C# =>

1. Interfaces

An interface is a completely "**abstract class**", which can only contain abstract methods and properties (with empty bodies). Another way to achieve [abstraction](https://www.w3schools.com/cs/cs_abstract.asp) in C#, is with interfaces.

1. Extension Methods

Extension methods enable you to "add" methods to existing types without creating a new derived type, recompiling, or otherwise modifying the original type.

1. Func and Action delegates

A delegate is a type-safe function pointer that can reference a method that has the same signature as that of the delegate. Delegates are used to define callback methods and implement event handling

The basic difference between Func and Action delegates is that while the former is used for delegates that return value, the latter can be used for those delegates in which you don't have any return value.

Func is a delegate that points to a method that accepts one or more arguments and returns a value. Action is a delegate that points to a method which in turn accepts one or more arguments but returns no value.

1. var keyword

An implicitly typed local variable is strongly typed just as if you had declared the type yourself, but the compiler determines the type.

1. LINQ

Language-Integrated Query

var nQuery =

from VrNum in n1

where (VrNum % 2) == 0

select VrNum;

var nQuery =

from VrNum in n1

where VrNum > 0

where VrNum < 12

select VrNum;

Web =>

1. HTTP Protocol: An [application layer](https://en.wikipedia.org/wiki/Application_layer) protocol for distributed, collaborative,[hypermedia](https://en.wikipedia.org/wiki/Hypermedia) information systems.
2. GET: The GET method requests a representation of the specified resource. Requests using GET should only [retrieve data](https://en.wikipedia.org/wiki/Data_retrieval) and should have no other effect.
3. POST: The [POST method](https://en.wikipedia.org/wiki/POST_(HTTP)) requests that the server accept the entity enclosed in the request as a new subordinate of the [web resource](https://en.wikipedia.org/wiki/Web_resource) identified by the URI. The data POSTed might be, for example, an annotation for existing resources; a message for a bulletin board, newsgroup, mailing list, or comment thread; a block of data that is the result of submitting a [web form](https://en.wikipedia.org/wiki/Form_(HTML)) to a data-handling process; or an item to add to a database.
4. PUT: The PUT method requests that the enclosed entity be stored under the supplied [URI](https://en.wikipedia.org/wiki/Uniform_Resource_Identifier). If the URI refers to an already existing resource, it is modified; if the URI does not point to an existing resource, then the server can create the resource with that URI.
5. DELETE: The DELETE method deletes the specified resource.
6. Http Status Codes: Status codes are issued by a server in response to a client's request made to the server.  404
7. Http Form Tag <form>

Dependency Injection Pattern -- Very very important

a technique in which an [object](https://en.wikipedia.org/wiki/Object_(computer_science)) receives other objects that it depends on. These other objects are called dependencies. In the typical "using" relationship the receiving object is called a [client](https://en.wikipedia.org/wiki/Client_(computing)) and the passed (that is, "injected") object is called a [service](https://en.wikipedia.org/wiki/Service_(systems_architecture)). The code that passes the service to the client can be many kinds of things and is called the injector. Instead of the client specifying which service it will use, the injector tells the client what service to use. The "injection" refers to the passing of a dependency (a service) into the object (a client) that would use it.

<https://visualstudio.microsoft.com/vs/mac/>

<https://www.jetbrains.com/rider/>

<https://docs.microsoft.com/en-us/sql/linux/quickstart-install-connect-docker?view=sql-server-ver15&pivots=cs1-bash>

<https://hub.docker.com/_/microsoft-mssql-server>

https://dotnet.microsoft.com/