

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

متر indexOf()

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 به خیر. دوست عزیز متد `indexOf()` برای پیدا کردن ایندکس (index) یک کاراکتر مشخص یا زیر رشته ای (substring) از یک رشته (String) مشخص در جاوا کاربرد دارد.

ایندکس (index) چیست؟ به شماره خانه های کاراکترهای موجود در یک String ایندکس (index) می گویند.

مثال: ایندکس های رشته `String str = "javalike"` به صورت زیر است:

| 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 |
|---|---|---|---|---|---|---|---|
| j | a | v | a | l | i | k | e |

- مثلاً ایندکس کاراکتر 'j' برابر ۰ و ایندکس کاراکتر 'e' برابر ۷ می باشد.

متد `indexOf()` با چهار پارامتر مختلف در کلاس `String` وجود دارد:

```
int indexOf(int ch):
```

ایندکس (index) اولین کاراکتر `ch` موجود در یک `String` را برمی گرداند.

```
package javalike;

public class Test4 {

    public static void main(String[] args) {
        String str = "javalike";
        System.out.println(str.indexOf('a'));
    }
}
```

Core Java

آموزش ساده و آسان جاوا

خروجی (output):

1

- در برنامه بالا قصد داریم که شماره ایندکس کاراکتر 'a' را در رشته "javalike" پیدا کنیم. برای این کار از متد `indexOf(int ch)` استفاده کرده ایم. با وجود این که کاراکتر 'a' در دو ایندکس متفاوت از رشته "javalike" قرار گرفته است، یعنی هم در ایندکس شماره ۰ و هم در ایندکس شماره ۳ در رشته "javalike" کاراکتر 'a' موجود می باشد، اما متد `indexOf(int ch)`، ایندکس شماره ۱ را به ما داد. چرا؟ زیرا متد `indexOf(int ch)` ایندکس (index) اولین کاراکتر ch موجود در یک String را برمی گرداند.

نکته: در متد `indexOf(int ch)` اگر کاراکتر ch در یک String پیدا نشد، این متد مقدار -۱ را برای ما برمی گرداند.

Example:

```
package javalike;

public class Test4 {

    public static void main(String[] args) {
        String str = "javalike";
        System.out.println(str.indexOf('D'));
    }
}
```

خروجی (output):

-1

- به دلیل این که کاراکتر 'D' در رشته "javalike" وجود نداشت، متد `indexOf` مقدار -۱ را به عنوان خروجی به ما پس داد.

Core Java

آموزش ساده و آسان جاوا

`int indexOf(int ch, int fromIndex)`

این متد چکار می کند؟! این متد ایندکس اولین کاراکتر `ch` موجود در یک `String` را برمی گرداند. پارامتر `fromIndex` چه نقشی برعهده دارد؟ **نقطه شروع بررسی** کاراکترهای `String` برای پیدا کردن ایندکس کاراکتر `ch` از خانه (ایندکس) `fromIndex` رشته (`String`) می باشد. براتون مثال میزنم که این توضیح رو متوجه بشید.

```
package javalike;

public class Test4 {

    public static void main(String[] args) {
        String str = "javalike";
        System.out.println(str.indexOf('a'));
    }
}
```

خروجی (output):

1

- در مثال بالا متد `indexOf` شماره ایندکس اولین کاراکتر 'a' موجود در رشته "javalike" را برای ما به دست آورد. یعنی متد `indexOf` شماره ایندکس کاراکتر 'a' موجود در رشته "javalike" که بعد از حرف 'j' و قبل از حرف 'v' قرار داشت را برای ما برگرداند.
- حالا ما قصد داریم که شماره کاراکتر 'a' که بعد از حرف 'v' و قبل از حرف 'l' در رشته "javalike" قرار دارد را به دست آوریم. برای این کار ما باید به خانم!!! `indexOf` بگیم از یک نقطه مشخصی که ما برات تعیین می کنیم شروع به پیدا کردن ایندکس کاراکتر مورد نظر ما که در اینجا کاراکتر 'a' که بعد از حرف 'v' و قبل از حرف 'l' قرار دارد، کن.

Core Java

آموزش ساده و آسان جاوا

- ابتدا باید از متد `indexOf(int ch, int fromIndex)` استفاده کنیم. به جای پارامتر `ch` کاراکتر مورد نظر که در اینجا کاراکتر ما 'a' می باشد و به جای پارامتر `fromIndex` نقطه (ایندکس) شروع بررسی کاراکتر به کاراکتر رشته مورد نظر که در اینجا "javalike" است را قرا می دهیم.

Example:

```
package javalike;

public class Test4 {

    public static void main(String[] args) {
        String str = "javalike";
        System.out.println(str.indexOf('a', 2));
    }
}
```

خروجی (output):

3

- در مثال بالا ما ایندکس کاراکتر 'a' را از خانه شماره ۲ یعنی از کاراکتر 'v' رشته "javalike" شروع به بررسی و پیدا کردن، کردیم. چرا؟ چون هدف ما به دست آوردن ایندکس کاراکتر 'a' بعد از کاراکتر 'v' و قبل از کاراکتر 'l' که در رشته "javalike" قرار داشت، بود.
- دستور `str.indexOf('a', 2)` کاراکتر 'a' را از ایندکس شماره ۲ رشته "javalike" بررسی کرده، و در صورت وجود داشتن، ایندکس آن را به ما برمی گرداند. به عبارتی دیگر هنگام بررسی کاراکترهای 'a', 'j' که ابتدای رشته "javalike" قرار داشتند را نادیده گرفته است.
- به زبان ساده براتون بگم؟! معنی دستور `str.indexOf('a', 2)` اینه که، میگه ایندکس کاراکتر 'a' رو از کاراکتر ۲ به بعد رشته `str` بررسی و پیدا کن.

نتیجه کلی: در کل اگر قصد داریم از **نقطه خاصی (fromIndex)** از یک `String` شروع به **پیدا کردن ایندکس** کاراکتر مورد نظر (`ch`) خود کنیم از متد زیر استفاده می کنیم.

Core Java

آموزش ساده و آسان جاوا

```
int indexOf(int ch, int fromIndex)
```

متد بعدی.....

```
int indexOf(String str):
```

- ایندکس رشته `str` را در یک `String` مشخص برمی گرداند.

نکته: برخی مباحث هستند که توضیح دادنشون بیشتر آدمو رو برای یادگیری گیج میکنه، به همین خاطر بهتره با مثال های متعدد مفهوم اون مبحث رو یاد بگیریم.

Example:

```
package javalike;

public class Test4 {

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

خروجی (output):

4

- در مثال بالا رشته `"like"` در ایندکس شماره ۴ رشته `"javalike"` قرار داشت. به همین خاطر دستور `a.indexOf(str)` مقدار ۴ را برای ما برگرداند.

Core Java

آموزش ساده و آسان جاوا

Example:

```
package javalike;

public class Test4 {

    public static void main(String[] args) {
        String a = "java like free";
        String str1 = "java";
        String str2 = "like";
        String str3 = "free";
        String str4 = "like free";
        String str5 = "java like";
        String str6 = "ke";
        System.out.println(a.indexOf(str1));
        System.out.println(a.indexOf(str2));
        System.out.println(a.indexOf(str3));
        System.out.println(a.indexOf(str4));
        System.out.println(a.indexOf(str5));
        System.out.println(a.indexOf(str6));
    }
}
```

خروجی (output):

```
0
5
10
5
0
7
```

- در مثال بالا رشته "java" در خانه (ایندکس) شماره ۰ رشته a قرار دارد.
- در مثال بالا رشته "like" در خانه (ایندکس) شماره ۵ رشته a قرار دارد.
- در مثال بالا رشته "free" در خانه (ایندکس) شماره ۱۰ رشته a قرار دارد.
- در مثال بالا رشته "like free" در خانه (ایندکس) شماره ۵ رشته a قرار دارد.
- در مثال بالا رشته "java like" در خانه (ایندکس) شماره ۰ رشته a قرار دارد.
- در مثال بالا رشته "ke" در خانه (ایندکس) شماره ۷ رشته a قرار دارد.

Core Java

آموزش ساده و آسان جاوا

نکته: شماره ایندکس اولین کاراکتر رشته str در یک String مشخص ، تعیین کننده ایندکس رشته str می باشد.

مثلا: در مثال بالا رشته "free" در رشته "java like free"؛ در رشته "free" در رشته "java like free" برابر ۱۰ می باشد، پس شماره ایندکس رشته "free" در رشته "java like free" نیز برابر ۱۰ می باشد.

متد بعدی.....

```
int indexOf(String str, int fromIndex)
```

- این متد برای پیدا کردن ایندکس رشته str از ایندکس fromIndex شروع به بررسی String مورد نظر می کند. به عبارتی دیگر اولین رشته str بعد از ایندکس fromIndex رشته مورد نظر را پیدا می کند.

Example:

```
package javalike;

public class Test4 {

    public static void main(String[] args) {
        String a = "java like free java";

        String str1 = "java";

        System.out.println(a.indexOf(str1, 3));
    }
}
```

خروجی (output):

15

توضیحات:

```
System.out.println(a.indexOf(str1, 3));
```

Core Java

آموزش ساده و آسان جاوا

- در دستور بالا گفته شده است ایندکس رشته `str1` را از ایندکس سوم به بعد رشته `a` بررسی کن و در صورت موجود بودن برای ما برگردان.
- با وجود این که رشته `"java"` را در ابتدای رشته `a` داریم یعنی در ایندکس شماره ۰ اما به دلیل این که به متد `indexOf(str1, 3)` گفتیم ایندکس های سوم به بعد رشته `a` رو بررسی کن ، ایندکس های ۰ تا ۲ رشته `a` نادیده گرفته می شوند و متد `indexOf` از ایندکس سوم به بعد میگرده و ایندکس رشته `str1` را در رشته `a` پیدا می کند.

نکته: همه چهار متد مختلف `indexOf` در صورت پیدا نکردن کاراکتر یا `String` مورد نظر در یک `String` مشخص مقدار

۱- را برای ما برمیگردانند.

Example:

```
package javalike;

public class IndexOfExample {
    public static void main(String args[]) {
        String str1 = new String("This is a javalike tutorial");
        String str2 = new String("java");
        String str3 = new String("like");
        String str4 = new String("likes");
        System.out.println("Index of j in str1: " + str1.indexOf('j'));
        System.out.println("Index of j in str1 after 10th char:"
            + str1.indexOf('i', 10));
        System.out.println("Index of j in str1 after 30th char:"
            + str1.indexOf('j', 30));
        System.out
            .println("Index of string str2 in str1:" +
str1.indexOf(str2));
        System.out.println("Index of str2 after 15th char:"
            + str1.indexOf(str2, 15));
        System.out.println("Index of string str3:" + str1.indexOf(str3));
        System.out.println("Index of string str4:" + str1.indexOf(str4));
        System.out.println("Index of hardcoded string:" + str1.indexOf("is"));
        System.out.println("Index of hardcoded string after 4th char:"
            + str1.indexOf("is", 4));
    }
}
```

Core Java

آموزش ساده و آسان جاوا

خروجی (output):

```
Index of j in str1: 10
Index of j in str1 after 10th char:15
Index of j in str1 after 30th char:-1
Index of string str2 in str1:10
Index of str2 after 15th char:-1
Index of string str3:14
Index of string str4:-1
Index of hardcoded string:2
Index of hardcoded string after 4th char:5
```

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.