی:

```
My name is Zahra
```

## راهی ساده تر برای الحاق چند رشته به یکدیگر:

کافیست با قرار دادن عملگر + (جمع) میان رشته ها، آنها را به هم وصل کنیم .

```
String s1 = "Hello ";  
String s2="Iran";  
String s3="!";  
String s4=s1+s2+s3;
```

مقدار s4 برابر است با

Hello Iran!

حالا تصمیم با شماست که برای الحاق رشته ها به یکدیگر کدام راه را انتخاب کنید. من که راه دوم و راحت تر را

انتخاب میکنم 😊

مثال:

```
package javalike;  
  
//https://telegram.me/javalike  
//author:Rahman Zahrai  
public class SampleString {  
  
    public static void main(String[] args) {  
        String string1 = "saw I was ";  
        System.out.println("Dot " + string1 + "Tod");  
    }  
}
```

خروجی:

Dot saw I was Tod

- همان طور که مشاهده میکنید با استفاده از عملگر جمع (+) سه رشته را به هم اتصال و در پایان در خروجی چاپ کرده است.

## متدهای کلاس String (String Methods)

در اینجا لیستی از متدهای پر کاربرد کلاس String جمع اوری شده است:

## char charAt(int index)

نکته: هر کاراکتر یا حرف درون یک رشته یک اندیس یا شماره دارد به عبارتی اولین کاراکتر رشته دارای شماره ۰، دومین دارای شماره ۱ سومین دارای شماره ۲، و...، nمین کاراکتر دارای شماره طول رشته منهای یک می باشد (stringName.length-1) مثلا رشته "Hello" طولی به اندازه ۵، کاراکتر H آن دارای شماره ۰، و کاراکتر o دارای شماره ۴. در کل هر کاراکتر یک رشته در یک خانه قرار دارد که از شماره ۰ شروع می شود.

0	1	2	3	4
H	e	l	l	o

حروف یا کارکترهای رشته Hello بصورت جدول بالا که حروف آن از شماره ۰ تا length-1 می باشد.

حالا کاربرد متد `char charAt(int index)`: این متد عدد صحیح به عنوان ورودی میگیرد، و این عدد اندیس مکان کاراکتر موجود در رشته می باشد، که با توجه به اندیس مقدار کاراکتر موجود در مکان آن اندیس را بر میگرداند.

به طور واضح و بی مقدمه این متد مقدار کاراکتر مکان `index` ام رشته را بر می گرداند.

هیچی بهتر از مثال عملی برای درک کردن نیست!!!

```
package javalike;


//https://telegram.me/javalike
//author:Rahman Zahrai
public class SampleString {

    public static void main(String[] args) {
        String string = "Everything is great!";

        System.out.println(string.charAt(19));
    }
}
```

خروجی:

!

مقدار کاراکتر در مکان ۱۹ ام رشته را برمیگرداند. 

## boolean equals(Object anObject)

این متد برای مقایسه یک رشته با یک شی استفاده می شود که عموماً برای مقایسه دو رشته از لحاظ برابری کاربرد دارد. یعنی اگر مقدار دو رشته برابر true و اگر نابرابر باشد false بر می گرداند.  
مثال:

```
package javalike;

//https://telegram.me/javalike
//author:Rahman Zahrai
public class SampleString {

    public static void main(String[] args) {
        String string = "Everything is great!";

        System.out.println(string.equals("Everything"));
    }
}
```

خروجی:

```
false
```


## int indexOf(String str)

یک رشته (تنها یک حرف بصورت رشته) میگیرد و شماره مکان آن را به ما می دهد.

```
package javalike;
//https://telegram.me/javalike
//author:Rahman Zahrai
public class SampleString {
    public static void main(String[] args) {
        String string = "Everything is great!";
        System.out.println(string.indexOf("s"));
    }
}
```

خروجی:

```
12
```

رشته "S" در مکان ۱۱ ام رشته string قرار داشت. 

## String replace(arg0, arg1)

رشته arg1 را جایگزین رشته arg0 که از قبل درون رشته مان وجود داشته ، میکند.

```
package javalike;

//https://telegram.me/javalike
//author:Rahman Zahrai
public class SampleString {

    public static void main(String[] args) {
        String string = "Everything is great!";

        System.out.println(string.replace("thing", "day"));
    }
}
```

خروجی:

```
Everyday is great!
```

## String[] split(String regex)

نکته : در مورد این تابع ،مقداری که برمی گرداند به شکل **String[]** می باشد و این که ،این شکل پیاده سازی چیه و جزییاتش ....یک جلسه مفصل بهش می پردازیم. تنها این را بدانید که مقداری که این متد برمیگرداند بصورت آرایه ای از رشته (String) می باشد. حالا آرایه چیه؟؟؟؟؟؟!!!!!! آرایه یک ساختمان داده برای ذخیره مقدار می باشد چیزی شبیه متغیر با کاربرد و ظرفیت بیشتر!!! اصلا نگران نباشید کامل فراموشش کنید تا این که جلسات بعد بهش می پردازیم اونوقت به سراغ این بخش بیایید و می بینید که چقدر ساده بوده!!!!

این متد رشته اصلی را براساس رشته **regex** جدا میکند و مقدار حاصل را درون خانه های آرایه میریزد. به عبارتی رشته اصلی را براساس رشته regex تکه تکه می کند و هر تکه را درون خانه ای از آرایه قرار می دهد.

```
package javalike;

//https://telegram.me/javalike
```



```
//author:Rahman Zahrai
public class SampleString {

    public static void main(String[] args) {
        String string = "Everything is great!";
        String str[] = string.split(" ");
        System.out.println(str[0]);
        System.out.println(str[1]);
        System.out.println(str[2]);

    }

}
```

خروجی:

```
Everything
is
great!
```

خب regex ما در اینجا رشته " " (تک اسپیس با تک فاصله) می باشد، خب حالا رشته string را بر اساس رشته " " تکه تکه (جدا سازی) می کنیم که به ۳ تکه تبدیل می شود چون از رشته Everything شروع که کنیم به یک اسپیس (" ") برخورد کرده در نتیجه این تکه داخل خانه اول آرایه قرار میگیرد. بعد از اسپیس (" ") بعد برخورد میکنیم و رشته is نیز تکه دوم ما را تشکیل می دهد و درون خانه دوم آرایه قرار میگیرد، حالا از اسپیس (" ") دوم تا great! در نهایت تکه سوم ما را تشکیل می دهد و درون خانه سوم آرایه قرار میگیرد.

حالا مقدار خانه اول تا سوم آرایه را (خانه آرایه نیز شبیه String از صفر شروع تا طول آرایه -۱ خاتمه می یابد) چاپ می کنیم.

اصلا نگران آرایه نباشید این متد یکی از پرکاربرد ترین متد کلاس String است که لازم بود در موردش توضیح بدم اما چون هنوز به مبحث آرایه نرسیدیم فعلا نادیده بگیرینش و وقتی آرایه را آموزش دادم به سراغ این بخش بیایید!!!

بازم تکرار میکنم هیچ ترس و نگرانی در مورد آرایه نداشته باشید!!!! جلسه بعدی حلش میکنیم 😊

## String substring(int beginIndex, int endIndex)

این متد زیر رشته ای از رشته اصلی را در بازه `[beginIndex, endIndex)` برمیگرداند.

یعنی از مقدار `beginIndex` شروع می کند تا بازه مقدار کمتر از `endIndex` (این مقدار شامل بازه نمی شود) را بصورت رشته بر می گرداند.

حداقل مقدار برای `beginIndex` می تواند صفر و حداکثر مقدار برای `endIndex` می تواند طول رشته - ۱ باشد.

برای مقدار دهی کردن این متد باید قانون `beginIndex < endIndex` رعایت شود.

```
package javalike;

//https://telegram.me/javalike
//author:Rahman Zahrai
public class SampleString {

    public static void main(String[] args) {
        String string = "Everything is great!";
        System.out.println(string.substring(0, 10));

    }

}
```

خروجی:

```
Everything
```

## String toLowerCase()

این متد تمام کاراکترهای موجود درون رشته را تبدیل به حروف کوچک می کند.

## String toUpperCase()

این متد تمام کاراکترهای موجود در یک رشته را به حروف یا کاراکترهای بزرگ تبدیل می سازد.

```
package javalike;

//https://telegram.me/javalike
//author:Rahman Zahrai
```

```
public class SampleString {

    public static void main(String[] args) {
        String string = "EverYThiNg Is GreAt!";
        System.out.println(string);
        System.out.println("toLower= "+string.toLowerCase());
        System.out.println("toUpper= "+string.toUpperCase());
    }
}
```

خروجی:

```
EverYThiNg Is GreAt!
toLower= everything is great!
toUpper= EVERYTHING IS GREAT!
```

کاملاً واضح هست که ابتدا رشته اصلی بدون تغییر چاپ، خط بعد تمامی حروف رشته اصلی به حروف کوچک تبدیل و سپس چاپ می شود، و در خط آخر بعد تمامی حروف رشته اصلی به حروف بزرگ تبدیل و سپس چاپ می شود.

کلاس String یکی از مهم ترین کلاس های جاواست که کاربرد فراوان دارد. این کلاس هنوز هم متد دارد که پرکاربرد ترین هاشو بررسی کردیم. از نظر من نیاز نیست این متد ها رو حفظ کنید و یا حتی یاد بگیرید!!!! چون وقتی مورد خاصی رو نیاز دارید میرید یاد میگیرید، استفاده که کردید بعد بازم فراموشش کنید!!!! نیاز نیست همه چیزو یاد بگیرید یا حفظ کنید!!! تنها زمانی که ضرورت یافت از شون استفاده می کنید همین. من همین الان که دارم این آموزش را می نویسم هیچ از جاوا یادم نمیاد و بلد نیستم!!!! اما زمانی که موردی رو نیاز داشتیم با یک سرچ کوچک یادم میاد و استفاده میکنم!!!!!! الکی ذهنمون رو پر نکنیم، ذهن باید خالی باشد تا کار کند!!!

امیدوارم از این جلسه آموزشی راضی باشید، این آموزش ها رو با دوستان خود یا کسی که نیاز دارد به اشتراک بگذارید البته با ذکر منبع ☺

پیروز و موفق باشید

سایت آموزش زبان جاوا به زبان ساده، آسان و شیرین!!!

[www.JAVAPRO.ir](http://www.JAVAPRO.ir)

آموزش جاوا SE را با تجربه شخصی و به زبان خودمونی یاد بگیرید!!!!

# بازدید از کانال

# بازدید از سایت

هر روز مفاهیم و مثال های جدید به سایت اضافه می شود برای اطلاع از مطالب جدید روی سایت عضو کانال شوید