



While

Looping the old way

A very common scenario using goto is the following:

```
int playerHealth = 10;
```

```
AttackStart:
```

```
if(playerHealth > 0){ // continue fight, if the player is still alive
```

```
    // simulate fight:
```

```
    int damage = Random.Shared.Next(0, 3);
```

```
    playerHealth -= damage;
```

```
    Console.WriteLine($"Player takes {damage} Damage and has {playerHealth} HP left.");
```

```
    goto AttackStart; // and after those things, we return to the condition again.
```

```
}
```

```
Console.WriteLine("Unfortunately, the player has died.");
```

The better easier way

Which can be reduced to:

```
LoopStart:  
if(condition){  
    // do something  
    goto LoopStart;  
}  
// condition no longer true
```

And for this standard scenario C# has the following keyword:

While

```
while( DoSomething ) {  
    // Do something...  
}  
// Condition no longer true
```

Save us some lines of code

Looking at the previous example again in full context, with while it looks like this:

```
int playerHealth = 10;

while(playerHealth > 0){ // continue fight, if the player is still alive
    // simulate fight:
    int damage = Random.Shared.Next(0, 3);
    playerHealth -= damage;
    Console.WriteLine($"Player takes {damage} Damage and has {playerHealth} HP left.");
}

Console.WriteLine("Unfortunately, the player has died.");
```

You see, it saved us some lines of code :)

Similar to if

You can use the while-keyword in combination with a bool-expression to form

- a loop that repeats
- as long as the bool-expression returns true

The syntax is as follows:

```
while(conditionExpression){ // while-scope-start  
  // put the code here, that you want to repeat while the bool-expression is true  
} // while-scope-end
```

Think about counting for example

In the following example, the numbers 0, 1 and 2 are printed to the console when i reaches the value 3

- the expression $i < 3$
- becomes $3 < 3$
- which is false
- therefore, the loop is interrupted and the code continues outside the loop

```
int i = 0;
while (i < 3)
{
    Console.WriteLine(i); // print i to the console
    i++; // increase i by 1
}
```

Which gives the output of 0, 1 and 2

Oh no, not this guy again!

Here's another example where we ask the user to buy our game.. again :D

```
string userAnswer = "No";

// Repeat the loop while the user has not answered "Yes", yet.
while(userAnswer != "Yes")
{
    // Ask the user to agree.
    Console.WriteLine("Do you want to buy my game? Yes or No?");

    // Assign the user's answer to `userAnswer`
    userAnswer = Console.ReadLine();
}

// We can only get here, if the user has agreed.
Console.WriteLine("Thanks for buying my game.");
```

Goal - Mini nim

- Implement a mini nim game using the while loop
- The game starts with 10 matches
- While there are still matches left..
 - Print how many are left
 - Ask the user to draw 1-3
 - Reduce the number given by the user
- End the game when there are no matches left