Conversions

Implict Conversion: No special syntax no data loss no precision loss Explicit conversions: Castings potential data loss, pootential precision loss

```
int age = 31;
float weight = 78.7f;


Console.WriteLine($"age: {age}, weight: {weight}");
```

Typing Casting

casting one variable type into another

```
//implicit type casting

long myAge = age;
Console.WriteLine(myAge);

//explicit conversions

int lessWeight = (int)weight; // precision loss
int notMyAge = (int)myAge; // no precission loss

Console.WriteLine($"age: {notMyAge}. weight: {lessWeight}");

// values don't fit if converted into a size smaller than them and give bad
outputs
// try this to see it
long bigNumber = 9223372036854775807;
int smallType = (int)bigNumber;
Console.WriteLine(smallType);
```

More fun with strings

```
/// strings
string message = "Hello World";
Console.WriteLine(message);

//empty string
// this is important because string have a native method to return an empty value
// empty is different than null and can have meaning
```

```
string emptyMess = string.Empty;
Console.WriteLine(emptyMess);

// how to esacpe a special character using '\'
string FilePath = "C:\\temp";
Console.WriteLine(FilePath);


// right way to do file paths. as they are special using the @ sign allows us to
do that
// https://www.dotnetperls.com/path - more info on paths
string FilePath2 = @"C:\temp";
Console.WriteLine(FilePath2);

// how do i combine two string together???
// string concentation
string firstName = "Jake";
string lastName = "Baggs";

string fullName = firstName + " " + lastName;
Console.WriteLine(fullName);


//how to get a string's using native methods length
Console.WriteLine(FilePath.Length);


// string interpolation aka how do we use vairables in our strings
string fullName2 = $"{firstName} {lastName}";
Console.WriteLine(fullName2);


// substrings - rip out part of a string
// remember index starting position than range of how far
// however over method if you get confused
string first3char = fullName.Substring(0,3);
Console.WriteLine(first3char);

// string parsing
string luckyNum = "16";
// lets up convert a string representation of a number to an int
int parseLuckyNum = int.Parse(luckyNum);
Console.WriteLine(parseLuckyNum);
double temperature = double.Parse("30.7");
Console.WriteLine(temperature);
```