Control Flow

Control Flow

Sequential lines of code are evaluated in a row.

To make decisions and take different paths we need `control flow`

If Statements

Simplest of the `control flow` statements.

Basic structure:

```
if (someBooleanCondition) {
  // Code to run if the condition is true
}
```

If / Else statements

Can also handle the case when the condition is false

```
if (someBooleanCondition) {
  // Code if the condition is true
}
else {
  // Code if the condition is false
}
```

Boolean Conditions

Examples of boolean conditions

Kind	Example
equality	`name == "Paul"`
inequality	`name != "Paul"`
greater than	`score > 90`
less than	`score < 60`
greater or equal to	`score >= 90`
less or equal to	`score <= 60`

Example

```
if (name == "Paul") {
  Console.WriteLine("Greetings!");
} else {
  Console.WriteLine("Wait, who are you?");
}
```

Multiple Booleans: `Or`, `And`

Combine two conditions.

`Or` is written as `||` while `And` is written as `&&`

Condition that is true if the score is lower than `20` **OR** greater than `90`:

```
if (score < 20 || score > 90) {
  // Some code here
}
```

Condition that is true if the person is named `Paul` **AND** the score is more than `85`:

```
if (name == "Paul" && score > 85) {
  // Some code here
}
```

Truth Table

A && B

A	B	Result
T	T	T
T	F	F
F	T	F
F	F	F

A || B

A	B	Result
T	T	T
T	F	T
F	T	T
F	F	F

Chaining

A series of `if/else` statements can be chained.

```
if (name == "Paul") {
  Console.WriteLine("Here");
} else if (name == "Dorothy") {
  Console.WriteLine("Also here");
} else if (name == "Sam") {
  Console.WriteLine("Here again");
} else {
  Console.WriteLine("Didn't find anything");
}
```

We can do better

Introducing `switch`

```
switch (name)
{
    case "Paul":
        Console.WriteLine("Here");
        break;
    case "Dorothy":
        Console.WriteLINE("Also Here");
        break;
    case "Sam":
        Console.WriteLine("Here Again");
        break;
    default:
        Console.WriteLine("Didn't find anything");
        break;
}
```

Other neat `switch` features

We can handle multiple values by repeating the case statement:

In this code we will see the message `Here` for `name` if it is either `Paul` **OR** `Peter` **OR** `Mary`

```
switch (name)
{
    case "Paul":
    case "Peter":
    case "Mary":
        Console.WriteLine("Here");
        break;
    case "Dorothy":
        Console.WriteLine("Also Here");
        break;
    case "Sam":
        Console.WriteLine("Here Again");
        break;
    default:
        Console.WriteLine("Didn't find anything");
        break;
}
```

Case conditionals

Let's say we are working with an `int` variable named `score` and we wanted to print a grade associated to a score.

```
var score = 95;

switch (score)
{
    case < 65:
        Console.WriteLine("F");
        break;
    case < 70:
        Console.WriteLine("D");
        break;
    case < 80:
        Console.WriteLine("C");
        break;
    case < 90:
        Console.WriteLine("B");
        break;
    case >= 90:
        Console.WriteLine("A");
        break;
    default:
        Console.WriteLine("Hmmm, I don't recognize this score");
        break;
}
```

What about repeating code?
What about repeating code?
What about repeating code?
What about repeating code?
What about repeating code?

Loops

while

While

The `while` statement repeats the code inside the { } braces as long as the condition supplied remains true.

Ask the user their name and greet them until the user enters the text `quit`. The code would look like this:

```
Console.Write("What is your name? ");
var name = Console.ReadLine();

while (name != "quit") {
  Console.WriteLine($"Hello {name}");

  Console.Write("What is your name? ");
  name = Console.ReadLine();
}
```