

Controlling Program Flow with Iteration Statements



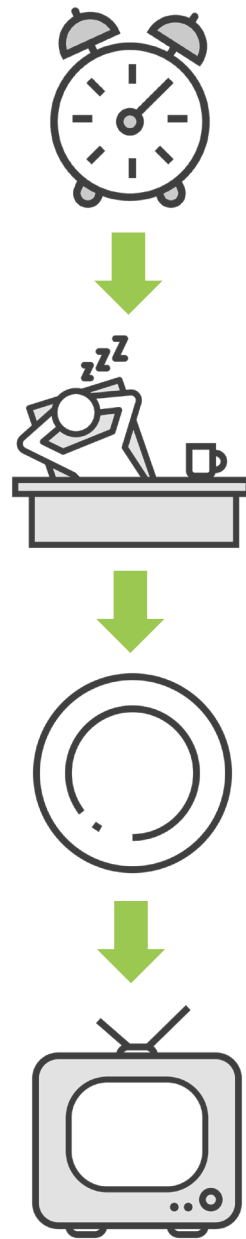
Alex Wolf

Software Developer

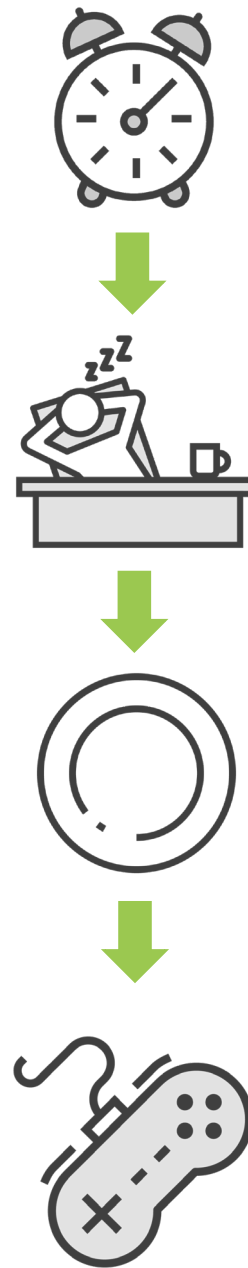
www.crywolfcode.com



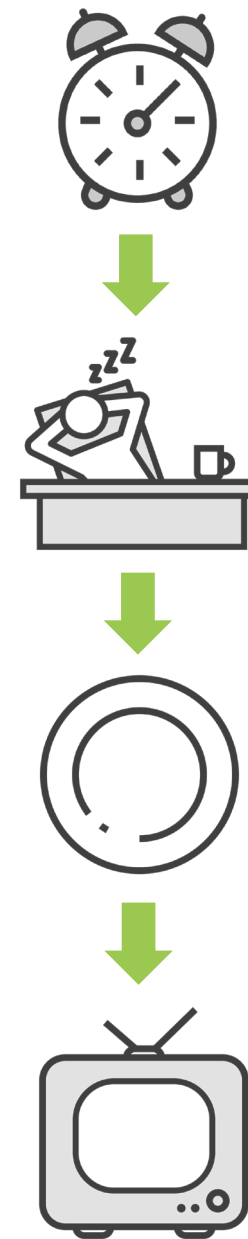
Identifying Repetitive Routines



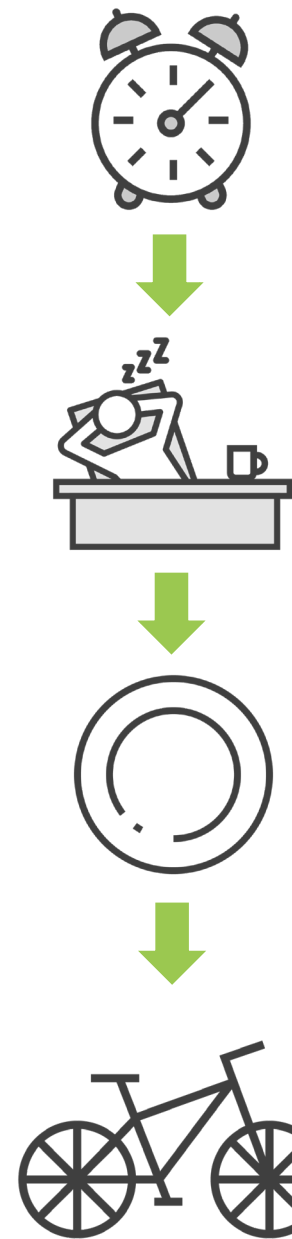
Monday



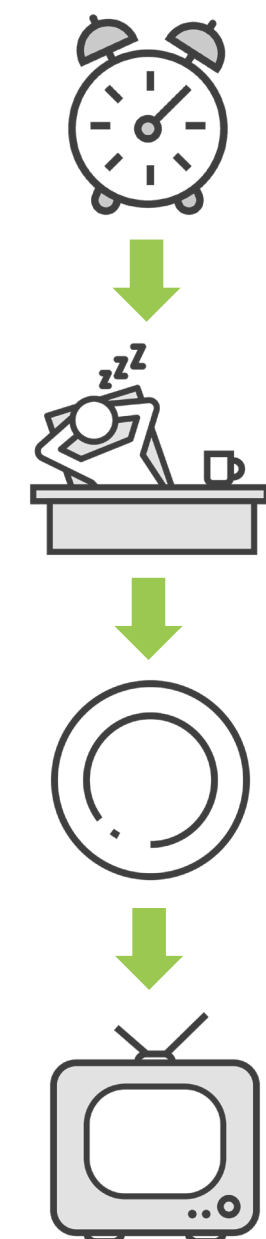
Tuesday



Wednesday



Thursday



Friday



Describing Loops in Daily Life

While it's a weekday...

Phil repeats these tasks...

Until it's the weekend



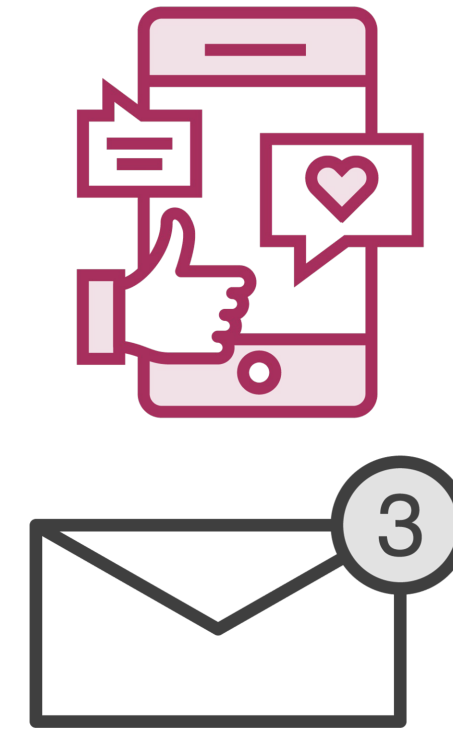
Repetitive Tasks in Software Development



Scan through
discussion comments



Process items in a
shopping cart



Send many
personalized messages



```
EmailCustomer( "123@sample.com" );
```

```
EmailCustomer( "456@sample.com" );
```

```
EmailCustomer( "789@sample.com" );
```

```
EmailCustomer( "101@sample.com" );
```

```
EmailCustomer( "155@sample.com" );
```

```
// What if we have thousands?
```

◀ Tedious repetition..

◀ ...eventually becomes impossible to manage

Iteration Statements

Statements that repeatedly loop through a block of code until a condition is met.



Types of Loops in C#



For Loop

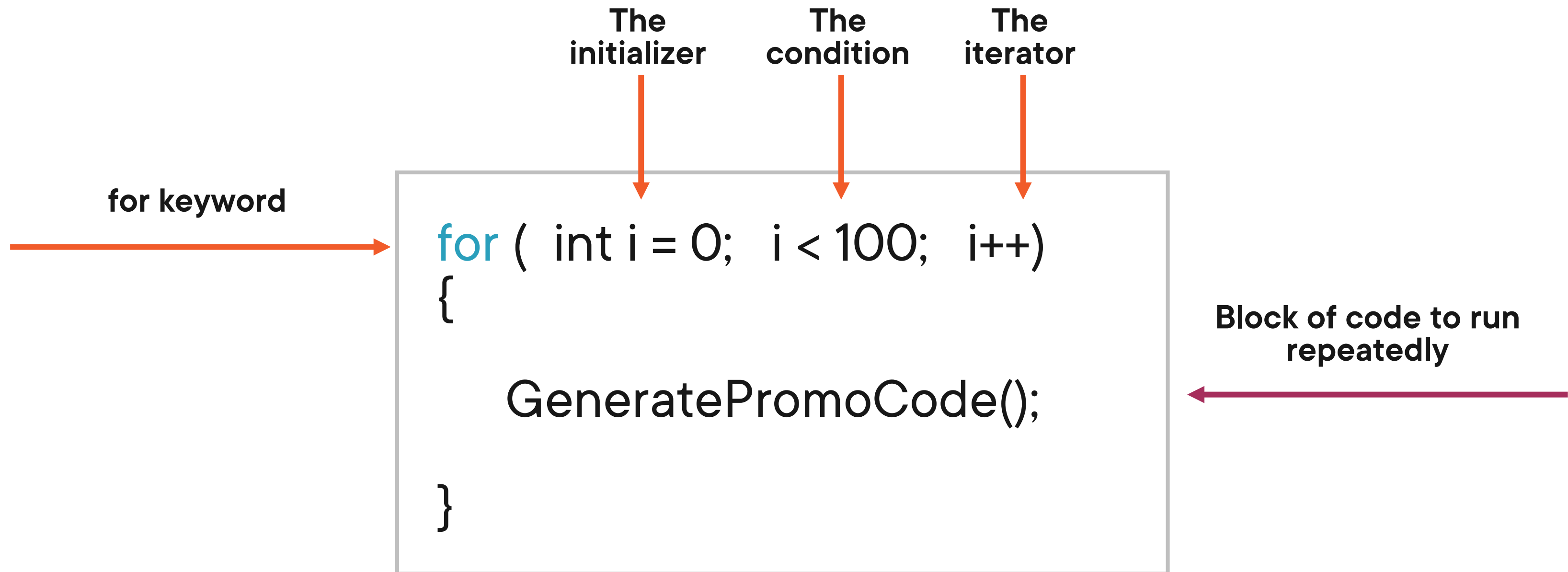
Foreach Loop

While Loops

Do...While Loop



The Structure of a For Loop



Mapping Activities to Loops

- 1) Starting on Monday... 2) Verify it's a week day... 4) Then start a new day

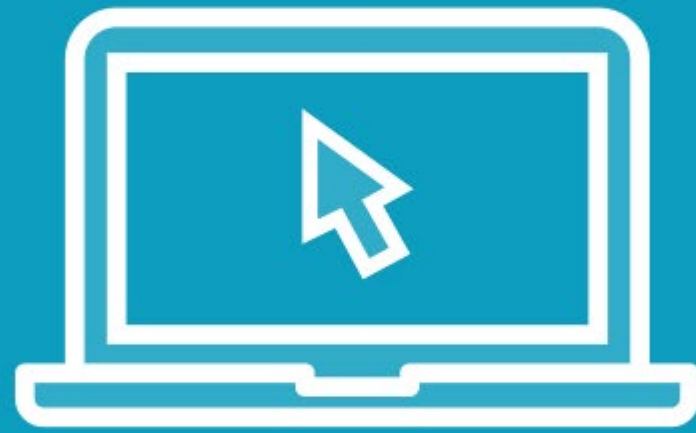
```
for ( int i = 1; i < 6; i++ )  
{  
    WakeUp();  
    GoToWork();  
    MakeDinner();  
    Relax();  
}
```



- 3) Then complete these tasks



Demo



Processing a Set of Data Using a For Loop



Exploring Different Types of Loops



Types of Loops in C#

For Loop

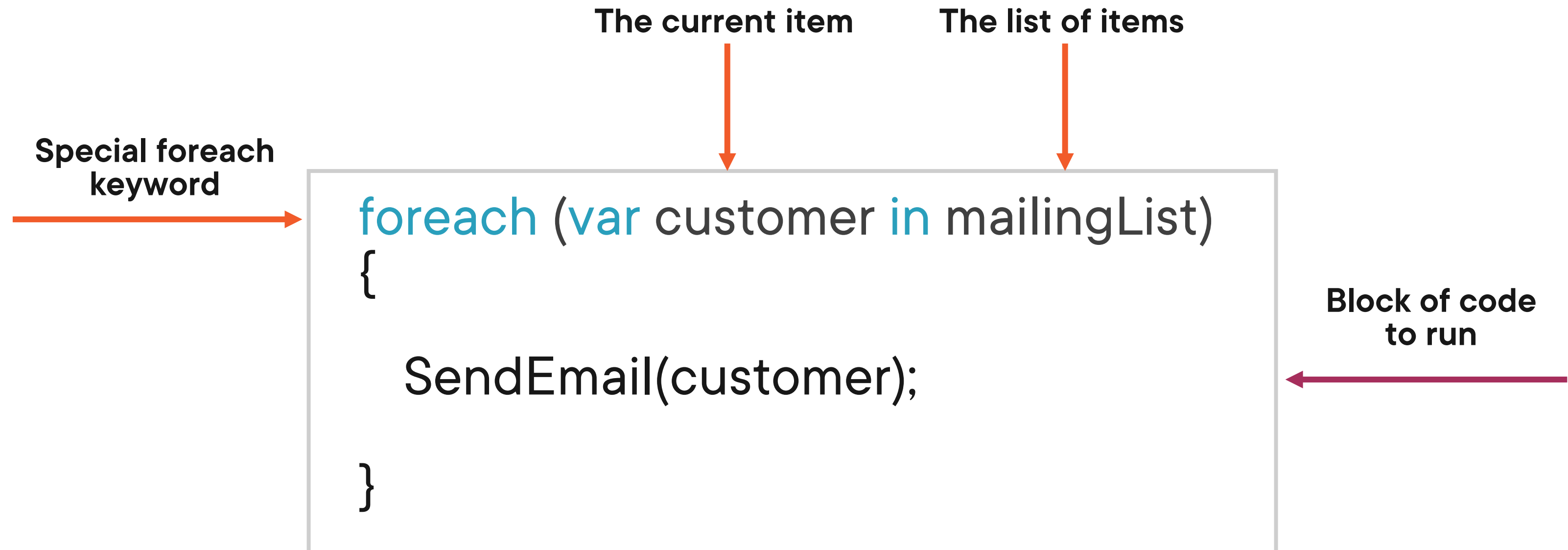
Foreach Loop

While Loops

Do...While Loop



Simplifying Loops with Foreach

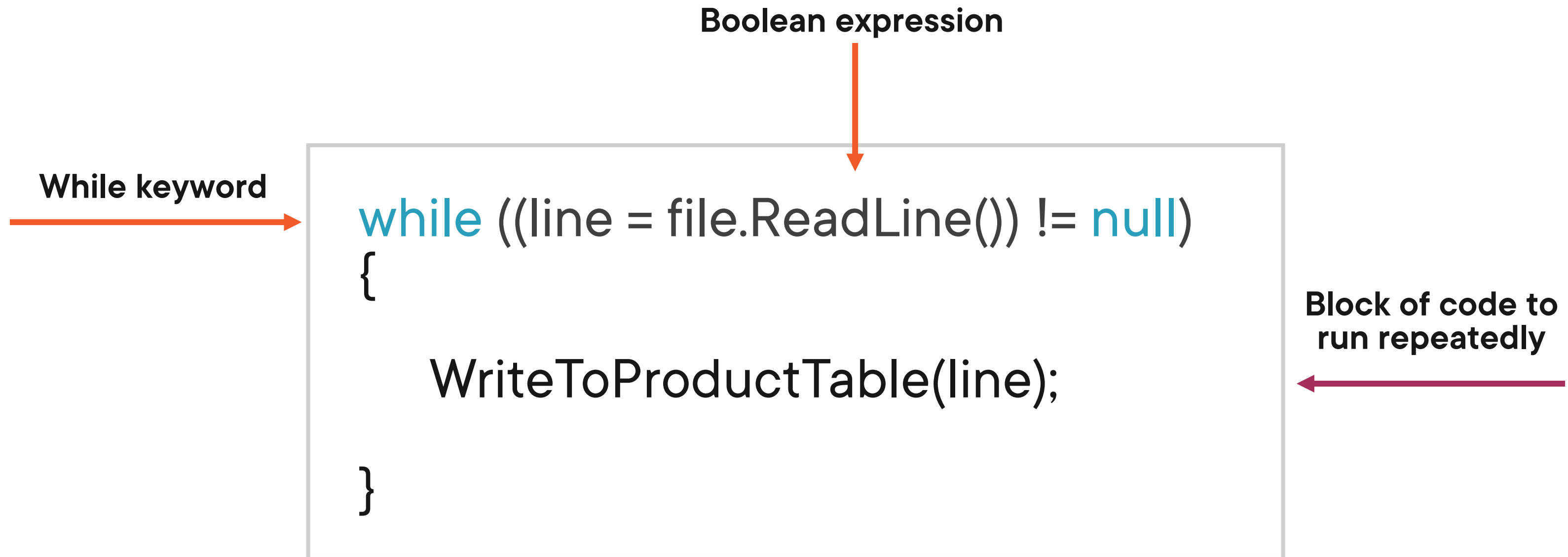




A note about foreach loops.



Understanding While Loops



Examples of While Loops

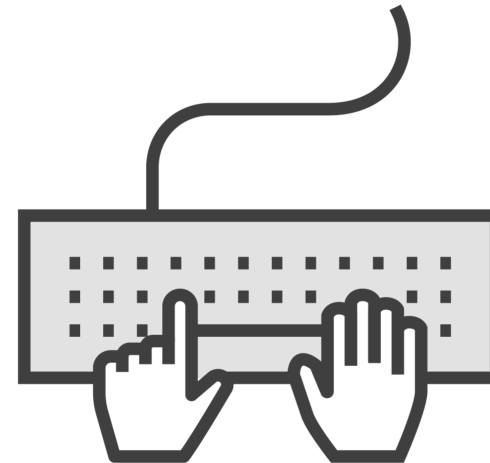
While the button
is pressed...



...the character runs.



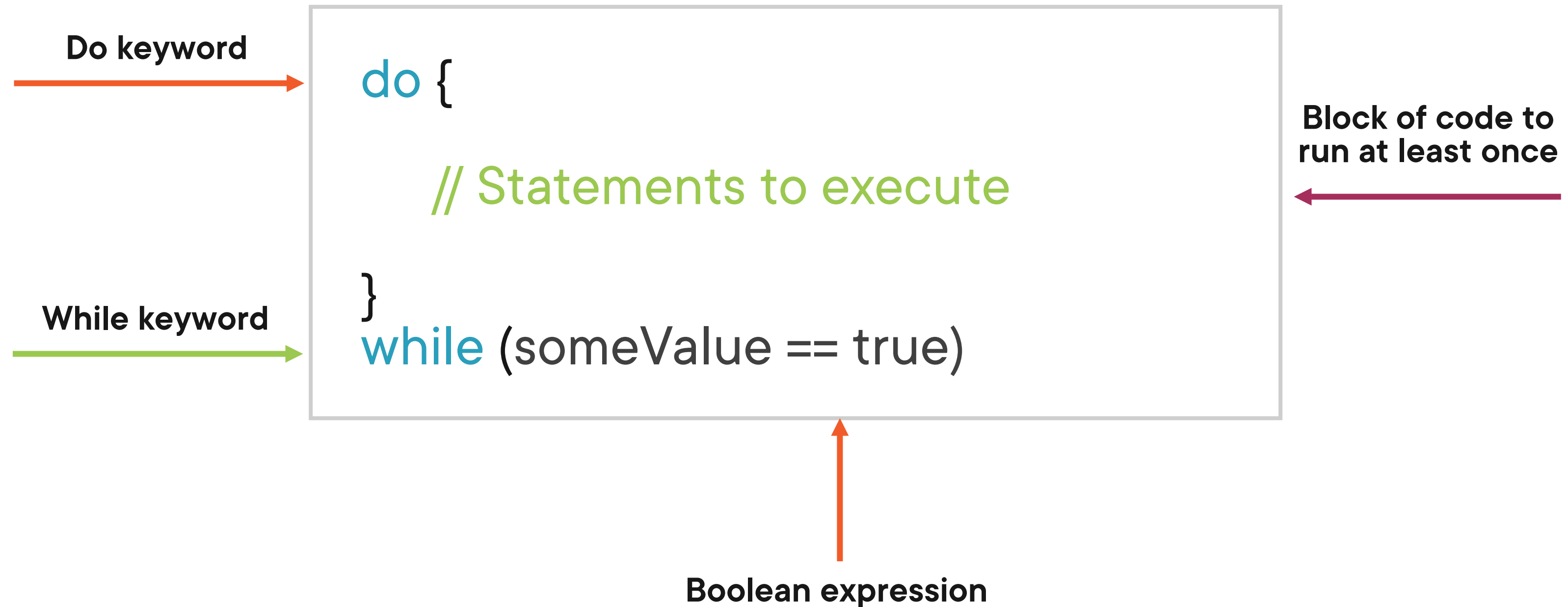
While the key is
held down...



...draw more data.




Understanding Do-While Loops



Breaking out of Loops

Break keyword



```
foreach (var customer in mailingList)
{
    break;

    SendEmail(customer);
}
```





A note about Linq.



Demo



Simplifying Loops with Foreach



Demo



Working with While Statements



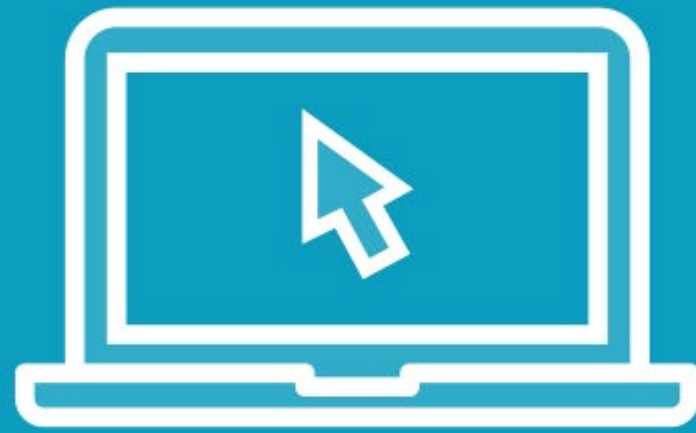
Demo



Understanding Variable Scope



Demo



Organizing the Application Reports



Overview/ Summary



- **Iteration Statements** allow us to repeat a block of code many times
- **For Loops** rely on an incrementing variable to repeatedly execute code while a condition is true
- **Foreach Loops** provide a simpler syntax alternative to For Loops in common use cases
- **While Loops** continue to repeat a block of code as long as a condition is true
- **Do While Loops** are the same as While Loops, but guarantee at least one execution
- **Variables** can be used in the code block they are declared in, as well as any nested code blocks

