

در مثال زیر نحوه تعریف و مقدار دهی متغیرها نمایش داده شده است :

```

1 package myfirstprogram;
2
3 import java.text.MessageFormat;
4
5
6 public class MyFirstProgram
7 {
8     public static void main(String[] args)
9     {
10         //Declare variables
11         int num1;
12         int num2;
13         double num3;
14         double num4;
15         boolean boolVal;
16         char myChar;
17
18         //Assign values to variables
19         num1 = 1;
20         num2 = 2;
21         num3 = 3.54;
22         num4 = 4.12;
23         boolVal = true;
24         myChar = 'R';
25
26         //Show the values of the variables
27         System.out.println(MessageFormat.format("num1 = {0}", num1));
28         System.out.println(MessageFormat.format("num3 = {0}", num3));
29         System.out.println(MessageFormat.format("num4 = {0}", num4));
30         System.out.println(MessageFormat.format("boolVal = {0}", boolVal));
31         System.out.println(MessageFormat.format("num2 = {0}", num2));
32         System.out.println(MessageFormat.format("myChar = {0}", myChar));
33     }
34 }
```

```

num1 = 1
num2 = 2
num3 = 3.54
num4 = 4.12
boolVal = true
myChar = R

```

تعریف متغیر

در خطوط 11-16 متغیرهایی با نوع و نام متفاوت تعریف شده‌اند. ابتدا باید نوع داده‌هایی را که این متغیرها قرار است در خود ذخیره کنند را مشخص کنیم و سپس یک نام برای آنها در نظر بگیریم و در آخر سیمیکولن بگذاریم. همیشه به یاد داشته باشید که قبل از مقدار دهی و استفاده از متغیر باید آن را تعریف کرد.

```

int num1;
int num2;
double num3;
double num4;
bool boolVal;
char myChar;

```

نحوه تعریف متغیر به صورت زیر است :

```

data_type identifier;

```

date_type همان نوع داده است مانند int ,double و... .

Identifier نیز نام متغیر است که به ما امکان استفاده و دسترسی به مقدار متغیر را می‌دهد.

برای تعریف چند متغیر از یک نوع می‌توان به صورت زیر عمل کرد :

```

data_type identifier1, identifier2, ... indentifierN;

```

مثال

```

int num1, num2, num3, num4, num5;

```

در مثال بالا 5 متغیر از نوع صحیح تعریف شده است. توجه داشته باشید که بین متغیرها باید علامت کاما (,) باشد.

نامگذاری متغیرها

- نام متغیر باید با یک حرف یا زیرخط و به دنبال آن حرف یا عدد شروع شود.
- نمی‌توان از کاراکترهای خاص مانند #، %، & یا عدد برای شروع نام متغیر استفاده کرد مانند 2numbers.
- نام متغیر نباید دارای فاصله باشد. برای نام‌های چند حرفی می‌توان به جای فاصله از علامت زیرخط یا _ استفاده کرد.

نامهای مجاز :

num1	myNumber	studentCount	total	first_name	_minimum
num2	myChar	average	amountDue	last_name	_maximum
name	counter	sum	isLeapYear	color_of_car	_age

نامهای غیر مجاز :

123	#numbers#	#ofstudents	1abc2
123abc	\$money	first name	ty.np
my number	this&that	last name	1:00

اگر به نامهای مجاز در مثال بالا توجه کنید متوجه قراردادهای به کار رفته در نامگذاری آن‌ها خواهید شد. یکی از روشهای نامگذاری، نامگذاری کوهان شتری است. در این روش که برای متغیرهای دو کلمه‌ای به کار می‌رود، اولین حرف اولین کلمه با حرف کوچک و اولین حرف دومین کلمه با حرف بزرگ نمایش داده می‌شود مانند : myNumber توجه کنید که اولین حرف کلمه Number با حرف بزرگ شروع شده است. مثال دیگر کلمه numberOfStudents است. اگر

توجه کنید، بعد از اولین کلمه، حرف اول سایر کلمات با حروف بزرگ نمایش داده شده است.

محدوده متغیر

متغیرهای ابتدای درس در داخل متد main() تعریف شده‌اند. در نتیجه، این متغیرها فقط در داخل متد main() قابل دسترسی و استفاده هستند. محدوده یک متغیر مشخص می‌کند که متغیر در کجای کد قابل دسترسی است. هنگامیکه برنامه به پایان متد main() می‌رسد متغیرها از محدوده خارج و بدون استفاده می‌شوند تا زمانی که برنامه در حال اجراست. محدوده متغیرها انواعی دارد که در درسهای بعدی با آنها آشنا می‌شوید. تشخیص محدوده متغیر بسیار مهم است چون به وسیله آن می‌فهمید که در کجای کد می‌توان از متغیر استفاده کرد. باید یاد آور شد که دو متغیر در یک محدوده نمی‌توانند دارای نام یکسان باشند. مثلاً کد زیر در برنامه ایجاد خطا می‌کند :

```

int num1;
int num1;

```

از آنجاییکه جاوا به بزرگی و کوچک بودن حروف حساس است می‌توان از این خاصیت برای تعریف چند متغیر هم نام ولی با حروف متفاوت (از لحاظ بزرگی و کوچکی) برای تعریف چند متغیر از یک نوع استفاده کرد مانند :

```

int num1;
int Num1;
int NUM1;

```

مقداردهی متغیرها

می‌توان فوراً بعد از تعریف متغیرها مقداری را به آنها اختصاص داد. این عمل را مقداردهی می‌نامند. در زیر نحوه مقدار دهی متغیرها نشان داده شده است :

```

data_type identifier = value;

```

به عنوان مثال :

```

int myNumber = 7;

```

همچنین می‌توان چندین متغیر را فقط با گذاشتن کاما بین آنها به سادگی مقدار دهی کرد :

```

data_type variable1 = value1, variable2 = value2, ... variableN, valueN;

```

```

int num1 = 1, num2 = 2, num3 = 3;

```

تعریف متغیر با مقدار دهی متغیرها متفاوت است. تعریف متغیر یعنی انتخاب نوع و نام برای متغیر ولی مقدار دهی یعنی اختصاص یک مقدار به متغیر.

اختصاص مقدار به متغیر

در زیر نحوه اختصاص مقادیر به متغیرها نشان داده شده است:

```

num1 = 1;
num2 = 2;
num3 = 3.54;
num4 = 4.12;
boolVal = true;
myChar = 'R';

```

به این نکته توجه کنید که شما به مغیری که هنوز تعریف نشده نمی‌توانید مقدار بدهید. شما فقط می‌توانید از متغیرهایی استفاده کنید که هم تعریف و هم مقدار دهی شده باشند. مثلاً متغیرهای بالا همه قابل استفاده هستند. در این مثال num1 و num2 هر دو تعریف شده‌اند و مقداری از نوع صحیح به آنها اختصاص داده شده است. اگر نوع داده با نوع متغیر یکی نباشد برنامه پیغام خطا می‌دهد.

جانگهدار (Placeholders)

به متد format() از کلاس MessageFormat در خطوط (32-27) توجه کنید. برای استفاده از متد format() و کلاس MessageFormat ابتدا باید Package مربوط به آنها را در برنامه وارد کنید (خط 3) :

```

import java.text.MessageFormat;

```

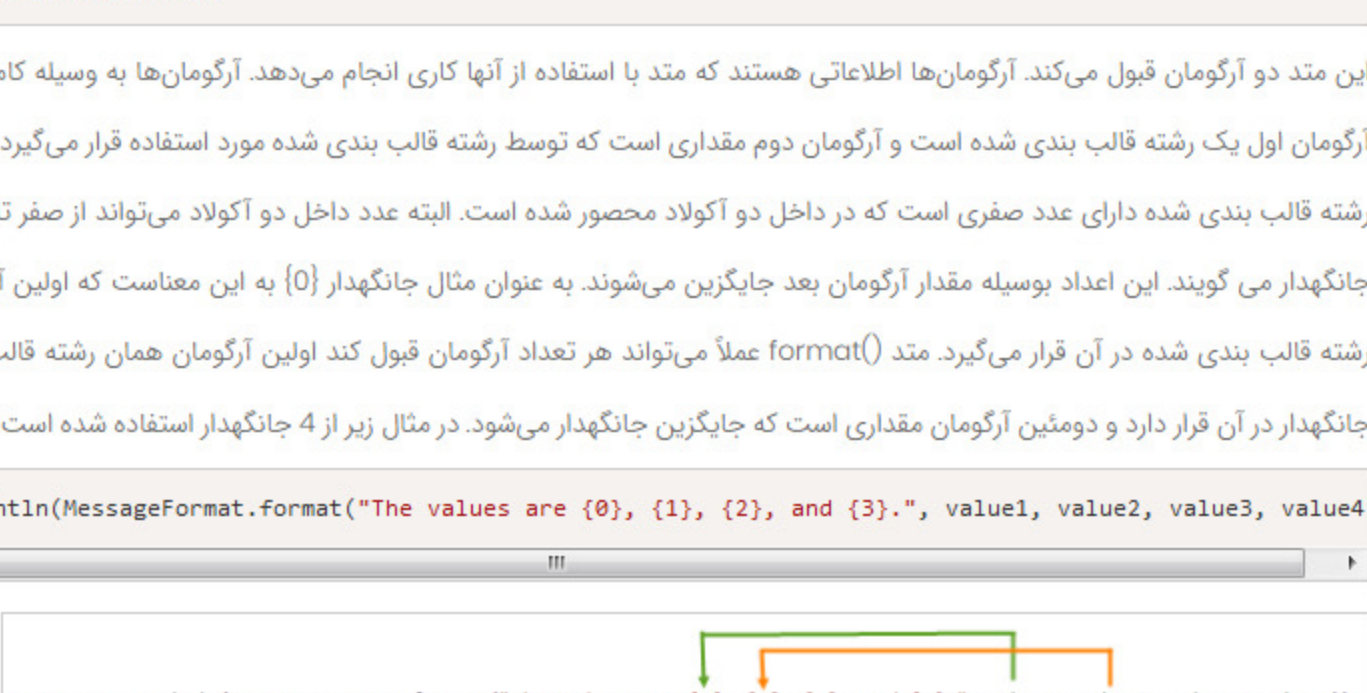
این متد دو آرگومان قبول می‌کند. آرگومان‌ها اطلاعاتی هستند که متد با استفاده از آنها کاری انجام می‌دهد. آرگومان‌ها به وسیله کاما از هم جدا می‌شوند. آرگومان اول یک رشته قالب بندی شده است و آرگومان دوم مقداری است که توسط رشته قالب بندی شده مورد استفاده قرار می‌گیرد. اگر به دقت نگاه کنید رشته قالب بندی شده دارای عدد صفری است که در داخل دو آکولاد محصور شده است. البته عدد داخل دو آکولاد می‌تواند از صفر تا n باشد. به این اعداد

جانگهدار می‌گویند. این اعداد بوسیله مقدار آرگومان بعد جایگزین می‌شوند. به عنوان مثال جانگهدار {0} به این معناست که اولین آرگومان (مقدار) بعد از رشته قالب بندی شده در آن قرار می‌گیرد. متد format() عملاً می‌تواند هر تعداد آرگومان قبول کند اولین آرگومان همان رشته قالب بندی شده است که جانگهدار در آن قرار دارد و دومین آرگومان مقداری است که جایگزین جانگهدار می‌شود. در مثال زیر از 4 جانگهدار استفاده شده است :

```

System.out.println(MessageFormat.format("The values are {0}, {1}, {2}, and {3}.", value1, value2, value3, value4

```



جانگهدارها از صفر شروع می‌شوند. تعداد جانگهدارها باید با تعداد آرگومانهای بعد از رشته قالب بندی شده برابر باشد. برای مثال اگر شما چهار جانگهدار مثل بالا داشته باشید باید چهار مقدار هم برای آنها بعد از رشته قالب بندی شده در نظر بگیرید. اولین جا نگهدار با دومین آرگومان و دومین جا نگهدار با سومین آرگومان جایگزین می‌شود. در ابتدا فهمیدن این مفهوم برای کسانی که تازه برنامه نویسی را شروع کرده‌اند سخت است اما در درسهای آینده مثالهای

یادی در این مورد مشاهده خواهید کرد.