

دستور break در جاوا

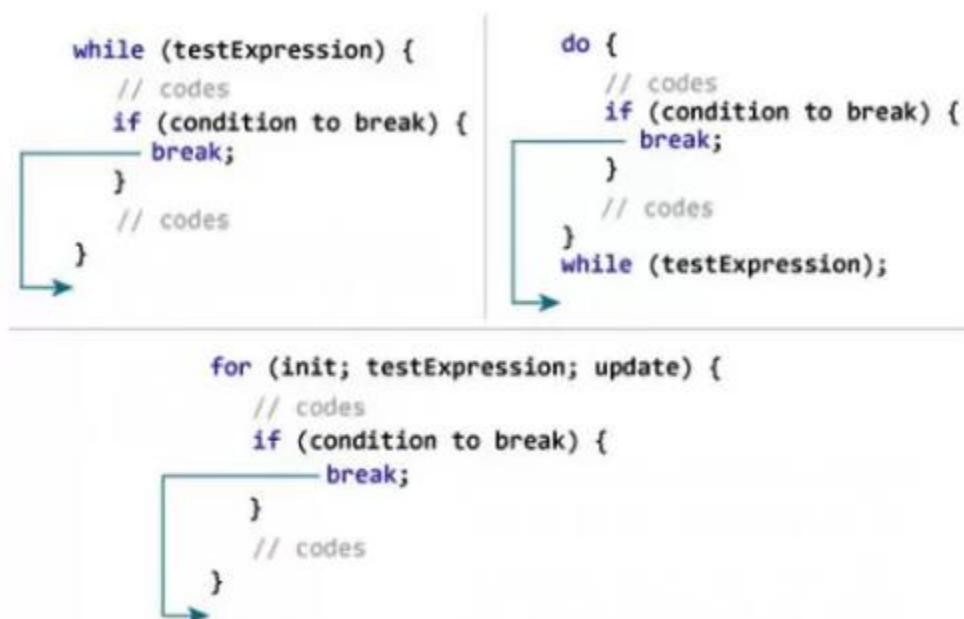
در این آموزش یاد خواهید گرفت که از break برای خاتمه دادن به یک حلقه استفاده کنید.

فرض کنید در حال کار با حلقه ها هستید. گاهی اوقات احتیاج دارید که از روی برخی دستورات داخل حلقه پرش کنید یا حلقه را فوراً بدون چک کردن شرط، خاتمه دهید. در چنین مواردی از دستورات break و continue استفاده می شود. عبارت break حلقه را فوراً خاتمه می دهد و کنترل برنامه به بعد از حلقه می رود. این مورد اکثر اوقات با بیانیه های تصمیم گیری استفاده می شود. (if ... else).

ساختار break عبارت است از:

break;

دستور break چگونه کار می کند؟



مثال ۱ break : در جاوا

```
1. class Test {
2.     public static void main(String[] args) {
3.
4.         for (int i = 1; i <= 10; ++i) {
5.             if (i == 5) {
6.                 break;
7.             }
8.             System.out.println(i);
9.         }
10.    }
11. }
```

خروجی

```
۱
۲
۳
۴
```

در برنامه بالا، هنگامی که مقدار i برابر با ۵ می شود، عبارت $i == 5$ را در داخل پرانتز `if` قرار می دهیم. سپس، دستور `break` اجرا می شود و حلقه `for` را خاتمه می دهد.

مثال ۲ : عبارت break در جاوا

برنامه زیر مجموع اعدادی که کاربر وارد می کند را تا زمانی که عدد منفی وارد نکرده حساب می کند.

برای گرفتن ورودی از کاربر، از شی `Scanner` استفاده می شود.

```
1. import java.util.Scanner;
2. class UserInputSum {
3.     public static void main(String[] args) {
4.
5.         Double number, sum = 0.0;
6.         Scanner input = new Scanner(System.in);
7.
8.         while (true) {
9.             System.out.print("Enter a number: ");
10.            number = input.nextDouble();
11.
12.            if (number < 0.0) {
13.                break;
14.            }
15.
16.            sum += number;
```

```
17. }
18. System.out.println("Sum = " + sum);
19. }
20. }
```

خروجی

```
Enter a number: 3.2
Enter a number: 5
Enter a number: 2.3
Enter a number: 0
Enter a number: -4.5
Sum = 10.5
```

در برنامه بالا، شرط حلقه `while` همیشه درست است و تا زمانی که کاربر عدد منفی وارد نکند، اجرا می شود. اگر کاربر عدد منفی وارد کرد، دستور داخل بدنه ی `if` اجرا می شود که حلقه `while` را خاتمه می دهد.

در صورت وجود حلقه های تو در تو، `break` داخلی ترین حلقه را خاتمه می دهد.

```
while (testExpression) { ←
    // codes
    while (testExpression) {
        // codes
        if (condition to break) {
            break;
        }
        // codes
    }
    // codes
}
```

در اینجا، عبارت `break` حلقه `while` داخلی را خاتمه می دهد و کنترل برنامه به حلقه بیرونی می پرد.


دستور `break` برچسب دار

عبارت `break` که تاکنون درباره آن بحث کرده ایم، فرم بدون علامت `break` است، که درونی ترین حلقه `for`، `while`، `do ... while`، `switch` را خاتمه می دهد. شکل دیگری از

دستور break وجود دارد که دارای برچسب break است که می تواند برای خاتمه حلقه بیرونی استفاده شود.

دستور break برچسب دار چگونه کار می کند؟

```
label:
for (int; testExpression, update) {
    // codes
    for (int; testExpression; update) {
        // codes
        if (condition to break) {
            break label;
        }
        // codes
    }
    // codes
}
```



در اینجا، برچسب یک شناسه است. هنگام اجرای break، دستور دارای برچسب خاتمه می یابد، و کنترل برنامه به بعد از دستور برچسب زده شده می پرد.

یک مثال دیگر:

```
1. while (testExpression) {
2. // codes
3. second:
4. while (testExpression) {
5. // codes
6. while(testExpression) {
7. // codes
8. break second;
9. }
10. }
11. // control jumps here
12. }
```

هنگام اجرای break کنترل برنامه به زیر عبارت با برچسب second می پرد.

مثال ۳: ساختار break برچسب دار

```
1. class LabeledBreak {
2. public static void main(String[] args) {
3.
4. first:
5. for (int i = 1; i < 5; i++) {
```

```

6. second:
7. for(int j = 1; j < 3; j ++ ) {
8. System.out.println("i = " + i + "; j = " +j);
9.
10. if ( i == 2)
11. break first;
12. }
13. }
14. }
15. }

```

خروجی

```

i = 1; j = 1
i = 1; j = 2
i = 2; j = 1

```

در اینجا روش دیگر برنامه بالا آورده شده است `break`. در برنامه زیر، عبارت `برچسب دار second` را خاتمه می دهد و کنترل برنامه به زیر عبارت `با برچسب second` می پرد.

```

1. class LabeledBreak {
2. public static void main(String[] args) {
3.
4. first:
5. for( int i = 1; i < 5; i++) {
6. second:
7. for(int j = 1; j < 3; j ++ ) {
8. System.out.println("i = " + i + "; j = " +j);
9.
10. if ( i == 2)
11. break second;
12. }
13. }
14. }
15. }

```

خروجی

```

i = 1; j = 1
i = 1; j = 2
i = 2; j = 1
i = 3; j = 1
i = 3; j = 2
i = 4; j = 1
i = 4; j = 2

```

دستور `break` برای خاتمه `switch` هم استفاده می شود.