

while, break and continue

Example #1.

```
public class BreakContinue {  
    /**/  
    public static void main ( String [] arg ) {  
        /**/  
        int [] ia=new int [] { 17, -32, 12, 43 };  
        /**/  
        System.out.println();  
        for ( int i=0; i<ia.length; ++i ) {  
            if ( ia[i] == 12 ) break;  
            System.out.println(ia[i]);  
        }  
        /**/  
        System.out.println();  
        for ( int i=0; i<ia.length; ++i ) {  
            if ( ia[i] == 12 ) continue;  
            System.out.println(ia[i]);  
        }  
    }  
}
```

The output of the above program is

```
17  
-32
```

```
17  
-32  
43
```

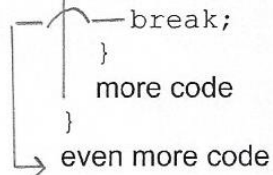
One sees that the `break` statement “breaks out” of the loop, while the `continue` statement makes the execution go back to the top of the loop.

Example #2. while loops.

```

boolean ok=true;
while ( ok ) {
    code which calculates booleanCondition
    if ( booleanCondition ) {
        break;
    }
    more code
}
even more code

```

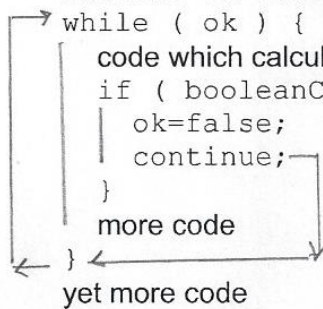


break "breaks out"
of the loop

```

boolean ok=true;
while ( ok ) {
    code which calculates booleanCondition
    if ( booleanCondition ) {
        ok=false;
        continue;
    }
    more code
}
yet more code

```



continue goes to the end of the
loop, and the loop is tested again

Example #3.

```

int k=-1;
while ( true ) {
    ++k;
    code
    if ( k == 5 ) break;
}

```

is equivalent to

```

for ( int k=0; k<6; ++k ) {
    code
}

```