Lesson 5

Conditionals Use for control logic in a program. There are several types

- if statments
- if-else statements
- if-else else statments

```
// if statements

int y = 16;

if (y== 16) {
    Console.WriteLine("the number is 16!");

}

Console.WriteLine("Conditionals are cool");
```

```
// if else statements

if (y < 16){
    Console.WriteLine("Your number is lower than 16");
}
else{
    Console.WriteLine("Your number is higher than 16");
}
```

```
// chaining if else - else

string name = "Jake";
if (name == "Jake"){
    Console.WriteLine("Hi, the name is Jake");
}
else if (name == "John"){
    Console.WriteLine("The name is John");
}

else {
    Console.WriteLine("Name is Unknown");
}
```

```
// Ternary conditional operator - short hand if else statements

int x = 4;
// new variable, equal to old variable, output based on value of old variable

string z = x == 4? "4.0" : "3.0";
Console.WriteLine(z);
```

Fun fact this was part of my interview questions for my current job!

Switch Statments

- Fancy if statments
- you define cases and need a default case for the undefined specific cases

```
// See https://aka.ms/new-console-template for more information
// use switch statements for big if else clauses
Console.WriteLine("Enter your age");
// remember defining string that could be null using ?
string? input = Console.ReadLine();
// string parsing is back and we converting it to an int!!!!!
int age = int.Parse(input);
// define switch and what variable will be compared
switch (age)
{
// example of a case
 case < 12:
    Console.WriteLine("You are a child");
    break;
 case < 18:
    Console.WriteLine("You are teenager");
    break;
 case < 21:
    Console.WriteLine("You cannot drink");
    break;
 case >= 30:
    Console.WriteLine("You are old");
    break;
 // what your program will return if you don't handle it another case
 default:
  Console.WriteLine("Every birthday is a blessing");
  break;
}
```

iterations: the back of bone of programming

three major types:

- for
- for each
- while
- do -while

extra stuff on while loop vs do-while

```
// while loop

int a = 2;
while (a < 5)
{
    Console.WriteLine($"while: {a}");
    a++;
}

// do-while

int b = 1;

do
{
    Console.WriteLine($"do-while: {b}");
    b++;
} while (b < 4);


// for loop

for (int i =1; i < 5; i++){
    Console.WriteLine($"for loop: {i}");

}
// for each
List<int> fibNumbers = [0, 1, 1, 2, 3, 5, 8, 13];
foreach (int element in fibNumbers)
{
    Console.Write($"{element} ");
}
```