



Break & Continue Statement

Lesson:
JAVA Chapter 09

Break Statement

Break Komutu

Normal koşullarda, bir **loop** koşul bölümü **true** sonuç verdiği sürece çalışmasına devam eder. Ancak, **loop** kod bloğu içinde özel bir tanımlama yaparak da **loop'u** sona erdirebilirsiniz.

Bu işlemi gerçekleştirmek için **break komutunu** kullanılır.

Bir **loop'u**, **loop'un** koşul bölümüne bağlı kalmaksızın sona erdirmek için **break komutu** kullanılır. **break komutu** **loop'a** bağlı kod bloğunun herhangi bir işlem satırında yer alabilir. Program kod bloğunda **break komutu'nu** görür görmez, **loop** başlangıç koşulu sağlanmış olsa bile, **loop'dan** çıkar ve **loop'dan** sonra gelen ilk işlem satırından çalışmasına devam eder.

İç içe(nested) **looplarda** **break komutu** kullanıldığından sadece içinde kullanıldığı **loop'dan** çıkışı sağlar:

Java **break** komutu

for loop

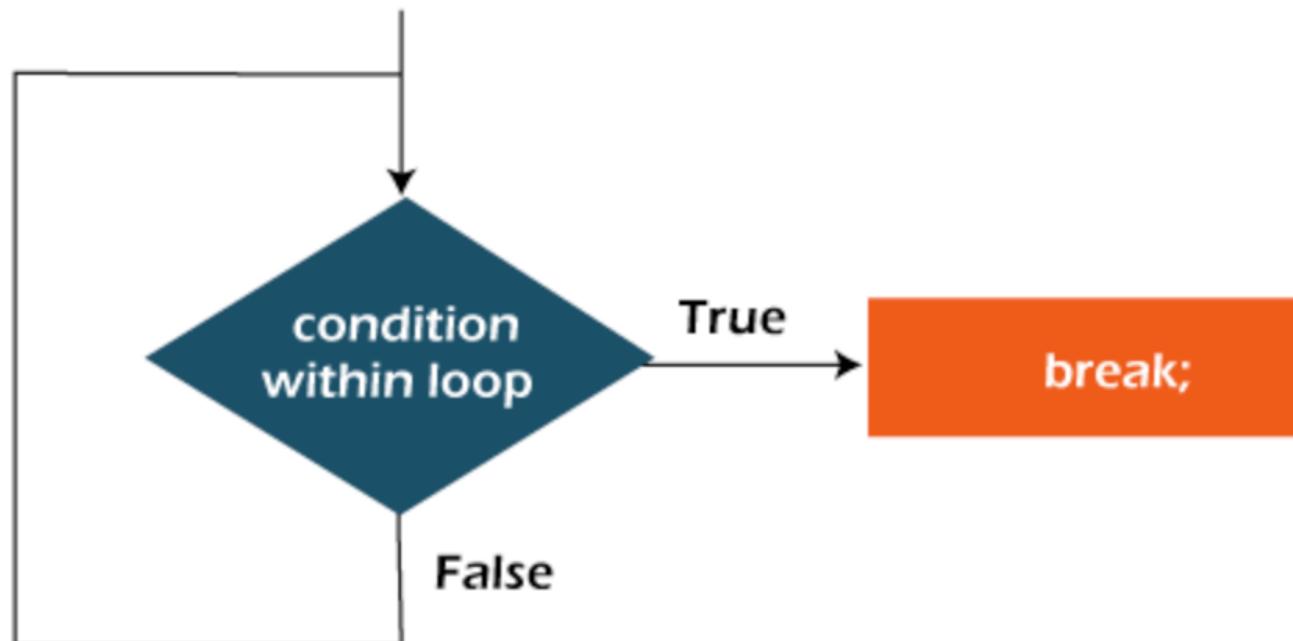
while loop

do-while loop

gibi tüm **loop'larda** kullanılabilir.



Break Statement



Flowchart of break statement



Break Statement

Aşağıdaki program, ekrana 1-3 arasındaki sayıları ekrana yazdıktan sonra döngüden çıkar ve çalışması sona erer.:

```
/* javaprog.java */
public class javaprog {
    public static void main(String[] args) {
        int id1;

        for (id1=1; id1<6; id1++) {
            System.out.println(id1);
            if (id1==3) break;
        }
    }
}
```

Komut satırında aşağıdaki satırları yazdığınızda program derlenir ve çalışır:

```
C:\>javac javaprog.java
C:\>java -Dfile.encoding=Cp857 javaprog
1
2
3
```

Yukarıdaki örnekte, program break işlem satırı tanımlanmadığında 1'den 5'e kadar olan sayıları ekrana yazar. Ancak, for döngüsüne bağlı kod bloğunun ikinci satırında break deyimi kullanıldığı için, program for döngüsünde id1 değişkenine 3 değerini atadığında if satırında koşul doğru sonuç verdiğiinde break deyimi devreye gireceğinden döngü hemen sona erer. Eğer break deyimi System.out.println(id1); işlem satırından önce yazılmış olsaydı, program 3 sayısını da yazmadan döngü sona erecekti.



Break Statement

```
int i = 0;  
while (i < 10) {  
    System.out.println(i);  
    i++;  
    if (i == 4) {  
        break;  
    }  
}
```



Continue Statement

Continue Komutu

continue komutu, bir **loop'un** herhangi bir satırından itibaren **loop'un** geri kalan işlem satırlarını çalıştırmadan, **loop'un** baş tarafına dönme olanağı sağlar.

do ve while loop'larda, program **continue komutu** ile karşılaşlığında, koşul bölümünü kontrol eder ve **loop'un** çalışmasını devam ettirir. **for loopda** ise, artırma ve koşul bölümünü çalıştırıldıktan sonra **loop'un** çalışmasını devam ettirir.

Java **continue komutu** loop'a devam etmek için kullanılır. Programın mevcut akışını sürdürür ve kalan kodu belirtilen koşulda atlar. Bir iç döngü olması durumunda, yalnızca iç döngüye devam eder.

Kısaca -> **Program**, bir **loop** içerisinde **continue komutu** ile karşılaşılırsa, ondan sonraki komutları atlar ve **loop'un** bir sonraki değerinden devam eder.

Java **continue komutu**

for loop

while loop

do-while loop

gibi tüm **loop'larda** kullanabiliriz .



Continue Statement

Aşağıdaki program, ekrana 1-10 arasındaki çift sayıları ekrana yazar:

```
/* javaprog.java */
public class javaprog {
    public static void main(String[] args) {
        int id1;

        for (id1=1; id1<11; id1++) {
            if (id1%2==1) continue;
            System.out.println(id1);
        }
    }
}
```

Komut satırında aşağıdaki satırları yazdığınızda program derlenir ve çalışır:

```
C:\>javac javaprog.java
C:\>java -Dfile.encoding=Cp857 javaprog
2
4
6
8
10
```

Program, for döngüsü içindeki continue deyiminden dolayı, döngünün ilk satırında eğer id1 değişkeni tek bir sayı ise geri kalan kalan satırları devre dışı bırakarak, bir sonraki döngü değeri ile çalışmasına devam eder.



Continue Statement

```
int i = 0;  
while (i < 10) {  
    if (i == 4) {  
        i++;  
        continue;  
    }  
    System.out.println(i);  
    i++;  
}
```



Continue Statement

Programımız 0'dan 10'a kadar sayıları/rakamları yazmak yerine i değeri 4 olduğunda **continue** nedeniyle 4'Ü ekrana yazmayıp, doğrudan döngünün başına gitti ve i değeri 5 için çalışmaya devam etti. Sonrasında i değeri 7 olduğunda **break** nedeniyle doğrudan döngü bloğundan çıktı ve 7,8,9 ve 10'u ekrana yazmadı.

```
public static void main(String[] args) {  
  
    for (int i = 0 ; i <= 10 ; i++){  
        if (i == 4) continue;  
        if (i == 7) break;  
        System.out.println(i);  
    }  
}
```

Program Çıktısı :

```
0  
1  
2  
3  
5  
6
```

```
Process finished with exit code 0
```



THANKS!

Any questions?

HalUk Bilgin | JAVA Backend Developer

BAŞARI GAYRETE AŞIKTIR ☺

