

# آموزش آرایه ها در جاوا

اگر می خواهید که به طور موثر کد گذاری کنید، مفهومی که در برنامه نویسی باید به آن عادت کنید array می باشد. در این بخش در مورد array ها و چگونگی استفاده از آنها فراخواهید گرفت.

## یک Array چیست؟

تاکنون با متغیرهایی کار کردید که فقط یک مقدار را در خود حفظ می کنند. متغیرهای صحیح که تنظیم کرده اید، فقط یک عدد را حفظ می کنند و متغیرهای رشته نیز فقط یک رشته از متن را در خود دارند. یک array (ردیف) روشی برای حفظ بیشتر از یک مقدار در یک زمان می باشد که در واقع شبیه لیستی از آیتم ها می باشد. یک array را می توان به عنوان ستون هایی در یک صفحه ی گسترده در نظر گرفت. شما می توانید یک صفحه ی گسترده تنها با یک ستون و یا تعداد زیادی ستون داشته باشید. داده ای که در یک ردیف مجزا حفظ می شود می تواند مشابه تصویر زیر باشد:

	Array_Values
0	10
1	14
2	36
3	27
4	43
5	18

مانند یک صفحه ی گسترده، arrayها برای هر ردیف دارای یک موقعیت عددی می باشند. این موقعیت ها در یک array از شروع شده و متداولا افزایش می یابند. هر موقعیت نیز در یک array می تواند یک مقدار در خود حفظ کند. در تصویر بالا موقعیت 0 دارای مقدار 10 و موقعیت 1 دارای 14، موقعیت 2 دارای مقدار 36 و غیره می باشد.

برای برقراری یک array از اعداد، مشابه تصویر بالا، باید به جاوا اعلام کنید که چه نوع داده ای قرار است وارد array شود ( اعداد صحیح، رشته ها، مقادیر Boolean و غیره). سپس لازم است که اعلام کنید که array دارای چند موقعیت می باشد:

```
int[ ] aryNums;
```

تنها تفاوت بین برقراری یک متغیر عدد صحیح طبیعی و یک array یک جفت کروشه ای است که بعد از نوع داده قرار می گیرد. کروشه ها برای اعلام به جاوا در مورد این که شما قصد برقراری یک array را دارید، کافی هستند. نام array مربوط به تصویر بالا aryNums می باشد. درست مانند متغیرهای طبیعی، می توانید آنها را هر چیزی که می خواهید نام گذاری کنید ( با همان استثنائاتی که قبلا ذکر کردیم).

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

88146323 - 88446780 - 88146330

اما این مورد فقط به جاوا می گوید که شما قصد تنظیم یک array عدد صحیح را دارید. این برنامه تعداد موقعیت هایی را که array باید حفظ کند، اعلام نمی کند. برای انجام این کار باید یک array object جدید تنظیم کنید:

```
aryNums = new int[6];
```

شما با نام array شروع کرده اید که با علامت تساوی دنبال می شود. پس از علامت تساوی به لغت کلیدی new و سپس مجدداً نوع داده ی خود، نیاز دارید پس از نوع داده یک جفت کروشه قرار می گیرد. بین کروشه ها نیاز به اندازه ی array دارید. اندازه در واقع تعداد موقعیت هایی است که array باید حفظ کند.

اگر تمایل دارید، می توانید همه ی آنها را روی یک قرار دهید:

```
int[ ] aryNums = new int[6];
```

بنابراین ما به جاوا اعلام می کنیم که یک array را با 6 موقعیت در آن تنظیم کند. پس از اجرای این خط، جاوا مقادیر پیش فرض را برای array اختصاص خواهد داد. از آنجایی که ما یک array مقدار صحیح تنظیم کرده ایم، مقادیر پیش فرض برای همه ی 6 موقعیت 0 خواهد بود.

برای اینکه در یک array مقادیر را به موقعیت های مختلف اختصاص دهید، این کار را به یک روش طبیعی انجام می دهید:

```
aryNums[0] = 10;
```

در اینجا مقدار 10 به موقعیت 0 در یک array به نام aryNums اختصاص داده می شود. مجدداً کروشه ها برای اشاره به هر موقعیت استفاده می شوند. اگر بخواهید مقدار 14 را به یک array با موقعیت 1 اختصاص دهید، کد مربوط مانند زیر خواهد بود:

```
aryNums[1] = 14;
```

و کد مربوط به مقدار 36 برای موقعیت 2 مانند زیر خواهد بود:

```
aryNums[2] = 36;
```

فراموش نکنید که از آنجایی که array ها از 0 شروع می شوند، سومین موقعیت در یک array دارای عدد شاخص 2 می باشد.

اگر بدانید که چه مقادیری قرار است در یک array قرار گیرند، می توانید آنها را مانند زیر تنظیم کنید:

```
int[ ] aryNums = { 1, 2, 3, 4 };
```

این متود در حال تنظیم یک array با استفاده از کروشه ها بعد از علامت تساوی، می باشد. در بین کروشه ها مقادیری را تایپ می کنید که array خواهد گرفت. اولین مقدار در موقعیت 0 و دومین مقدار در موقعیت 1 و غیره خواهد بود. توجه داشته باشید که هنوز

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

88146323 - 88446780 - 88146330

پس از int نیاز به کروشه دارید، اما نیازی به لغت کلیدی new یا تکرار نوع داده و یا کروشه ها ندارید. اما این فقط مربوط می شود به نوع داده های مقادیر int ، رشته و مقادیر char. به عبارت دیگر نیاز به لغت کلیدی new دارید. بنابراین می توانید این کار را انجام دهید:

```
String[ ] aryStrings = {"Autumn", "Spring", "Summer", "Winter" };
```

اما این کار را نمی توانید انجام دهید:

```
boolean[ ] aryBools = {false, true, false, true};
```

برای تنظیم یک boolean array هنوز نیاز به لغت کلیدی new دارید:

```
boolean[ ] aryBools = new boolean[ ] {false, true, false, true};
```

برای به دست آوردن مقادیر مربوط به array ، نام array را تایپ کنید که با موقعیت array در یک کروشه دنبال می شود. مانند زیر:

```
System.out.println( aryNums[2] );
```

کد بالا هر مقداری که در ردیفی با موقعیت 2 در یک array به نام aryNums باشد را چاپ می کند. اما اجازه بدهید کمی کدگذاری تمرین کنیم.

یک پروژه ی جدید آغاز کرده و آن را به میل خود نامگذاری کنید. فراموش نکنید که نام گروه را به یک مورد واضح تغییر دهید.

کد زیر را در متود جدید Main تایپ کنید:

آموزشگاه تحلیکرو داده ها

```

package prjarrays;

public class ArraysTest {

    public static void main(String[] args) {

        int[] aryNums;

        aryNums = new int[6];

        aryNums[0] = 10;
        aryNums[1] = 14;
        aryNums[2] = 36;
        aryNums[3] = 27;
        aryNums[4] = 43;
        aryNums[5] = 18;

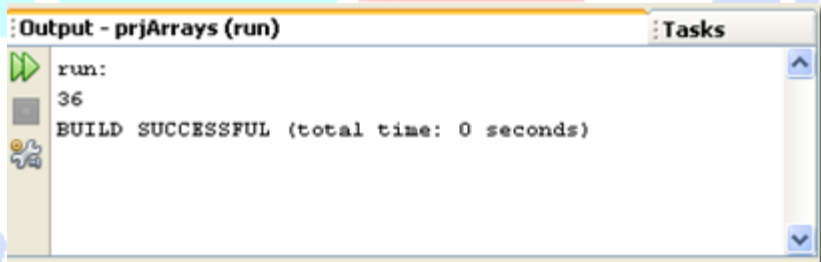
        System.out.println( aryNums[2] );

    }

}

```

وقتی برنامه را اجرا می کنید باید تصویر زیر را در پنجره ی Output مشاهده کنید:



موقعیت عددی array را در print line از 2 به 5 تغییر دهید و در عوض 18 باید چاپ شود. در بخش بعدی نگاهی به چگونگی استفاده از array ها با loop ها خواهیم انداخت.



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/>