



# G-CAMP WORKSHOP

BUILD THE RIGHT THING  
BUILD THING RIGHT  
SHIP FAST

*Lecturer* Kim Myoung-Ho  
*Nickname* 블스  
[blogstudy@naver.com](mailto:blogstudy@naver.com)

## 8강\_반복문

- for문
- foreach문
- while문
- do ~ while문
- break와 continue 키워드



## 8-1. for문

ex) lec08/loopPjt

- for문 : 괄호안의 결과에 따라 실행문의 반복실행 여부가 결정된다.

```
for (시작; 끝; 단계)
{
    실행문
}
```



괄호안의 결과가 True인 경우에만 반복 실행문이 실행한다.

```
for (int i = 0; i < 11; i++) { 실행문 }
```



'0'부터 '10'까지 '1'씩 증가하면서 실행문을 반복 실행해라!



# 8-1. for문



ex) lec08/loopPjt

- for문을 이용해서 구구단을 출력해 보자.

```
특정 구구단 출력
3 * 1 = 3
3 * 2 = 6
3 * 3 = 9
3 * 4 = 12
3 * 5 = 15
3 * 6 = 18
3 * 7 = 21
3 * 8 = 24
3 * 9 = 27
```

```
사용자가 입력한 구구단 출력
원하는 구구단을 입력하세요.
4
4 * 1 = 4
4 * 2 = 8
4 * 3 = 12
4 * 4 = 16
4 * 5 = 20
4 * 6 = 24
4 * 7 = 28
4 * 8 = 32
4 * 9 = 36
```

```
구구단 전체 출력 - II
구구단 전체를 출력합니다. (가로)
2 * 1 = 2   3 * 1 = 3   4 * 1 = 4   5 * 1 = 5   6 * 1 = 6   7 * 1 = 7   8 * 1 = 8   9 * 1 = 9
2 * 2 = 4   3 * 2 = 6   4 * 2 = 8   5 * 2 = 10  6 * 2 = 12  7 * 2 = 14  8 * 2 = 16  9 * 2 = 18
2 * 3 = 6   3 * 3 = 9   4 * 3 = 12  5 * 3 = 15  6 * 3 = 18  7 * 3 = 21  8 * 3 = 24  9 * 3 = 27
2 * 4 = 8   3 * 4 = 12  4 * 4 = 16  5 * 4 = 20  6 * 4 = 24  7 * 4 = 28  8 * 4 = 32  9 * 4 = 36
2 * 5 = 10  3 * 5 = 15  4 * 5 = 20  5 * 5 = 25  6 * 5 = 30  7 * 5 = 35  8 * 5 = 40  9 * 5 = 45
2 * 6 = 12  3 * 6 = 18  4 * 6 = 24  5 * 6 = 30  6 * 6 = 36  7 * 6 = 42  8 * 6 = 48  9 * 6 = 54
2 * 7 = 14  3 * 7 = 21  4 * 7 = 28  5 * 7 = 35  6 * 7 = 42  7 * 7 = 49  8 * 7 = 56  9 * 7 = 63
2 * 8 = 16  3 * 8 = 24  4 * 8 = 32  5 * 8 = 40  6 * 8 = 48  7 * 8 = 56  8 * 8 = 64  9 * 8 = 72
2 * 9 = 18  3 * 9 = 27  4 * 9 = 36  5 * 9 = 45  6 * 9 = 54  7 * 9 = 63  8 * 9 = 72  9 * 9 = 81
```



## 8-2. foreach문

- for문의 단축 문(연속된 데이터(배열 등)에 사용).

```
foreach ( 데이터 in 연속된 데이터 )  
{  
    실행문  
}
```

for (string str in strArr) {실행문}



**strArr의 데이터를 순차적으로 참조한다.**



- for문과 비슷한 반복문

```
while( 조건식 )  
{  
    실행문  
}
```

```
while ( i <= 9 ) { 실행문 }
```



**i가 9보다 작거나 같은동안 반복 실행한다.**



## 8-3. while문

ex) lec08/loopPjt

- while문을 이용해서 구구단을 출력해 보자.

```
while문을 이용한 구구단 출력
2 * 1 = 2      3 * 1 = 3      4 * 1 = 4      5 * 1 = 5      6 * 1 = 6      7 * 1 = 7      8 * 1 = 8      9 * 1 = 9
2 * 2 = 4      3 * 2 = 6      4 * 2 = 8      5 * 2 = 10     6 * 2 = 12     7 * 2 = 14     8 * 2 = 16     9 * 2 = 18
2 * 3 = 6      3 * 3 = 9      4 * 3 = 12     5 * 3 = 15     6 * 3 = 18     7 * 3 = 21     8 * 3 = 24     9 * 3 = 27
2 * 4 = 8      3 * 4 = 12     4 * 4 = 16     5 * 4 = 20     6 * 4 = 24     7 * 4 = 28     8 * 4 = 32     9 * 4 = 36
2 * 5 = 10     3 * 5 = 15     4 * 5 = 20     5 * 5 = 25     6 * 5 = 30     7 * 5 = 35     8 * 5 = 40     9 * 5 = 45
2 * 6 = 12     3 * 6 = 18     4 * 6 = 24     5 * 6 = 30     6 * 6 = 36     7 * 6 = 42     8 * 6 = 48     9 * 6 = 54
2 * 7 = 14     3 * 7 = 21     4 * 7 = 28     5 * 7 = 35     6 * 7 = 42     7 * 7 = 49     8 * 7 = 56     9 * 7 = 63
2 * 8 = 16     3 * 8 = 24     4 * 8 = 32     5 * 8 = 40     6 * 8 = 48     7 * 8 = 56     8 * 8 = 64     9 * 8 = 72
2 * 9 = 18     3 * 9 = 27     4 * 9 = 36     5 * 9 = 45     6 * 9 = 54     7 * 9 = 63     8 * 9 = 72     9 * 9 = 81
```





- while문과 비슷한 반복문

```
do  
{  
    실행문  
}  
while ( 조건식 );
```



조건 결과에 상관없이 무조건 한번은 실행 한다.

```
int kNum = 10;  
  
do  
{  
    Console.WriteLine($"kNum = {kNum}");  
}  
while (kNum < 0);
```



kNum = 10





## 8-5. break와 continue 키워드

- break : 빠져 나온다. / continue : 다음으로 넘어간다.

```
for(int i=0; i<10; i++)  
{  
    ...  
    break;  
    ...  
}
```

```
for(int i=0; i<10; i++)  
{  
    ...  
    continue;  
    ...  
}
```



## 8-5. break와 continue 키워드



ex) lec08/loopPjt

- 사용자가 입력한 구구단을 제외한 전체 구구단을 출력하자.

```
3
2 * 1 = 2      4 * 1 = 4      5 * 1 = 5      6 * 1 = 6      7 * 1 = 7      8 * 1 = 8      9 * 1 = 9
2 * 2 = 4      4 * 2 = 8      5 * 2 = 10     6 * 2 = 12     7 * 2 = 14     8 * 2 = 16     9 * 2 = 18
2 * 3 = 6      4 * 3 = 12     5 * 3 = 15     6 * 3 = 18     7 * 3 = 21     8 * 3 = 24     9 * 3 = 27
2 * 4 = 8      4 * 4 = 16     5 * 4 = 20     6 * 4 = 24     7 * 4 = 28     8 * 4 = 32     9 * 4 = 36
2 * 5 = 10     4 * 5 = 20     5 * 5 = 25     6 * 5 = 30     7 * 5 = 35     8 * 5 = 40     9 * 5 = 45
2 * 6 = 12     4 * 6 = 24     5 * 6 = 30     6 * 6 = 36     7 * 6 = 42     8 * 6 = 48     9 * 6 = 54
2 * 7 = 14     4 * 7 = 28     5 * 7 = 35     6 * 7 = 42     7 * 7 = 49     8 * 7 = 56     9 * 7 = 63
2 * 8 = 16     4 * 8 = 32     5 * 8 = 40     6 * 8 = 48     7 * 8 = 56     8 * 8 = 64     9 * 8 = 72
2 * 9 = 18     4 * 9 = 36     5 * 9 = 45     6 * 9 = 54     7 * 9 = 63     8 * 9 = 72     9 * 9 = 81
```