## Lesbrief 4: Loops

| * Probeer de voorbeelden uit. * Voer de opdrachten uit. |
| --- |

Soms krijg je te maken met herhalingen, ook wel loops genoemd. Loops voeren een bepaald stukje code herhaaldelijk uit. Er zijn verschillende manieren om loops te programmeren:

* For loops
* While loops
* Do - while loops

### While loop

Een *while* loop voert een stukje code herhaaldelijk uit zolang voldaan wordt aan een conditie.

De syntax van een while loop ziet er als volgt uit:

| *while (conditie) {  // voer deze code uit }* |
| --- |

De code tussen de accolades “{ }” wordt herhaaldelijk uitgevoerd totdat er niet meer aan de conditie wordt voldaan. Bij iedere loop controleert C# de conditie om te bepalen of die de loop - de code tussen de “{ }” - nogmaals moet uitvoeren.

**Voorbeeld**

Onderstaand programma toont een loop die 5 x wordt uitgevoerd en de cijfers van 1 tot en met 5 print.

| using System;  namespace Counter {  class Program  {  static void Main(string[] args)  {  int i = 1;  while (i <= 5)  {  Console.WriteLine(i);  i++;  }  }  } } |
| --- |

### do - while loop

Een *do while* loop doet bijna hetzelfde als een *while* loop met als uitzondering dat de loop minimaal 1 x wordt doorlopen. De syntax ziet er als volgt uit:

| *do {  //code } while (conditie);* |
| --- |

Bij een *while* loop wordt altijd eerst de conditie gecontroleerd alvorens de loop wordt uitgevoerd.

In onderstaand programma zou de loop niet worden uitgevoerd.

| using System;  namespace Counter {  class Program  {  static void Main(string[] args)  {  int i = 6;  while (i <= 5)  {  Console.WriteLine(i);  i++;  }  }  } } |
| --- |

In een *do while* wordt de conditie pas aan het einde van de loop gecontroleerd.

| using System;  namespace Counter {  class Program  {  static void Main(string[] args)  {  int i = 6;  do  {  Console.WriteLine(i);  i++;  } while (i <= 5);  }  } } |
| --- |

De loop wordt dus minimaal 1 x uitgevoerd.

**Voorbeeld**: Raad de geheime code

In onderstaand programma wordt de gebruiker gevraagd om een geheim woord te raden.

| using System;  namespace GuessGame {  class Program  {  static void Main(string[] args)  {  string secretWord = "informatica";  string guessedWord = "";   while (guessedWord != secretWord)  {  Console.Write("Enter guess: ");  guessedWord = Console.ReadLine();   }  Console.WriteLine("You've won!");  Console.ReadLine();  }  } } |
| --- |

| **Opdracht**   * Breidt bovenstaand programma uit zodanig dat de gebruiker 3 x de kans krijgt om het geheime woord te raden. Geef daarbij ook een toepasselijke melding. |
| --- |

### For loop

Een *For* loop voert ook een stukje code herhaaldelijk uit. Een verschil met de while of do-while loop is dat bij een for loop je van tevoren weet hoe vaak er herhaald moet worden.

De syntax ziet er als volgt uit:

| *for (initializer ; conditie ; iterator) {  //code }* |
| --- |

**Voorbeeld**

| using System;  namespace Counter {  class Program  {  static void Main(string[] args)  {  for (int teller = 1; teller <= 5; teller++)  {  Console.WriteLine(teller);  }  }  } } |
| --- |

De code bij de initializer wordt 1 x uitgevoerd. Hiermee initialiseer je een variabele. In bovenstaand voorbeeld wordt de variabele “teller” aangemaakt en krijgt als beginwaarde 1.

Let op, de variabele “teller” bestaat alleen in de for-loop.

Na de initialisatie volgt de conditie. De conditie wordt bij iedere loop gecontroleerd.

De iterator bepaald wat er na het uitvoeren van de loop gebeurt. In bovenstaand voorbeeld wordt de teller opgehoogd met 1.

| **Opdracht**   * Verander de GuessGame, gebruik een for loop om de gebruiker 3 x de kans te geven het geheime woord te raden. |
| --- |

## 