

آموزش استفاده از متد `indexOf`

متد `indexOf` برای قرار دادن یک کاراکتر یا یک رشته در داخل یک رشته ی دیگر استفاده می شود. به عنوان مثال می توانید از آن برای بررسی وجود کاراکتر `@` در یک آدرس ایمیل استفاده کنید. اجازه بدهید از این مثال در کد استفاده کنیم.

مجددا می توانید روی کد خود کامنت بگذارید یا آن را حذف کنید. اما کد جدیدی برای امتحان کردن وجود دارد:

```
package prjstrings;

public class StringManipulation {

    public static void main(String[] args) {

        char ampersand = '@';
        String email_address = "meme@me.com";

        int result;
        result = email_address.indexOf( ampersand );

        System.out.println( result );

    }
}
```

میخواهیم وجود نماد `@` را در آدرس ایمیل بررسی کنیم، بنابراین ابتدا یک متغیر `char` تنظیم کرده و به آن مقدار `'@'` اختصاص می دهیم. (به نماد علامت نقل قول متغیر `char` دقت داشته باشید). پس از تنظیم یک آدرس ایمیل، یک متغیر نتیجه داریم، این یک متغیر `int` می باشد. دلیل این که نتیجه یک عدد صحیح می باشد این است که متود `indexOf` یک مقدار گزارش خواهد داد. این مقدار عدد مربوط به موقعیت کاراکتر علامت را در رشته ی آدرس ایمیل باز خواهد گرداند. در اینجا خط مربوطه را مشاهده می کنید:

```
result = email_address.indexOf( ampersand );
```

رشته ای که در حال جستجوی آن هستید در ابتدا قرار می گیرد. پس از یک نقطه، `indexOf` را تایپ کنید. بین پرانتزهای `indexOf` گزینه های زیادی دارید: یکی از این گزینه ها تایپ کردن یک کاراکتر مجزا (یا نام متغیر `char` می باشد. ما متغیر علامت خود را `(ampersand)` بین پرانتزهای `indexOf` قرار داده ایم. جاوا موقعیت کاراکتر `@` را در آدرس ایمیل به ما خواهد گفت. این مقدار را در متغیر نتیجه ذخیره خواهد کرد.

وقتی کد را اجرا می کنید خروجی 4 خواهد بود. ممکن است تصور کنید که علامت `@` پنجمین کاراکتر در آدرس ایمیل می باشد، اما `indexOf` شمارش را از 0 شروع می کند.

آدرس آموزشگاه : تهران - خیابان شریعتی - بالا تر از خیابان ملک - جنب بانک صادرات - پلاک 651 طبقه دوم - واحد 7

88146323 - 88446780 - 88146330

به هر حال اگر کاراکتر در لغتی که آن را جستجو می کنید وجود نداشته باشد، indexOf مقدار 1- را گزارش می دهد. برای امتحان این مورد نماد @ را از آدرس ایمیل خود حذف کنید. سپس مجددا کد خود را اجرا کنید که 1- را به عنوان خروجی مشاهده خواهید کرد.

می توانید از مقدار گزارش شده ی 1- به نفع خود استفاده کنید. در اینجا مجددا کدی را مشاهده می کنید با یک عبارت IF که مقدار متغیر نتیجه را امتحان می کند:

```
package prjstrings;

public class StringManipulation {

    public static void main(String[] args) {

        char ampersand = '@';
        String email_address = "mememe.com";

        int result;
        result = email_address.indexOf( ampersand );

        if (result== -1 ) {
            System.out.println( "Invalid Email Address" );
        }
        else {
            System.out.println( "Email Address OK" );
        }
    }
}
```

بنابراین اگر نتیجه ی indexOf عدد 1- باشد، می توانیم به یوزر اجازه ی ادامه ی کار بدهیم.

می توانید از indexOf برای تست کردن بیشتر از یک کاراکتر استفاده کنید. کد زیر آدرس ایمیل را چک می کند اگر با "com" تمام می شود:

```

public static void main(String[] args) {

    String dotCom = ".com";
    String email_address = "meme@me.com";

    int result;
    result = email_address.indexOf( dotCom );

    if (result== -1 ) {
        System.out.println( "Invalid Email Address" );
    }
    else {
        System.out.println( "Email Address OK " );
    }
}

```

کد تقریباً یکسان می باشد، به جز اینکه اکنون در حال استفاده از متغیر String برای حفظ متنی هستیم که می خواهیم برای وجود (.com) و نه متغیر char آن را بررسی کنیم.

اگر متنی که در جستجوی آن هستیم در string که قبل از نقطه ی مربوط به indexOf یافت نشد، مجدداً نتیجه ی 1- گزارش داده خواهد شد. در غیر این صورت indexOf موقعیت اولین کاراکتر هماهنگ سازی را گزارش خواهد داد. در کد فوق وقتی شمارش را از 0 شروع می کنید، نقطه هفتمین کاراکتر در آدرس ایمیل می باشد.

می توانید یک موقعیت شروع نیز برای جستجوهای خود تعیین کنید. در مثال آدرس ایمیل می توانیم جستجو برای ".com" را پس از نماد @ آغاز کنیم. در اینجا کدی را می بینید که موقعیت نماد @ را تعیین می کند و سپس از آن به عنوان موقعیت آغازین برای جستجوی ".com" استفاده می کند.

آموزشگاه کلیکر داده ها

```

public static void main(String[] args) {

    char ampersand = '@';
    String dotCom = ".com";
    String email_address = "meme@me.com";

    int atPos = email_address.indexOf( ampersand );
    int result = email_address.indexOf(dotCom, atPos);

    if (result== -1 ) {
        System.out.println( "Invalid Email Address" );
    }
    else {
        System.out.println( "Email Address OK " + result );
    }
}

```

خط زیر در واقع خط جدید کد می باشد:

```
result = email_address.indexOf( dotCom, atPos );
```

تنها تفاوت افزودن یک متغیر اضافه بین پرانتزهای `indexOf` می باشد. ما هنوز رشته ای داریم که در جستجوی آن هستیم (که متنی است که در متغیر `dotcom` قرار می گیرد)، اما اکنون یک موقعیت ابتدایی برای جستجو داریم. این موقعیت مقدار متغیری به نام `atPos` می باشد. مقدار `atPos` با استفاده از `indexOf` برای تعیین موقعیت نماد `@` در آدرس ایمیل به دست می آید. بنابراین جاوا به جای شروع از 0، جستجو را از این موقعیت آغاز خواهد کرد که پیش فرض می باشد.

Ends With ... Starts With

برای برنامه ی فوق می توانید از متود داخلی `endsWith` استفاده کنید:

```
Boolean ending = email_address.endsWith( dotcom );
```

لازم است یک متغیر Boolean برای `endsWith` تنظیم کنید، زیرا متود پاسخ `true` یا `false` را گزارش خواهد داد. رشته ای که سعی در تست کردن آن دارید، بین پرانتزهای `endsWith` قرار می گیرد و متنی که در جستجوی آن هستید قبل از آن می آید. اگر متن در جستجوی رشته باشد، یک مقدار `true` گزارش داده می شود، در غیر اینصورت `false` خواهد بود. برای بررسی مقدار می توانید یک عبارت `if ... else` اضافه کنید:

```

System.out.println( "Invalid Email Address" );
}
else {
System.out.println( "Email Address OK " );
}

if (ending == false ) {

```

آدرس آموزشگاه : تهران - خیابان شریعتی - بالا تر از خیابان ملک - جنب بانک صادرات - پلاک 651 طبقه دوم - واحد 7

88146323 - 88446780 - 88146330

متود startsWith به روشی مشابه استفاده می شود.

```
Boolean startVal = email_address.startsWith( dotcom );
```

مجددا مقدار گزارش شده یک مقدار true یا false از Boolean می باشد.

در بخش بعدی چگونگی استفاده از متودی به نام substring را فرا خواهید گرفت .



آدرس آموزشگاه : تهران - خیابان شریعتی - بالا تر از خیابان ملک - جنب بانک صادرات - پلاک 651 طبقه دوم - واحد 7

88146323 - 88446780 - 88146330

<http://www.tahlildadeh.com/>