

آموزش متد Substring

یک متد واقعا مفید برای شما متود `substring` نامیده می شود. این متود به شما اجازه می دهد تا یک متن را از متن دیگر جدا سازید. برای مثال در برنامه ی فوق در آدرس ایمیل خود می توانیم پنج کاراکتر آخر را از کل آدرس جدا کرده و بررسی کنیم که آیا `co.uk` است یا نه.

برای تمرین با `substring` می خواهیم دو حرف اول از نام خانوادگی را تغییر دهیم و آنها را با دو حرف اول یک نام شخصی جابه جا سازیم و بالعکس. بنابراین اگر این نام را داشته باشیم:

"Bill Gates"

حروف "Ga" از "Gates" را با "Bi" از "Bill" جابه جا سازیم تا لغت "Bites" را بسازیم. بنابراین "Bi" از "Bill" با "Ga" از "Gates" جابه جا خواهد شد تا لغت "Gall" را بسازد. نام جدید چاپ شده "Gall Bites" خواهد بود.

ما از `substring` در قسمت بیشتر این برنامه استفاده خواهیم کرد `Substring`. به این صورت کار می کند:

```
String FullName = "Bill Gates";  
String FirstNameChars = "";
```

```
FirstNameChars = FullName.substring( 0, 2 );
```

در مورد "Bill Gates" برای جستجو یک رشته تنظیم می کنید. رشته ای که سعی در جستجوی آن دارید، بعد از یک علامت تساوی قرار می گیرد. بعد از یک نقطه (`dot`)، نام متود، `substring`، را تایپ کنید. دو روش برای استفاده از `substring` وجود دارد و تفاوت در اعداد بین پرانتزها می باشد. در کد فوق دو عدد 0 و 2 را داریم. این به این معناست که گرفتن کاراکترها در موقعیت 0 در رشته آغاز می شود و وقتی دو عدد دارید، این فرایند متوقف می شود. این دو کاراکتر بازگردانده می شوند و در متغیر `FirstNameChars` قرار می گیرند. اگر می خواهید درست در انتهای رشته قرار بگیرید، کفایت کار زیر را انجام دهید:

```
String test = FullName.substring( 2 );
```

این بار بین پرانتزهای `substring` تنها یک عدد داریم. اکنون جاوا از کاراکتر دو در رشته ی `FirstName` آغاز خواهد کرد و سپس کاراکترها را از موقعیت 2 درست در انتهای رشته می گیرد.

یک برنامه ی جدید آغاز کنید تا این برنامه را امتحان کنید. یک خط چاپ به انتهای کد خود اضافه کرده و کد شما نیز باید مانند زیر باشد:

```

package nameswapper;

public class NameSwap {

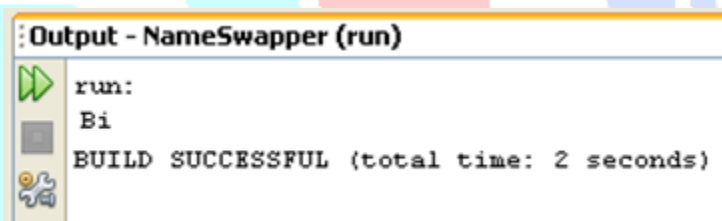
    public static void main(String[] args) {

        String FullName = "Bill Gates";
        String FirstNameChars = "";

        FirstNameChars = FullName.substring(0, 2);
        System.out.println(FirstNameChars);
    }
}

```

وقتی که برنامه اجرا می شود، پنجره ی Output باید شبیه به تصویر زیر باشد:



بنابراین متود substring به ما اجازه ی گرفتن دو کاراکتر اول نام "Bill" را داده است.

برای گرفتن دو کاراکتر اول، یک 0 و یک 2 بین پرانتزهای substring داشتیم. ممکن است تصور کنید برای گرفتن "Ga" از "Gates" می توانستیم این کار را انجام دهیم:

```
= FullName.substring(5, 2);
```

بعد از همه ی اینها هنوز دو کاراکتر می خواهیم. فقط این بار 5 به جاوا می گوید که از "G" در "Gates" آغاز کنید. (اولین موقعیت در یک رشته موقعیت 0 می باشد و نه 1). بنابراین از موقعیت 5 در این رشته آغاز کرده و دو کاراکتر را بگیرید.

به هر حال اجرای آن کد منجر به بروز خطا خواهد شد. این به این خاطر است که دومین عدد بین پرانتزهای substring به معنای تعداد کاراکترهایی که می خواهید بگیرید نیست، بلکه به معنای موقعیت نهایی در رشته می باشد. با مشخص کردن 2، به جاوا می گوئیم که در کاراکتری در موقعیت 2 از رشته به اتمام برسد. از آنجایی که نمی توانید از موقعیت 6 به عقب بازگشته و به موقعیه 2 بروید، خطایی دریافت می کنید.

آدرس آموزشگاه : تهران - خیابان شریعتی - بالا تر از خیابان ملک - جنب بانک صادرات - پلاک 651 طبقه دوم - واحد7

88146323 - 88446780 - 88146330

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

(نکته: اگر شمارش را در رشته ی "Bill" از 0 شروع کنید، ممکن است تصور کنید که موقعیت 2 حرف "l" می باشد و حدس شما درست است. اما substring قبل از کاراکتر در آن موقعیت آغاز می کند و پس از آن).

بنابراین برای گرفتن "Ga" از "Gates" می توانید مانند زیر عمل کنید:

```
FullName.substring( 5, FullName.length( ) - 3 );
```

اکنون دومین عدد طول رشته منهای 3 کاراکتر می باشد. طول رشته تعداد کاراکترهایی است که آن رشته دارد "Bill". "Gates" دارای 10 کاراکتر شامل فاصله، می باشد. اگر سه کاراکتر را کنار بگذارید، 7 کاراکتر خواهید داشت. بنابراین ما به substring می گوئیم که در کارامتر 5 شروع شده و در کاراکتر 7 به پایان برسد.

این برنامه برای افرادی با نام "Bill Gates" به درستی کار می کند. اما اگر نام شما "Billy Gates" می بود، این برنامه به درستی کار نمی کرد. بنابراین کد فوق کاراکتر فاصله به علاوه ی حرف "G" را می گیرد که به هیچ وجه کاراکترهای مورد نظر ما نیستند. ما می خواهیم که برنامه با هر دو نامی که وارد می شوند، کار کند. بنابراین باید کمی زیرک تر باشیم.

کاری که می توانیم انجام دهیم این است که به موقعیت فاصله در هر دو نام دقت داشته باشیم. دو کاراکتری که می خواهیم از نام دوم بگیریم همیشه درست بعد از کاراکتر فاصله قرار می گیرند. ما به کدی احتیاج داریم که دو کاراکتر اول بعد از فاصله را بگیرد.

برای اشاره به موقعیت فاصله می توانیم از indexOf استفاده کنیم.

```
int spacePos = FullName.indexOf(" ");
```

برای تعیین کاراکتر فاصله، می توانید یک فاصله بین دو نماد نقل قول ("") قرار دهید. این کاراکتر پس از پرانتزهای indexOf قرار می گیرد. مقدار بازگشت داده شده یک عدد صحیح خواهد بود که موقعیت اولین رویداد از کاراکتر فاصله در رشته ی FullName می باشد.

این تمرین را می توانید با اضافه کردن خط فوق به کد خود امتحان کنید: برای چک کردن Output یک خط چاپی اضافه کنید:

آموزشگاه تحلیگر داده ها

```

public static void main(String[] args) {

    String FullName = "Bill Gates";
    String FirstNameChars = "";

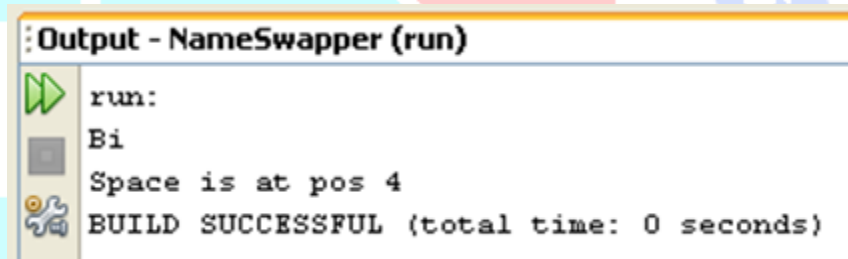
    FirstNameChars = FullName.substring(0, 2);
    System.out.println( FirstNameChars );

    int spacePos = FullName.indexOf(" ");
    System.out.println( "Space is at pos " + spacePos);

}

```

برنامه را اجرا کنید تا خروجی زیر را مشاهده کنید:



```

Output - NameSwapper (run)
run:
Bi
Space is at pos 4
BUILD SUCCESSFUL (total time: 0 seconds)

```

بنابراین در این رشته فاصله در موقعیت 4 می باشد. ما می توانیم از این واقعیت برای گرفتن دو کاراکتر اول از "Gates" یا در واقع هر نام دوم دیگری استفاده کنیم. به جاوا می گوئیم که از اولین کاراکتر بعد از فاصله آغاز و در دو کاراکتر بعدی به اتمام برساند.

```
FullName.substring( spacePos + 1, (spacePos + 1) + 2)
```

بنابراین دو عدد بین پرانتزهای substring عبارتند از:

```
spacePos + 1, (spacePos + 1) + 2
```

ما می خواهیم که از اولین کاراکتر بعد از فاصله آغاز کنیم (space + 1) و دو کاراکتر بعد از این موقعیت به اتمام برسانیم که به این شکل می باشد $(spacePos + 1) + 2$:

خطوط زیر را به کد خود اضافه کنید (خطوطی که هایلایت شده اند). متود جدید `substring` در دو خط قرار می گیرد، اما شما اگر تمایل داشته باشید می توانید روی یک خط قرار دهید:).

```

public static void main(String[] args) {

    String FullName = "Bill Gates";
    String FirstNameChars = "";

    FirstNameChars = FullName.substring(0, 2);
    System.out.println( FirstNameChars );

    int spacePos = FullName.indexOf(" ");
    System.out.println( "Space is at pos " + spacePos);

    String SurNameChars = "";
    SurNameChars = FullName.substring(spacePos + 1,
                                      (spacePos + 1) + 2);

    System.out.println( SurNameChars );

}

```

وقتی که برنامه را اجرا می کنید، پنجره ی Output مانند زیر خواهید بود:

```

Output - NameSwapper (run)
run:
Bi
Space is at pos 4
Ga
BUILD SUCCESSFUL (total time: 1 second)

```

بنابراین اکنون ما "Bi" از Bill و "Ga" از Gates را داریم. آنچه اکنون باید انجام دهیم گرفتن بقیه ی کاراکترها از دو نام و سپس جابه جایی آنها می باشد.

مجددا می توانیم از substring برای گرفتن بقیه ی کاراکترها از نام اول استفاده کنیم:

```

String OtherFirstChars = "";
OtherFirstChars = FullName.substring( 2, spacePos );
System.out.println( OtherFirstChars );

```

و همچنین کاراکترهای باقیمانده از نام دوم:

```

String OtherSurNameChars = "";
OtherSurNameChars = FullName.substring((spacePos + 1) + 2,

```

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

88146323 - 88446780 - 88146330

```
FullName.length() );  
System.out.println( OtherSurNameChars );
```

به اعداد بین پرانتزهای substring دقت کنید. برای گرفتن دیگر کاراکترهای نام اول، اعداد عبارتند از:

```
2, spacePos
```

این به جاوا می گوید که از موقعیت 2 شروع کرده و درست به موقعیت فاصله برود. به هر حال گرفتن بقیه ی نام دوم کمی دشوارتر خواهد بود:

```
(spacePos + 1) + 2, FullName.length( )
```

$(spacePos + 1) + 2$ موقعیت شروع سومین کاراکتر از دومین نام می باشد. ما می خواهیم که در طول رشته به پایان برسیم که بقیه ی کاراکترها را دریافت خواهیم کرد.

خطوط های لایت شده ی زیر را به کد خود اضافه کنید:



آموزشگاه تحلیکر داده ها

```

public static void main(String[] args) {

    String FullName = "Bill Gates";
    String FirstNameChars = "";

    FirstNameChars = FullName.substring(0, 2);
    System.out.println( FirstNameChars );

    int spacePos = FullName.indexOf(" ");
    System.out.println( "Space is at pos " + spacePos);

    String SurNameChars = "";
    SurNameChars = FullName.substring(spacePos + 1,
                                     (spacePos + 1) + 2);

    System.out.println( SurNameChars );

    String OtherFirstChars = "";
    OtherFirstChars = FullName.substring(2, spacePos);
    System.out.println( "other first: " + OtherFirstChars );

    String OtherSurNameChars = "";
    OtherSurNameChars = FullName.substring((spacePos + 1) + 2,
                                          FullName.length());

    System.out.println("other second: " + OtherSurNameChars );

}

```

خروجی مانند زیر خواهید بود:

```

:Output - NameSwapper (run)
run:
Bi
Space is at pos 4
Ga
other first: ll
other second: tes
BUILD SUCCESSFUL (total time: 2 seconds)

```

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/>

اکنون تمام قسمت های نام را داریم. برای اتصال آنها به یکدیگر، می توانیم از الحاق (concatenation) استفاده کنیم:

```
String NewName = "";  
NewName = SurNameChars + OtherFirstChars + " " +  
           FirstNameChars + OtherSurNameChars;  
  
System.out.println( "New Name = " + NewName);
```

Output - NameSwapper (run)

```
run:  
Bi  
Space is at pos 4  
Ga  
other first: ll  
other second: tes  
New Name = Gall Bites  
BUILD SUCCESSFUL (total time: 2 seconds)
```

می توانیم از خطوط چاپی رها شویم و از یک یوزر بخواهیم که نام اول و نام دوم را وارد کند. در اینجا برنامه ی جدید را مشاهده می کنید (تنها افزایش برای ورودی صفحه کلید می باشد که قبلا از آن استفاده کرده اید:)

آموزشگاه تحلیکیر داده ها

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/>


```

package nameswapper;
import java.util.Scanner;

public class NameSwap {

    public static void main(String[] args) {

        Scanner user_input = new Scanner(System.in);

        System.out.println( "Enter a First Name and Surname:" );
        String FullName = user_input.nextLine();

        String FirstNameChars = "";
        FirstNameChars = FullName.substring(0, 2);

        int spacePos = FullName.indexOf(" ");
        String SurNameChars = "";
        SurNameChars = FullName.substring(spacePos + 1,
                                         (spacePos + 1) + 2);

        String OtherFirstChars = "";
        OtherFirstChars = FullName.substring(2, spacePos);

        String OtherSurNameChars = "";
        OtherSurNameChars = FullName.substring((spacePos + 1) + 2,
                                              FullName.length());

        String NewName = "";
        NewName = SurNameChars + OtherFirstChars + " " +
                 FirstNameChars + OtherSurNameChars;
        System.out.println( "You are: " + NewName);
    }
}

```

وقتی برنامه را اجرا می کنید و نام و نام خانوادگی را وارد می کنید، پنجره ی Output باید مانند تصویر زیر باشد:

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/>

```
Output - NameSwapper (run)
run:
Enter a First Name and Surname:
Kenny Carney
You are: Canny Kerney
BUILD SUCCESSFUL (total time: 9 seconds)
```

البته باید چند برنامه‌ی بررسی خطا نیز اضافه کنیم. اما خواهیم پذیرفت که یوزر می‌تواند یک نام و یک نام خانوادگی با یک فاصله بین آنها وارد کند. در غیر اینصورت برنامه با شکست مواجه می‌شود. به هر حال اجازه بدهید ادامه داده و نگاهی به متود equals داشته باشیم.



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/>