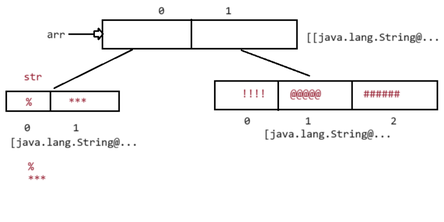
```
String[] arr[] ={{"%","***"},{"!!!!","@@@@@","######"}};

        for(String str[] : arr){
            for(String s:str)
            {
                System.out.println(s);
                if(s.length()==4)
                    break;
            }
            break;
        }
    }
```

Takes the control outside the loop




```
class Test
{
    public void m1(int i,float f)
    {
        System.out.println("int-float arg method");
    }
    public void m1(float f,int i)
    {
        System.out.println("float-int arg method");
    }
}

public static void main(String[] args)
{
    Test t= new Test();
    t.m1(10,10);
}
```

int int

int ===> float int ===> float
int ===>long int ===>long
int ===>double int ===>double

byte -----> short -----> int -----> long ---> float ----> double

char

```
class Test{
    public void m1(){
        System.out.println("Zero Arg");
    }
    public void m1(int a){
        System.out.println(a);
    }
    public void m1(int a, int b){
        System.out.println(a+b);
    }
    public void m1(int a, int b, int c){
        System.out.println(a+b+c);
    }
    public void m1(int a,int b, int c, int d){
        System.out.println(a+b+c+d);
    }

    public static void main(String[] args) {

    }
}
```

m1()
m1(int)
m1(int,int)
m1(int,int,int)
m1(int,int,int,int)

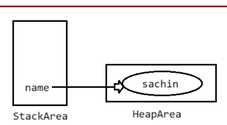
var-args

```
public void m1(int...  data)
{

}
```

m1()
m1(int)
m1(int,int)
m1(int,int,int)
m1(int,int,int,int)

```
String name = "sachin";
```



1. Immutable data
   Can't be changed
2. Mutable data
   Can be changed

String data

Immutable String
1. String

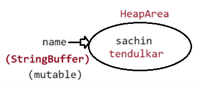
Mutable String
1. StringBuffer
2. StringBuilder(1.5V)

Immutable

```
String name = "sachin";
name.concat("tendulkar");
System.out.println(name);//sachin
```

Mutable

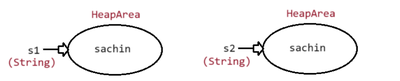
```
StringBuffer name = new StringBuffer("sachin");
name.append("tendulkar");
System.out.println(name);//sachintendulkar
```




```
String s1 = new String("sachin");
String s2 = new String("sachin");

System.out.println(s1==s2);//false
System.out.println(s1.equals(s2));//true
```

String

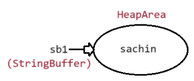


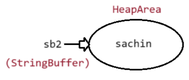
== -> compares the reference

```
StringBuffer sb1 =new StringBuffer("sachin");
StringBuffer sb2 =new StringBuffer("sachin");

System.out.println(sb1==sb2); //false
System.out.println(sb1.equals(sb2));//false

System.out.println(sb1); //sachin
System.out.println(sb2); //sachin
```





== -> compares the reference