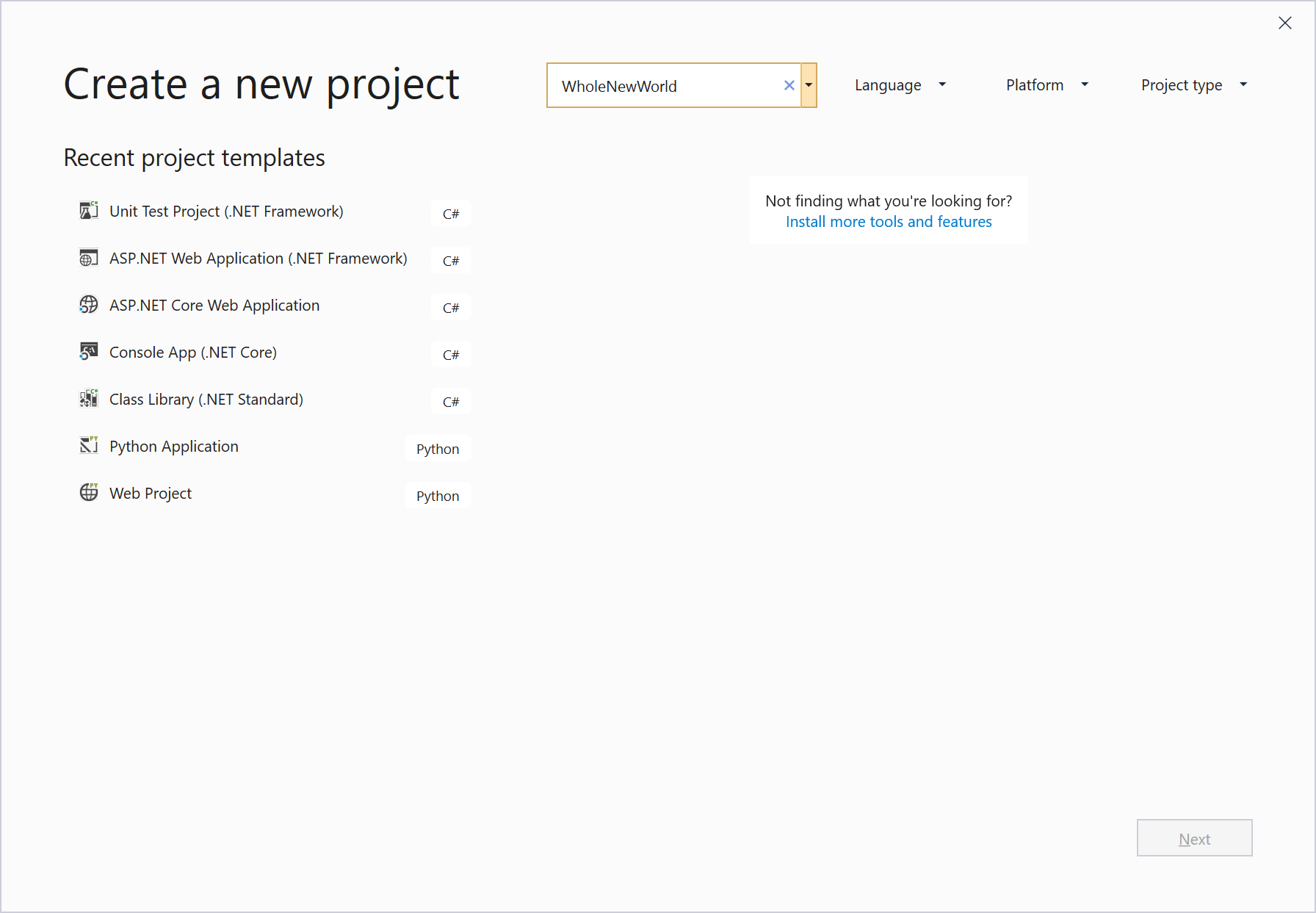
## **Question: What do you mean by .Net?**

OK, bigger response here. GOTO the take it from these guys link: <https://dotnet.microsoft.com/learn/dotnet/what-is-dotnet-framework>

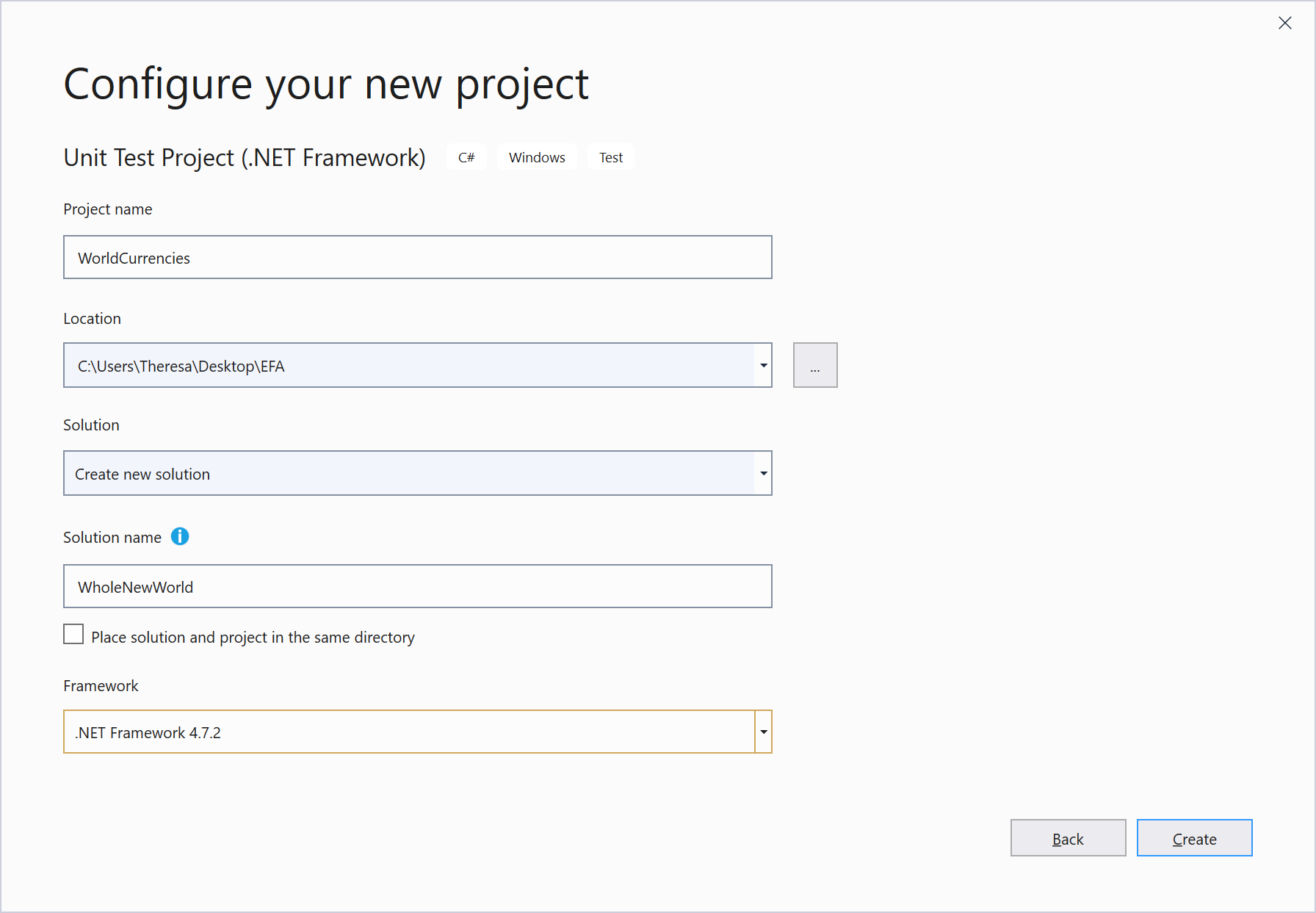
**Question: OK, so what exactly is the relationship between namespace, solution, assembly and project, on the .Net platform?**

Well, it’s all about organization. Let’s

A solution, is the whole shebang, the whole grouping of code you are creating when you select *File*, and then *New project*, from the Visual Studio menu bar. It is also referred to as the namespace. Below, WholeNewWorld, is the solution or namespace.



To the solution, you can add a series of tests, and a number of classes or class libraries, as well as user interface components and web components. These sub-projects are also referred to as assemblies. Below, WorldCurrencies, would be an assembly.



Remember, the solution holds the assemblies.

**Question: What’s a type?**

Computers work with data, units of information that describe numbers, or names and addresses, and much more. A type tells you what a piece of data can do.

For instance, numeric data is all about numbers. You can add, subtract, and do math stuff with numbers. You can get the percentage 16 is of 32. However, with data like “Dublin is the best city ever”, you cannot expect to get a percentage, or be able to divide or multiply.

Computer programming languages use types to help develop working code. If you are doing numeric calculations, then you use numeric types.

**Question: What’s a variable?**

Classic definition for a variable is ‘named location in computer memory’. Your computer has distinct ways of storing information - temporary storage called memory (you can get technical and call it RAM), and permanent storage. The RAM (Random Access memory) is a series of cells that store information. These cells have addresses, much like your own address, 12 Clonmel Court …., Well, if you create a variable, you are usurping these cells, and calling them by a name. A variable is a convenience for accessing the information, and you typically have to give it an associated type.

**Question: OK, so now I know what a type is (I think I do), what are the types I can use with C#?**

C#, like many other modern programming languages, organizes it’s types into 2 umbrella categories, value types and reference types. Traditionally in computer programming, value types have been the different categories of numbers, like whole numbers or ones that have a decimal place, and single characters, and also boolean values (true/false). A variable of a value type contains a single value of the type. Reference types have been the ones that reference or point to, a memory address, and there is no ambiguity here. You will see later that things called objects are references to memory locations.

So, value types in C#? Don’t freak out on me, in C# the 2 broad categories of value types are Struct and Enumeration. If you feel like some bedtime reading, have a look at this, from the Microsoft documentation:

<https://docs.microsoft.com/en-us/dotnet/csharp/language-reference/keywords/value-types>

In C#, the simple value types are described here: <https://docs.microsoft.com/en-us/dotnet/csharp/language-reference/language-specification/types#simple-types>

Take Byte, for example. The range of values that a variable of type Byte can have, is 0 to 255.

There are 2 distinct floating point types, float and double. Again, these represent distinct ranges.

Decimal was created for financial calculations (and you want precision here), and has it’s own range of values – bigger than float and double.

**What is that f suffix? D suffix?**

Just an indication that a number is of a particular type,

float num1 = 123.34f;

double num2 = 123.45d;

**What’s a char?**

char is a value type that represents a single character, like ‘a’, ‘&’.

**What’s meant by Strongly Typed?**

There are lots of languages out there, and they treat types differently. C# is strongly typed, which means that when you declare a variable of type int, and try to do an operation with these 2 numbers, you will get told off by C#. Try the following yourself:

int num1 = 3;

float num2 = 4f;

int num3 = num1\*num2;

Did you get something like ‘Cannot implicit type double to int. An explicit conversion exists. Are you missing a cast’?

**Question: What’s a cast?**

This is where you explicitly convert a quantity to a different type. In the example above, you might cast or convert the result to be an int.

int num3 = (int)(num1\*num2)

Try it yourself!