

آموزش زبان برنامه نویسی جاوا

متغیرها

جلسه پنجم

نویسنده: رحمان زارعی

جاوا را ساده، آسان و شیرین بنوشید!!!!



- متغیرها برای ذخیره داده‌ها استفاده می‌شوند.
- متغیرها وقتی تعریف می‌شوند محلی از حافظه برای ذخیره مقدارشون اختصاص داده می‌شود.
- در سیستم عامل انواع مختلفی از متغیرها حافظه بهشون اختصاص داده می‌شود.
- هر متغیر یک نام دارد که به دلخواه انتخاب می‌شود و یک نوع که از انواع متغیرها با توجه به نیاز تعیین می‌شود.
- برای تعریف متغیر ابتدا نوع متغیر را مشخص کرده و سپس نامی به دلخواه برایش انتخاب می‌کنیم.

```
VariableType VariableName;
```

- نام متغیر منحصر به فرد است یعنی نمی‌توان همزمان دو متغیر همنام باشند.
- ما می‌توانیم متغیرها را در انواع مختلفی نظیر `integer`, `character`, `String` و... در حافظه ذخیره کنیم.

کل متغیرها در جاوا به دو بخش تقسیم می‌شود:

۱. متغیرهای Primitive (اولیه)
۲. متغیرهای Reference/Object (شی/ارجاعی)

متغیرهای Primitive (اولیه):

- ۸ نوع متغیر `primitive` (اولیه) در جاوا وجود دارد.
- انواع متغیر `primitive` (اولیه) یک کلمه کلیدی (`keyword`) دارند که توسط آن تعریف می‌شوند.

در پایین به جزییات هشت نوع متغیر primitive (اولیه) می پردازیم:

متغیر byte:

- این متغیر از نوع عدد صحیح می باشد یعنی تنها برای ذخیره اعداد صحیح در حافظه مثل ۱۰ و ۵۰ بکار می رود.
- حداقل مقداری که این متغیر می تواند بگیرد -۱۲۷ می باشد.
- حداکثر مقداری که این متغیر می تواند بگیرد ۱۲۷ می باشد.
- مقدار پیشفرض (Default value) این متغیر پس از تعریف 0 (صفر) می باشد.
- بایت ها (byte) اساسا برای ذخیره اعداد صحیح خیلی کوچک که در محدوده -۱۲۷ تا ۱۲۷ هستند استفاده می شوند.

مثال:

```
byte a=120;
byte b=-80;
```

متغیر short:

- این متغیر نیز از نوع عدد صحیح می باشد یعنی تنها برای ذخیره اعداد صحیح در حافظه بکار می رود.
- حداقل مقداری که این متغیر می تواند بگیرد -32,768 می باشد.
- حداکثر مقداری که این متغیر می تواند بگیرد 32,767 می باشد.
- مقدار پیشفرض (Default value) این متغیر پس از تعریف 0 (صفر) می باشد.
- متغیرهای short اساسا برای ذخیره اعداد صحیح کوچک که در محدوده -32,768 تا 32,767 هستند استفاده می شوند.

مثال:

```
short s = 10000;
short r = -20000;
```

متغیر int:

- این متغیر نیز از نوع عدد صحیح می باشد یعنی تنها برای ذخیره اعداد صحیح در حافظه بکار می رود.
- حداقل مقداری که این متغیر می تواند بگیرد -2,147,483,648 می باشد.

- حداکثر مقداری که این متغیر می تواند بگیرد 2,147,483,647 می باشد.
- مقدار پیشفرض (Default value) این متغیر پس از تعریف 0 (صفر) می باشد.
- متغیرهای int اساسا برای ذخیره اعداد صحیح بزرگ که در محدوده -2,147,483,648 تا 2,147,483,647 هستند استفاده می شوند.
- مثال:

```
int a = 100000;  
int b = -200000;
```

متغیر long:

- این متغیر نیز از نوع عدد صحیح می باشد یعنی تنها برای برای ذخیره اعداد صحیح در حافظه بکار می رود.
- حداقل مقداری که این متغیر می تواند بگیرد -9,223,372,036,854,775,808 می باشد.
- حداکثر مقداری که این متغیر می تواند بگیرد 9,223,372,036,854,775,807 می باشد.
- مقدار پیشفرض (Default value) این متغیر پس از تعریف 0 (صفر) می باشد.
- متغیرهای long اساسا برای ذخیره اعداد صحیح خیلی بزرگ که در محدوده -9,223,372,036,854,775,808 تا 9,223,372,036,854,775,807 هستند استفاده می شوند.
- مثال:

```
long a = 100000;  
long b = -200000;
```

متغیر float:

- این متغیر نیز از نوع عدد اعشاری می باشد یعنی تنها برای برای ذخیره اعداد اعشاری در حافظه بکار می رود.
- مقدار پیشفرض (Default value) این متغیر پس از تعریف 0.0f می باشد.

مثال:

```
float f1 = 234.5f;
```

نکته ۱: حتما هنگام مقدار دهی به متغیر float که از نوع اعشاری می باشد آخرش از حرف انگلیسی f کوچک باید استفاده شود.

نکته ۲:

- هنگام تعریف متغیر، نام متغیر تنها می تواند با حروف یا همون کاراکترهای انگلیسی (a-z) یا _ (آندرلاین) یا حرف دلار (\$) شروع شود.
 - بجای کاراکتر انگلیسی می توانیم از کاراکترهای سایر زبان ها نظیر فارسی هم استفاده کنیم اما اصلا توصیه نمی شود.
 - نمی توان نام متغیر رو با عدد شروع کرد. اما میشود بعد از این که ابتدای نام متغیر رو با حرف انگلیسی یا _ (آندرلاین) یا حرف دلار (\$) شروع کردیم **بعدش از عدد** استفاده کنیم.
 - در تعریف نام متغیر نمی توان از حروفی نظیر @،#،%،-،!،+،&،^ و... استفاده کرد.
 - در تعریف نام متغیر نمی توان از نقطه استفاده کرد.
 - در تعریف نام متغیر نمی توان میان کاراکترهای نام متغیر فاصله انداخت.
 - در تعیین نام متغیر نمی توان از کلمات کلیدی جاوا (Java keyword) به تنهایی نظیر public,static,int,void,long,float و... استفاده کرد،مگر این که کنار این کلمات کلیدی از عدد یا حروف دیگری استفاده شود نظیر public1 یا int145 یا voida و...
- مثال ۱:** در زیر انواع شکل صحیح تعیین نام متغیر رو مشاهده می کنید:

```
float f1 = 234.5f;
float f568 = 234.5f;
float _f = 234.5f;
float _fkkk5j = 234.5f;
float f_2000 = 234.5f;
float int87k = 234.5f;
float public1 = 234.5f;
float $=5.1f;
float $250=8.8f;
float _4=7.36f;
```

مثال ۱: در زیر انواع شکل اشتباه تعیین نام متغیر رو مشاهده می کنید!!!!

```
float 1f = 234.5f;
float int = 234.5f;
float void = 234.5f;
float 1254= 234.5f;
float 2@ = 234.5f;
float %$3= 234.5f;
float a*& 234.5f;
float 4_=7.36f;
float a.b=13.6f;
float ali mohammadi =12.6f;
```



- هنگام نام گذاری متغیر، جاوا حساس به حروف بزرگ یا کوچک می باشد که به آن **case sensitive** می گویند، مثلا نام **JAFAR** و **jafar** پیش ما یکسان هستند اما در جاوا کاملا با هم فرق دارند. یا دو متغیر **treeCount** و **TreeCount** کاملا با هم فرق دارند. پس اگر دو متغیر همانام بودند اما در حروف بزرگ یا کوچک متفاوت بودند را **case sensitive** می گوئیم. دو نام که حالت **case sensitive** نسبت به هم داشتند یعنی تنها در حروف بزرگ یا کوچک متفاوت بودند را می توانیم به عنوان نام متغیرها استفاده کنیم.
- بهتر است نام متغیر با حرف کوچک شروع شود.

متغیر double:

- این متغیر نیز از نوع عدد اعشاری می باشد یعنی تنها برای برای ذخیره اعداد اعشاری در حافظه بکار می رود.
- مقدار پیشفرض (Default value) این متغیر پس از تعریف **0.0d** می باشد که می توان از بکار بردن حرف **d** آخر مقدار صرف نظر کرد.
- بیشتر برای تعریف متغیر از نوع اعشاری از این متغیر استفاده میکنند. پس ما **float** رو تقریبا فراموش میکنیم و هنگام تعریف متغیر از نوع اعشاری از متغیر **double** استفاده میکنیم.

مثال:

```
double d1 = 123.4;
```

متغیر boolean:

- این متغیر یک متغیر منطقی هست.

- برای این متغیر تنها دو مقدار وجود دارد : true یا false
- این نوع متغیر برای دستورهایی که درست هستند یا غلط و flag ها و شرط ها استفاده می شود. میدونم معلوم نیست چی به چی شد!!!! پس با ی مثال مفهوم این متغیر رو جا میندازم:
فرض کنید یک متغیر به نام flag داریم که مقدارش true است حال شرط زیر رو مشاهده می کنیم:
اگر flag=true است پیام "Hello iran" رو چاپ کن
اگر false=flag هست پیام "Error" رو چاپ کن.
یا مثال دیگر:
امروز هوا "آفتابی" هست. در اینجا آفتابی رو true و بارانی رو false در نظر می گیریم.
اگر امروز هوا "آفتابی" باشد: علی به خرید می رود. علی به خرید می رود چون هوای امروز آفتابی هست و شرط درست(true) است.
اگر امروز هوا "بارانی" باشد: علی در خانه می ماند. علی در خانه نمی ماند چون هوای امروز آفتابی هست و شرط نادرست (false) است.
*در مورد دستورات شرطی و... در جلسات آینده صحبت خواهیم کرد فعلا در همین حد بدونید در آینده در خلال مثال های کاربردی بیشتر با این مفاهیم آشنا خواهیم شد.
- مقدار پیشفرض (Default value) این متغیر پس از تعریف false می باشد.
مثال:

```
boolean one = true;
```

متغیر char:

- این متغیر برای تعریف کارکتر به کار می رود
- هر حرف یک کارکتر حساب می شود فرق نمی کند یک حرف انگلیسی باشد یا یک علامت ریاضی و...
- حرف یا کارکتر رو هنگام تعریف بین دو علامت ' ' قرار می دهیم:

```
char ch='a';  
char ch1='$';
```

هر حرف که یک کارکتر حساب میشود یک معادل عدد یا شماره منحصر به فرد دارد!!

مثال: فرض کنید حرف a یک کارکتر هست شماره حرف یا کارکتر a برابر است با ۹۷ یعنی می توانیم بجای این که یک کارکتر تعریف کنیم و مقدار آن رو 'a' قرار دهیم می توانیم بعد از تعریف کارکتر شماره حرف a که ۹۷ هست رو بگذاریم:

این دو تعریف زیر هیچ فرقی با هم نمیکنند و هر دو دربردارنده حرف یا کارکتر a هستند تنها یکی حرف a بهش داده شده و دیگری شماره حرف a:

```
char ch='a';
char ch1=97;
```

*جالب است بدانید تمام شماره های کارکتر ها منحصر به فرد حتی شماره حروف بزرگ و کوچک هم باهم فرق می کنند.
کارکتر ها علاوه بر حروف انگلیسی شامل حروفی نظیر ?,/,.,=,+,-,,^,%,\$,#,@,!, و... هم می شوند.

- حداقل مقداری که این متغیر می تواند بگیرد بصورت زیر است:

```
0 or '\u0000'
```

- حداکثر مقداری که این متغیر می تواند بگیرد بصورت زیر است:

```
'\uffff' or 65,535
```

- مقدار پیشفرض (Default value) این متغیر پس از تعریف 0 یا ' ' می باشد.
- متغیر char برای ذخیره هر کارکتری استفاده میشود.

مثال:

```
char letterA = 'A';
```

متغیر های Reference (ارجاعی):

فعلا زوده در مورد این نوع متغیر ها صحبت کنیم به امید خدا در جلسات آینده بهش می پردازیم.

خب تا این جا درمورد توضیح متغیر ها نگه می داریم و میریم سراغ مثال و تمرین به نظر شما خودم تنها برنامه نویسی رو با مثال و تمرین میشه یاد گرفت و تا زمانی که ادم عملا کد نزنه و خودش رو به چالش نکشه هزاران صفحه در مورد برنامه نویسی بخونه فایده نداره!!!!!!!!!!!!!!

مطمئنا سایر جزییات و مفاهیم در خلال تمرین و مثال بهتر قابل فهم می باشد.

- فرض بر این است که شما با توجه به جلسات قبل با طریقه ایجاد پروژه و پکیج و کلاس و متد main آشنا شده اید.

مثال ۱: برنامه ای به زبان جاوا بنویسید که دارای یک متغیر با مقدار صحیح ۱۰۰۰ و یک عدد اعشاری با مقدار ۵/۰۰۴ باشد و مقدار این دو متغیر در خروجی جداگانه چاپ شود.

توضیحات:

- خب در نگاه اول باید دو متغیر تعریف کنیم یکی نوعش صحیح است و دیگری نوعش اعشار
- در نگاه دوم گفته مقدار دهی بهشون کنیم که مقدار متغیر صحیح ۸۰۰۰۰ است میایم نگاه محدوده متغیر های صحیح میکنیم و بررسی میکنیم که ۱۰۰۰ در کدام محدوده متغیر های عدد صحیح قرار میگیرد و ازش استفاده می کنیم. با توجه به محدوده متغیر های صحیح می توان از نوع های صحیح زیر استفاده کرد:

- int
- long

اما بهتره که از int استفاده کنیم چون وقتی ۸۰ هزار در بازه int قرار دارد چرا بیایم از یک بازه بسیار بزرگ نظیر long استفاده کنیم زیبا تر هست که در برنامه نویسی این موارد رعایت شود.

دلیل عدم استفاده از short و byte کوچک بودن این متغیر های صحیح نسبت به مقدار داده شده در صورت مسئله هست. در مورد متغیر اعشاری هم همان طور که مشخص هست از double استفاده میکنیم

- در نگاه سوم در خروجی مقدار متغیر چاپ شود همان طور که می دونید برای چاپ در خروجی در محیط کنسول جاوا از دستور `System.out.println();` (این دستور برای چاپ متن یا مقدار یک متغیر یا همزمان چاپ متن در کنار مقدار متغیر بکار می رود) استفاده می شود.

برای چاپ یک متن کفایت متن مورد نظر رو درون دو "" گذاشته و در ورودی دستو `System.out.println();` داخل پرانتز قرار بدهیم.

```
System.out.println("Hello Iran");
```

برای چاپ مقدار متغیر کفایت نام متغیر در ورودی دستو `System.out.println();` داخل پرانتز قرار گیرد.

```
System.out.println(variableName);
```

برای چاپ یک متن در کنار مقدار یک متغیر کفایت متن مورد نظر رو درون دو "" قرار داده و در کنارش نام متغیر رو قرار دهیم و با قرار دادن علامت + (بعلاوه) (plus) بین نام متغیر و متن مورد نظر که درون دو "" هست این دو رو بهم متصل میکنیم. *در کل علامت + نقش وصل کردن دو متن به هم یا دو مقدار به هم یا یک متن و مقدار در کنار هم رو دارد.

۱. نام متغیر+"متن مورد نظر"

```
System.out.println(variableName+"sample text");
```

۲. نام متغیر+نام متغیر

```
System.out.println(variableName+ variableName);
```

۳. "متن مورد نظر یک"+"متن مورد نظر دو"

```
System.out.println("+sample text1"+"sample text2");
```

خب برویم سراغ اصل مطلب و حل مسئله مون!!!!

کلاسی به نام Example1 ایجاد میکنیم.

درون متد main موجود در کلاس خواسته مسئله رو که در بالا بررسی کردیم پیاده سازی میکنیم.

سورس کد مثال ۱ بصورت زیر است:

- با کپی پیست کردن این سورس کد در eclipse برنامه تست کنید.
- یکی از بهترین راه های یاد گیری هر زبان برنامه نویسی اینه که کد رو برا خودتون امتحان کنید و با ایجاد تغییرات در کد و به چالش کشیدن خودتون مفاهیم رو یاد بگیرید.

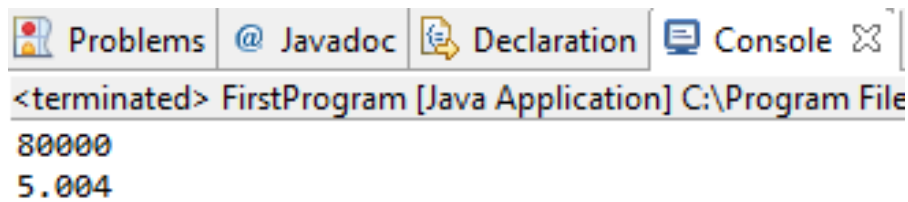
```
package iran;

public class Example1 {

    public static void main(String[] args) {
        int a=80000;
        double b=5.004;
        System.out.println(a);
        System.out.println(b);
    }
}
```

خروجی: تصویر (۱)

```
80000
5.004
```



خروجی:



تصویر (۱)

روش کپی کردن یک سورس کد در Eclipse:

راه های زیادی وجود دارد باهم یکی از این راه ها (سریع ترین راه) بررسی میکنیم:

۱ ابتدا سورس کد مورد نظر رو کپی کنید.

۲ کلیک سمت راست کرده و گزینه کپی رو انتخاب کنید.

۳ بعد در بخش فولدر پروژه رو یکی از پکیج های موجود کلیک سمت راست کرده و گزینه پیست رو بزنید. خواهید

دید که بصورت اتوماتیک کلاس شما ایجاد می شود که حاوی دستورات کپی شده می باشد. تصویر (۲) و (۳) و (۴)

فرض کنید یک سورس کد به شکل زیر داریم:

```
package iran;
```

```
public class FirstProgram {
```

```
public void main(String[] args) {
```

```
// TODO Auto-generated method stub
```

```
System.out.println("Hello Iran");
```

```
} //End method main
```

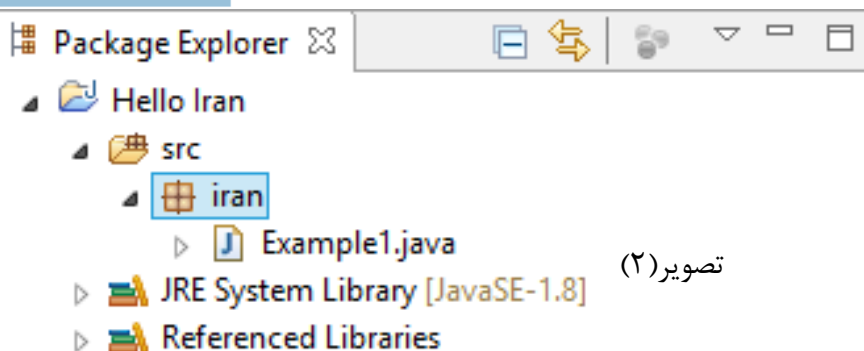
```
} //End class
```

1 انتخاب دستورات

2 کلیک سمت راست

3

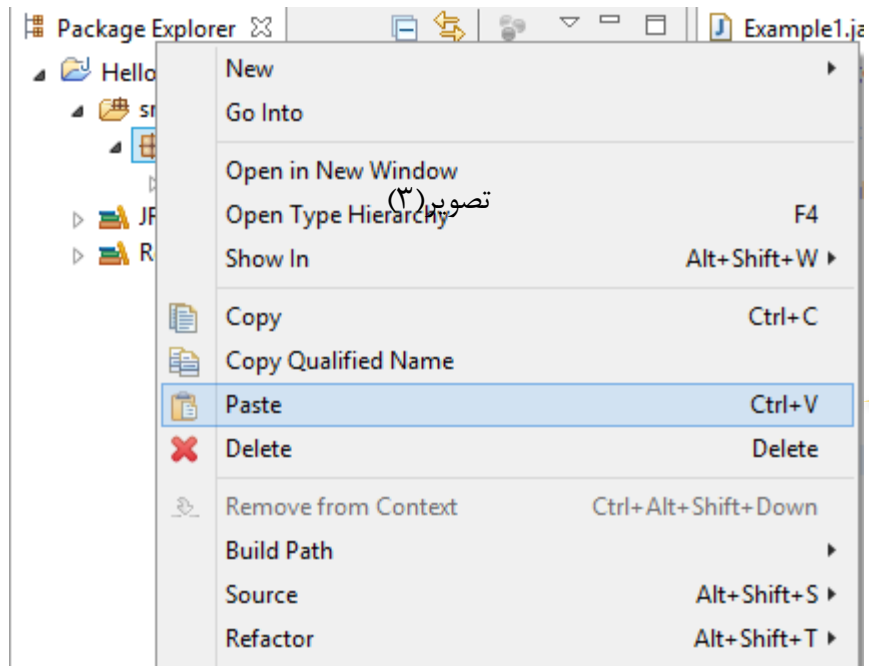
Copy Ctrl+C
Highlight Text
Strikethrough Text
Add Note to Replace Text
Add Note to Text



تصویر (۲)

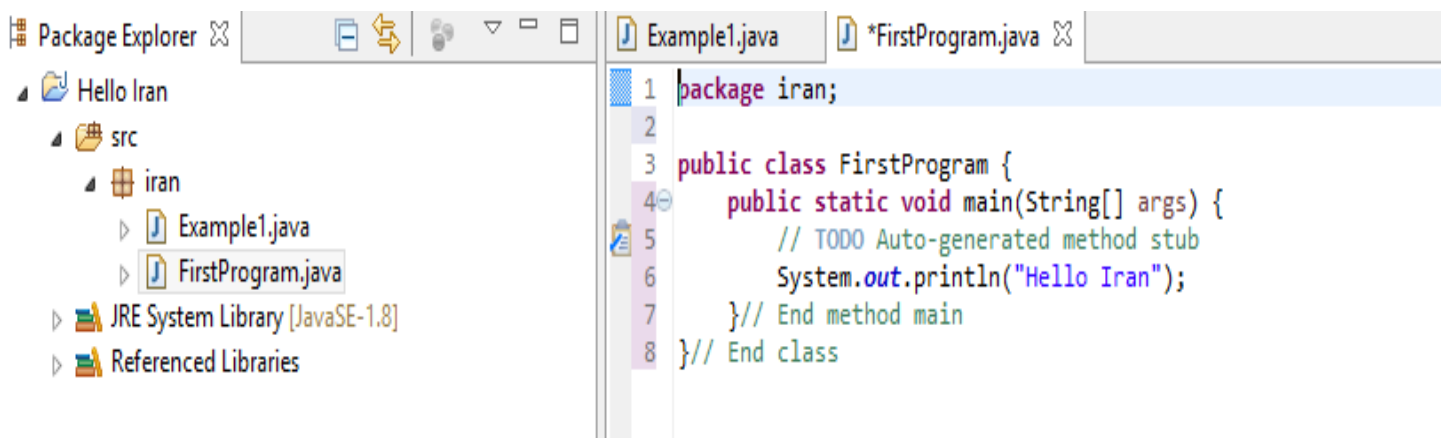
۴ روی پکیج موجود درون فولدر پروژه کلیک سمت راست کرده

۴



تصویر (۴)

در نهایت فایل جاوا و سورس کد کپی شده در Eclipse رو مشاهده میکنید: تصویر (۵)



تصویر (۵)

- اگه می بینید یک مثال ساده با این همه توضیح و فلسفه بافی هست تنها برای درک بهتر هست یواش یواش که جزئیات رو یادگیریم دیگه بیشتر فقط مثال با جواب کار خواهیم کرد.

مثال ۲: حال برنامه Example1 رو بصورت زیر داریم قصد داریم تغییراتی در آن ایجاد کنیم:

```
package iran;

public class Example1 {

    public static void main(String[] args) {
        int a=80000;
        double b=5.004;
        System.out.println(a);
        System.out.println(b);
    }
}
```

میخواهیم کاری کنیم با یک بار چاپ کردن همزمان مقدار متغیر a,b در یک سطر چاپ شود، تغییرات بصورت زیر است:

```
package iran;

public class Example1 {

    public static void main(String[] args) {
        int a=80000;
        double b=5.004;
        System.out.println(a+b);
    }
}
```

خروجی: قرمز مقدار متغیر a و آبی مقدار متغیر b

80005.004

با گذاشتن علامت + بین نام های دو متغیر آن ها رو به هم وصل کرد.

نکته:

- ما در جاوا به متن یا هر چیزی که بین دو "" قرار گیرد رشته (String) می‌گوییم.
- فاصله خالی یا هر چیزی دیگر از یک حرف تا n حرف که بین دو "" قرار گیرد رشته (String) شناخته می‌شود.
- موضوع String (رشته) خود مبحث بزرگ و مهمی هست و نیاز به یک جلسه آموزش دارد و جلسات آموزشی آینده بهش می‌پردازیم.

نمونه مثال هایی از رشته (String) در جاوا:

```
"Hello Iran"

"hello"

"Iran"
"a"

" "

"I am"

"156464565bjgjhghjghgvjhvb"

"836864836y48y3487y3@#$$%$E%^$^T&*JHGHCFS"

"1216262626"

"-----"

"_____"

"+*/-"
```

ادامه مثال: به کد زیر توجه کنید حال قصد داریم تغییرات دیگری اعمال کنیم:

```
package iran;

public class Example1 {

    public static void main(String[] args) {
        int a=80000;
    }
}
```

```

        double b=5.004;
        System.out.println(a+b);
    }
}

```

حالا ما قصد داریم وقتی همزمان مقدار دو متغیر چاپ می شود بین دو مقدار فاصله وجود داشته باشد که قاطعی نشوند. برای این کار از دو "" خالی که با زدن space بینشون به رشته " " تبدیل می شود استفاده می کنیم و با علامت + این فاصله (رشته) رو بین دو متغیر قرار می دهیم. تغییرات بصورت زیر است:

```

package iran;

public class Example1 {

    public static void main(String[] args) {
        int a=80000;
        double b=5.004;
        System.out.println(a+ " "+b);
    }
}

```

خروجی: خروجی آبی مربوط به متغیر a و خروجی قرمز مربوط به متغیر b

80000 5.004

پس راحت با علامت + می توان متغیرها و رشته ها رو به هم وصل کرد و در اینجا به ترتیب ابتدا مقدار a، رشته " " و در نهایت b چاپ می شود.

ادامه مثال: قصد داریم در کد بالا تغییرات جدیدی اعمال کنیم که بصورت زیر است:

میخواهیم در هنگام چاپ شدن مقدار متغیرها مشخص شود که مقادیر چاپ شده مربوط به کدام متغیر هست یعنی بگیم:

a=80000 b=5.004

تغییرات اعمال شده بصورت زیر است:

```

package iran;

public class Example1 {

    public static void main(String[] args) {
        int a=80000;
    }
}

```



```

double b=5.004;
System.out.println("a="+a+" "+"b="+b);

}

```

خروجی:

```
a=80000 b=5.004
```

الان دیگه مشخص هست کدوم مقدار به کدام متغیر مربوط می شود.

- همان طور که گفتیم وقتی سری حروف یا متن بین دو "" قرار گیرد رشته یا String می گوییم.
- وقتی یک متن و... بین دو "" قرار میگیرد عینا همون متن نمایش داده میشود.
- مثلا رشته "a" یک String هست اما a نام متغیر که مقداری دارد هست.
- برنامه Eclipse دیگه خودش کارش درسته!!! میتونه بین "a" و a تفاوت قائل شود و اولی رو رشته و دومی رو نام متغیر تشخیص دهد.

ادامه مثال: قصد داریم هنگام چاپ شدن خروجی بصورت زیر باشد:

```
a=80000
b=5.004
```

تغییرات اعمال شده بصورت زیر است:

```

package iran;

public class Example1 {

    public static void main(String[] args) {
        int a=80000;
        double b=5.004;
        System.out.println("a="+a+"\n"+"b="+b);
    }
}

```

خروجی:

```
a=80000
b=5.004
```

در اینجا از دستور `\n` که بین دو `"` قرار گرفته برای رفتن به سطر جدید (`newline`) استفاده شده است.

الگوریتم برنامه بصورت زیر است:

تعیین پکیج به نام `iran`

نام کلاس `Example1` که با حرف بزرگ شروع شده است.

در متد `main` دو متغیر تعریف کردیم و مستقیم بهشون مقدار دهی کردیم.

در متد `main` از دستور چاپ استفاده کردیم که بصورت زیر اجرا می شود:

چاپ رشته `"a="`

چاپ مقدار متغیر `a`

چاپ رشته `"\n"` که با چاپ شدنش به سطر بعد می رویم.

چاپ رشته `"b="`

چاپ مقدار متغیر `b`

با علامت `+` همه این ها هم به هم متصل گردیدند.

ادامه مثال: تا اینجا ما بصورت مستقیم متغیر ها رو مقدار دهی کردیم یعنی بعد از تعریف متغیر بهش مقدار دادیم

حالا قصد داریم بصورت غیر مستقیم به یکی از متغیر ها مقدار دهیم. تغییرات کد قبل بصورت زیر است:

```
package iran;

public class Example1 {

    public static void main(String[] args) {
        int a=80000;
        double b;
        b=5.004;
        System.out.println("a="+a+"\n"+"b="+b);
    }
}
```

خروجی مثل قبل است هیچ تغییری در خروجی اتفاق نمی افتد تنها ما ابتدا متغیر **b** رو از نوع **double** تعریف کردیم و در خط بعد بهش مقدار دهی کردیم. بار دوم دیگر نیاز نیست نوع متغیر رو مشخص کنیم همون یک بار که نوعش رو مشخص کردیم کافیه تنها با آوردن نام متغیر و دادن مقدار بهش، مقداردهی انجام می شود.

ادامه مثال: به تغییرات ایجاد شده در کد قبل در زیر توجه کنید:

```
package iran;

public class Example1 {

    public static void main(String[] args) {
        int a=80000;
        double b;
        b=5.004;
        a=300;
        System.out.println("a="+a+"\n"+"b="+b);
    }
}
```

خروجی: وقتی ما ابتدا به متغیر **a** مقدار ۸۰۰۰۰ نسبت داده ایم حفظه ای به متغیر **a** اختصاص داده می شود و درونش مقدار ۸۰۰۰۰ قرار می گیرد. حال در خط بعدی به **a** مقدار ۳۰۰ نسبت داده شده است پس در حافظه مقدار ۸۰۰۰۰ جای خود رو به مقدار ۳۰۰ می دهد. در برنامه نویسی آخرین مقداری که متغیر داده شده اعمال می شود.

```
a=300
b=5.004
```

برای یادگیری کد هایی که تا حالا کار کردیم رو برای خودتون داخل محیط **Eclipse** تست کنید تغییر بدهید برای خودتون چالش بتراشید و آن رو حل کنید ب یک مثال اکتفا نکنید اگه می بینید سخته نگران نباشید!!! منم که می خواستم **Eclipse** رو برای اولین بار باز کنم سخت بود از ساده ترین کد شروع کنید! ترس خودتون کنار بزارید! منم برای اولین بار با برنامه چاپ **hello** شروع کردم پس گام به گام بیاید جلو با مشکل برخورد کردید حتی اگه ساده بود و یک هفته گیرش بودید تا حلش کنید نگران نباشید چون با حلش دیگه برای بار دوم مطمئنا سریع تر می تونید یک مسئله رو حل کنید.

مثال بیشتر.....

در این بخش مثال هایی همراه با جواب مربوط به همین جلسه برای تمرین و یادگیری بهتر طراحی شده است با اجرا کردن و تغییر دادن این کدها خودتون رو به چالش بکشید!!!!!!!

```
package iran;

public class Example1 {

    public static void main(String[] args) {
        int a=80000;
        double b;
        char ch='k';
        b=5.004;
        a=300;
        System.out.println("a="+a+"\n"+"b="+b);
        System.out.println(ch);
    }
}
```

خروجی:

```
a=300
b=5.004
k
```

```
package iran;

public class Example1 {

    public static void main(String[] args) {
        int a=80000;
        double b;
        char ch='s';
        char ch2=97;
        char ch3='a';
        b=5.004;
        a=300;
        System.out.println("a="+a+"\n"+"b="+b);
        System.out.println(ch+"alam");
        System.out.print(ch2+"="+ch3);
    }
}
```

```

    }
}

```

خروجی:

```

a=300
b=5.004
salam
a=a

```

```

package iran;

public class Example1 {

    public static void main(String[] args) {
        float d = 2.65f;
        System.out.println("@#TY**@I(UO!UP!UP@5236");
        System.out.println(d + " " + "is" + " " + "flaot");

        System.out.println("zende bad iran ziba");
    }
}

```

خروجی:

```

@#TY**@I(UO!UP!UP@5236
2.65 is flaot
zende bad iran ziba

```

```

package iran;

public class Example1 {

    public static void main(String[] args) {
        boolean a=true;
        boolean b=false;
        System.out.println(a);
        System.out.println(b);
        System.out.println("a="+a+" "+"b="+b);
    }
}

```

خروجی:

```
true
false
a=true b=false
```

```
package iran;

public class Example1 {

    public static void main(String[] args) {
        long a = 1000000000;
        int b = 1000000000;
        short c = 10000;
        byte d = 100;

        System.out.println(a + ">" + b + ">" + c + ">" + d);
    }
}
```

خروجی:

```
1000000000>1000000000>10000>100
```

```
package iran;

public class Example1 {

    public static void main(String[] args) {
        System.out.println("800">"600">"400">"200");
        System.out.println(200);
        System.out.println(800">"600">"400">"200");
    }
}
```

خروجی:

```
800>600>400>200
200
800>600>400>200
```

نکته آخر: در جلسات آینده وقتی به مبحث عملگرها و عملیات ریاضی رسیدیم تفاوت این دو خروجی براتون خواهم گفت. تنها این رو بدونید که خروجی اولی رشته و متن (String) هست و خروجی دوم و سوم عدد هست و می شود عملیات ریاضی نظیر جمع و ضرب و تفریق و... روشن انجام داد.!!!!!!

پیروز و موفق باشید

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

www.JAVAPRO.ir

آموزش جاوا SE را با تجربه شخصی 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.