

Core Java

آموزش ساده و آسان جاوا

به نام خدا

تقدیرم به هموطنان عزیزم

جاوا را با لذت یاد بگیرید!

Core Java

آموزش ساده و آسان جاوا

آموزش زبان برنامه نویسی جاوا

نمونه مثال از کلاس String

متد startsWith()

نویسنده : رحمان زارعی

جاوا را ساده، آسان و شیرین بنوشید!!!



این جلسه آموزشی رایگان است، فروش و ویرایش آن ممنوع و حرام می باشد. اما این کتاب را می توانید همین جور که هست در سایت و شبکه اجتماعی خود به اشتراک بگذارید.

Core Java

آموزش ساده و آسان جاوا

سلام. روزت بخیر دوست من ☺

در این جلسه آموزشی قصد داریم به بررسی متد `startsWith()` یکی از متدهای موجود در کلاس `String` بپردازیم.

برای چک کردن پیشوند (prefix) یک `String` از این متد استفاده می کنیم. میدونم الان سوال برات پیش اومده که چی شد!! یعنی چه؟؟!!

بریم سراغ زبان فارسی ()))))):

به چه چیزی پیشوند می گوئیم؟!

در دستور زبان، به کلمه یا حرفی که **اول** کلمه دیگر قرار بگیرد و معنی آن را تغییر دهد ، پیشوند می گوئیم.

خب ما در اینجا کاری به تغییر معنی کلمه نداریم! پس با چه کار داریم؟! در بالا گفتیم ، "به کلمه ای که اول کلمه ی دیگر قرار بگیرد پیشوند می گوئیم" ما به این بخش از معنی پیشوند کار داریم.

در جمله "به کلمه ای که **اول** کلمه ی دیگر قرار بگیرد پیشوند می گوئیم" اون واژه **اول** خیلی برامون مهم است!

مثال:

پیشوند کلمه "برانگیختن" می تواند کلمه "بر" باشد. چرا؟ چون کلمه "بر" **اول** یا **ابتدای** کلمه "برانگیختن" قرار گرفته است.

پیشوند کلمه "فراخور" می تواند کلمه "فر" باشد. چرا؟ چون کلمه "فر" **اول** و **ابتدای** کلمه "فراخور" قرار گرفته است.

- بنا بر رسم اخلاق منبع سایتی که معنی کلمه پیشوندی رو ازش استخراج کردم رو براتون میزارم:

<https://jadvalyab.ir/fa2en/پیشوند/>

پس در صورتی کلمه `a1` می تواند پیشوند کلمه `a2` باشد که کلمه `a1` ابتدای کلمه `a2` قرار بگیرد.

```
a1="java";
a2="javalike";
```

در مثال بالا `a1` پیشوند (prefix) `a2` است، زیرا `String` (رشته) `a1` ابتدای کلمه `a2` قرار گرفته است.

Core Java

آموزش ساده و آسان جاوا

```
a1="like";  
a2="javalikey";
```

در مثل بالا a1 پیشوند (prefix) کلمه a2 نیست. زیرا کلمه a1 ابتدای کلمه a2 قرار نگرفته است.

- پس تا اینجا یاد گرفتیم که کلمه ای که **ابتدا و آغاز** کلمه دیگری قرار بگیرد، اون کلمه نقش پیشوند (prefix) کلمه دیگر رو برعهده دارد.

همان طور که ابتدای جلسه گفتیم متد `startsWith()` برای چک کردن پیشوند (prefix) یک `String` کاربرد دارد. حالا معنای پیشوند (prefix) برای متد `startsWith()` چیست؟

با مثال براتون توضیح میدم که مفهومی بهتر روشن بشه:

فرض کنید یک `String` با نام `a1` به صورت زیر داشته باشیم:

```
String a1 = "javapro.ir";
```

و `String` دیگری با نام `a2` به صورت زیر داشته باشیم:

```
String a2 = "javapro";
```

- `String` (رشته) `a2` پیشوند `String` (رشته) `a1` می باشد. چرا؟ چون رشته `a2` ابتدای رشته `a1` قرار گرفته است.

حال فرض کنید رشته `a1` به صورت زیر باشد:

```
String a1 = "Java Tutorial for Complete Beginners";
```

و رشته `a2` نیز به صورت زیر باشد:

```
String a2 = "Java Tutorial";
```

- در اینجا `String` (رشته) `a2` پیشوند `String` (رشته) `a1` می باشد. چرا؟ چون رشته `a2` ابتدای رشته `a1` قرار گرفته است.

Core Java

آموزش ساده و آسان جاوا

- در اینجا خبری از کلمه نبود! بلکه حرف از رشته (String) بود، "به دنباله ای از کاراکترها، رشته می گوییم" پس اگر رشته ای (String) ابتدای یا آغاز کننده رشته ای (String) دیگر بود، به اون رشته آغاز کننده پیشوند (prefix) می گوییم.

اووووووووووووف!!! کل اینارو گفتم که بدونید که کار متد `startsWith()` چیست! متد `startsWith()` میاد یک String رو چک می کنه که آیا پیشوند (prefix) رشته ای دیگر هست یا خیر؟! جلوتر مثال می زنیم که بهتر متوجه بشید.

متد `startsWith()` با دو پارامتر مجزا وجود دارد:

```
boolean startsWith(String str):
```

- این متد رشته `str` را چک می کند که آیا پیشوند رشته ای دیگر است یا خیر؟ به عبارت دیگر چک می کند که آیا رشته `str` آغاز کننده و ابتدای رشته دیگری هست یا خیر؟ اگر بود مقدار `true` و اگر نبود مقدار `false` را به ما پس می دهد.

Example:

```
1. package javalike;
2. public class StartsWithMethodExample {
3.     public static void main(String args[]) {
4.         String str1 = "Java Tutorial for Complete Beginners";
5.         String str2 = "Java Tutorial";
6.         System.out.println(str1.startsWith(str2));
7.         System.out.println(str1.startsWith("for Complete"));
8.         System.out.println(str1.startsWith("Java"));
9.         System.out.println(str1.startsWith("Ja"));
10.        System.out.println(str1.startsWith("J"));
11.        System.out.println(str1.startsWith("Tutorial"));
12.        System.out.println(str1.startsWith("Beginners"));
13.    }
14. }
```

خروجی (output):

```
true
```

Core Java

آموزش ساده و آسان جاوا

```
false
true
true
true
false
false
```

۴. یک String تعریف کرده و آن را مقداردهی اولیه کرده ایم.

۵. یک String دیگر تعریف کرده و آن را مقداردهی اولیه کرده ایم.

۶. با استفاده از متد `startsWith` را چک کرده ایم که آیا رشته `str2` پیشوند (prefix) رشته `str1` هست یا خیر. به دلیل این که رشته `str2` ابتدای رشته `str1` قرار گرفته است، پس `str2` پیشوند `str1` می باشد و خروجی مقدار `true` می شود.

۷. در خط هفتم رشته "for Complete" جزیی از رشته `str1` می باشد اما به دلیل این که ابتدای رشته `str1` قرار نگرفته است، پس رشته "for Complete" پیشوند رشته `str1` نمی باشد و مقدار خروجی `false` می باشد.

۸. در خط هشتم رشته "Java" ابتدای رشته `str1` قرار گرفته است، پس رشته "Java" پیشوند رشته `str1` می باشد، و مقدار خروجی `true` می باشد.

۹. در خط نهم رشته "Ja" ابتدای رشته `str1` قرار گرفته است، پس رشته "Ja" پیشوند رشته `str1` می باشد، و مقدار خروجی `true` می باشد

۱۰. در خط دهم رشته "J" ابتدای رشته `str1` قرار گرفته است، پس رشته "J" پیشوند رشته `str1` می باشد، و مقدار خروجی `true` می باشد

۱۱. در خط یازدهم رشته "Tutorial" جزیی از رشته `str1` می باشد اما به دلیل این که ابتدای رشته `str1` قرار نگرفته است، پس رشته "Tutorial" پیشوند رشته `str1` نمی باشد و مقدار خروجی `false` می باشد.

۱۲. در خط دوازدهم رشته "Beginners" جزیی از رشته `str1` می باشد اما به دلیل این که ابتدای رشته `str1` قرار نگرفته است، پس رشته "Beginners" پیشوند رشته `str1` نمی باشد و مقدار خروجی `false` می باشد.

نتیجه گیری کلی: اگر رشته `s1` آغاز کننده و ابتدای رشته `s2` قرار گرفته باشد، رشته `s1` پیشوند (prefix) رشته `s2` می

باشد، حالا متد `startsWith` کارش چک کردن این است که یک رشته پیشوند رشته دیگر هست یا خیر.

Core Java

آموزش ساده و آسان جاوا

چطور بود؟ امیدوارم تا اینجا به صورت ساده و روان آموزش داده باشم. 😊

`boolean startsWith(String str, int fromIndex)`

- این متد نیز برای چک کردن پیشوند بودن یا نبودن یک `String` در `String` ای دیگر کاربرد دارد. تنها یک تفاوت دارد و اون پارامتر `fromIndex` می باشد.
- نقش پارامتر `fromIndex` چیست؟ اگر رشته `str` از نقطه `fromIndex` رشته مورد نظر آغاز شده باشد، رشته `str` پیشوند رشته مورد نظر می باشد. باز میدونم معلوم نشد چی چی شد!!!! مثال میزنم که بهتر متوجه بشید.
- فرض کنید یک `String` با نام `a` به صورت زیر داشته باشیم:

```
String a="java like Tutorial";
```

و همچنین `String` دیگری با نام `str` به صورت زیر داشته باشیم:

```
String str="like Tutorial";
```

حال متد `startsWith(String str, int fromIndex)` را به صورت زیر صدا می زنیم:

```
a.startsWith(str, 5)
```

- به دلیل این که از ایندکس شماره پنجم رشته `str` پیشوند رشته `a` می باشد، متد `startsWith` مقدار `true` برمی گرداند.
- برای درک بهتر به ایندکس های رشته `a` در زیر توجه کنید:

Core Java

آموزش ساده و آسان جاوا

ایندکس های رشته `a="java like Tutorial"` به صورت زیر می باشد:

0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17
j	a	v	a		l	i	k	e		T	u	t	o	r	i	a	l

از طرفی طبق متد `a.startsWith(str, 5)` مقدار `fromIndex` برابر ۵ است، و رشته ای که قراره به عنوان پیشوند بودن یا نبودن رشته `a` چک شود برابر `str="like Tutorial"` می باشد. در اینجا گفته شده که `fromIndex` برابر ۵ می باشد، یعنی رشته `str` را از ایندکس پنجم رشته `a` بررسی کن، اگر پیشوند بود مقدار `true` و در غیر این صورت مقدار `false` رو متد `a.startsWith(str, 5)` به ما پس می دهد.

ایندکس پنجم به بعد رشته `a` را با رنگ سبز مشخص کرده ایم:

0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17
j	a	v	a		l	i	k	e		T	u	t	o	r	i	a	l

حال اگر رشته `str` از ایندکس پنجم رشته `a` پیشوند بود متد `a.startsWith(str, 5)` مقدار `true` را برمی گرداند، همان طور که می بینید رشته `str` ابتدای ایندکس پنجم رشته `a` قرار گرفته است (ناحیه سبز رنگ) پس رشته `str` پیشوند رشته `a` می باشد.

a.startsWith(str, n)

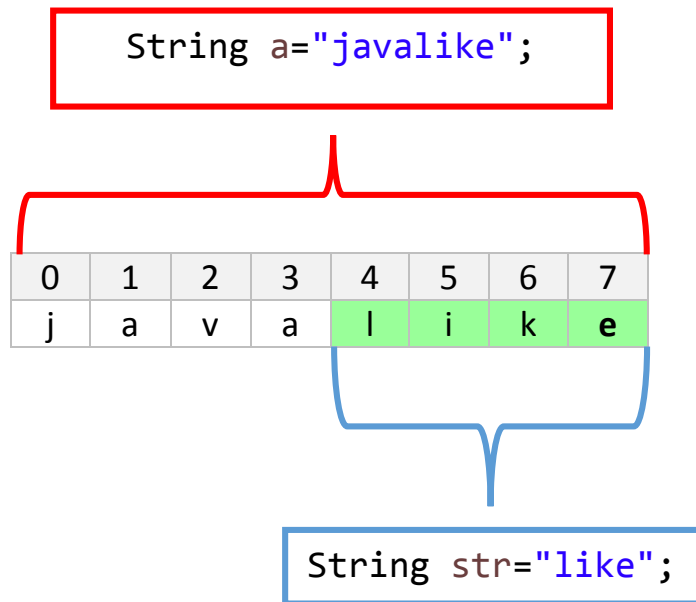
به طور کلی در متد `startsWith` اگر رشته `str` در ایندکس `n` ام رشته `a` پیشوند بود، این متد مقدار `true` برمی گرداند و در غیر این صورت مقدار `false` را به ما پس می دهد. یا به عبارتی دیگر اگر رشته `str` آغازکننده در ایندکس `n` ام رشته `a` بود، رشته `str` پیشوند رشته `a` خواهد بود و متد `startsWith` مقدار `true` برمی گرداند و در غیر این صورت مقدار `false` برمی گرداند.

خب شاید براتون توضیحات واضح نبوده باشه، چاره چیست؟ نگران نباشید، به مثال های زیر توجه کنید قطعا مفهومی دستتون میاد:

Core Java

آموزش ساده و آسان جاوا

```
String a="javalike";
String str="like";
```



شکل (۱)

طبق شکل (۱)، ما یک String (رشته) با نام a و یک String (رشته) با نام str داریم. ایندکس های رشته a را در شکل (۱) مشاهده می کنید که با توجه به تعداد کاراکترهای رشته a از ۰ تا ۷ می باشد. حال قصد داریم پیشوند بودن رشته str در ایندکس های متفاوت رشته a را با استفاده از متد `startsWith` چک کنیم:

Example:

```
1. package javalike;
2. public class TestStartsWith {
3.     public static void main(String[] args) {
4.         String a = "javalike";
5.         String str = "like";
6.         System.out.println(a.startsWith(str, 4));
7.     }
```

Core Java

آموزش ساده و آسان جاوا

| 8. }

خروجی (output):

true

- با توجه به شکل (۱) و دستور خط شماره ۶، رشته str در ایندکس ۴ رشته a، پیشوند رشته a می باشد. به همین دلیل خروجی true می باشد.

Example:

```
package javalike;

public class TestStartsWith {

    public static void main(String[] args) {
        String a = "javalike";
        String str = "like";
        System.out.println(a.startsWith(str, 0));
    }
}
```

خروجی (output):

false

- خروجی false است. زیرا رشته str در ایندکس شماره 0 رشته a، پیشوند رشته a نمی باشد. به عبارتی رشته str در ایندکس شماره 0 رشته a آغازکننده نمی باشد.

Example:

```
package javalike;

public class StartsWithExample {
    public static void main(String args[]) {
        String str = new String("quick brown fox jumps over the lazy dog");
        System.out.println("String str starts with quick: "
            + str.startsWith("quick"));
        System.out.println("String str starts with brown: "
            + str.startsWith("brown"));
    }
}
```

Core Java

آموزش ساده و آسان جاوا

```

        System.out
        .println("substring of str(starting from 6th index) has
brown prefix: "
                + str.startsWith("brown", 6));
        System.out
        .println("substring of str(starting from 6th index) has
quick prefix: "
                + str.startsWith("quick", 6));
    }
}

```

خروجی (output):

```

String str starts with quick: true
String str starts with brown: false
substring of str(starting from 6th index) has brown prefix: true
substring of str(starting from 6th index) has quick prefix: false

```

توضیحات:

```

String str = new String("quick brown fox jumps over the lazy dog");
System.out.println("String str starts with quick: "
+ str.startsWith("quick"));

```

- یک رشته با نام `str` داریم و بررسی کردیم که آیا رشته `"quick"` پیشوند رشته `str` هست یا خیر. به دلیل این که رشته `"quick"` ابتدا و آغاز کننده رشته `str` است، پیشوند رشته `str` می باشد و خروجی متد `startsWith` مقدار `true` خواهد بود.

```

System.out.println("String str starts with brown: "
+ str.startsWith("brown"));

```

- به دلیل این که رشته `"brown"` ابتدا و آغاز کننده در رشته `str` نیست یا به عبارت دیگر پیشوند رشته `str` نیست خروجی متد `startsWith` مقدار `false` خواهد بود.

```

System.out.println("substring of str(starting from 6th index) has brown prefix:
"+ str.startsWith("brown", 6));

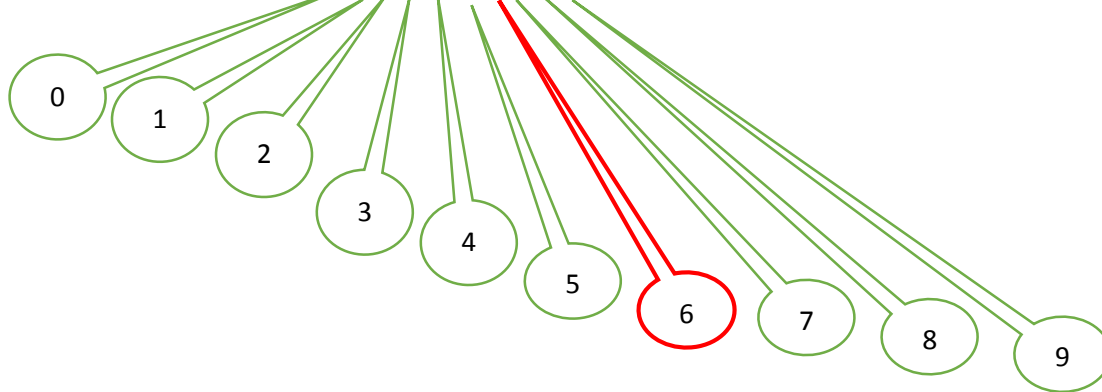
```

Core Java

آموزش ساده و آسان جاوا

خب اینجا از متد `startsWith` که ایندکس میگیرد استفاده کرده ایم. گفته آیا رشته "brown" در ایندکس شماره شش رشته `str` پیشوند است یا خیر؟ با هم ایندکس های رشته `str` را می شماریم:

"quick brown fox jumps over the lazy dog"



- همان طور که در بالا می بینید، رشته "brown" در ایندکس شماره ۶ رشته `str` پیشوند می باشد، زیر رشته `str.startsWith("brown", 6)` پس خروجی دستور قرار گرفته است. برابر `true` می باشد.

```
System.out.println("substring of str(starting from 6th index) has quick prefix: "+ str.startsWith("quick", 6));
```

- به دلیل این که رشته "quick" در ایندکس شماره ۶ رشته `str` پیشوند نمی باشد، مقدار خروجی دستور `str.startsWith("quick", 6)` برابر `false` می باشد.

پیروز و موفق باشید

Core Java

آموزش ساده و آسان جاوا

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

www.JAVAPRO.ir

آموزش جاوا SE را با تجربه شفاهی 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.

دقل و تصرف ، ویرایش و کپی زدن تمامی آموزش های جاوا لایک به دور از افلاق حرفه ای ست و مرا می باشد.