**Building a Web App with ASP.NET Core, MVC, Entity Framework Core, Bootstrap, and Angular:**

**Getting Started:**

**Why ASP.NET Core?**

* The Web Stack is \*old\*!
* Weighed down by old, unused code
* Machine level upgrades is problematic
* Had Reached a Ceiling of Performance

**What Is ASP.NET Core?**

* Complete re-write of the platform
* Cross-platform and open source
* Single platform for MVC and WebAPI
* Everything is a dependency
* Low memory footprint
* Multiple deployment (Bereitstellung) support
* Cloud, IIS, Self-Hosting, etc.
* Frameworks in which it can run of:
* .Net 4.x
* The .Net you´ve always known
* Limited to Windows Only
* .Net Core
* The Cross-platform .NET Framework
* Linux, Mac and Windows support
* Completely Composed
* That means that everything above the .NET core layer is a Package
* Frameworks are the bootstrap and CLR
* MVC, Logging, Identity are just packages
* Everything is optional
* Embraces Open Web Development
* Uses existing web development tooling
* NPM for tooling support
* Bower/NPM for client-side libraries
* Grunt and Gulp for building automation
* NuGet for .NET Packages
* None of it is required

**Installing ASP.NET Core**

Dot.net -> will take u directly to the homepage

**Hello World!**

* He is creating the code via command prompt –

**Creating a Project with Visual Studio**

**Serving Your First File**

**HTML5 and CSS Basics:**

**What Is HTML?**

* HyperText Markup Language
* Angle bracket based structure
* Derived from SGML

<html>

<head></head>

<body>

<h1>Hello World</h1>

</body>

</html>

* Angle-Brackets != XML
* Not case sensitive
* Mixed closing rules

HTML is basically the structuring language for the webApp

**HTML Basics**

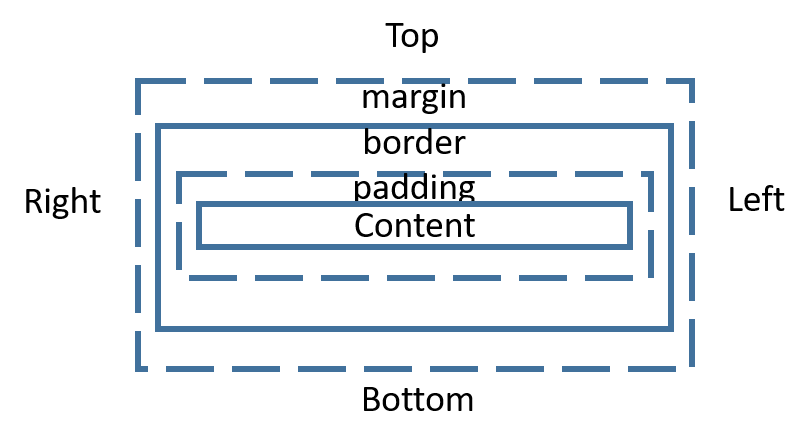
**HTML Forms**

**CSS Basics**

**CSS Naming**

**CSS Classes**

**The Box Model**



**Getting Started with JavaScript:**

**What Is JavaScript?**

* Object Oriented
* Prototypical Inheritance instead of Classes
* Dynamic
* Not type-less. But dynamically typed
* Compiled?
* Just-in-time compiled
* Without an intermediate format (e.g. MSIL, Bytecode)

**Adding a JavaScript File**

**JavaScript Events**

**Using NPM**

**Introducing jQuery**

**Practical jQuery**

**Using ASP.Net MVC:**

**What Is MVC?**

Server-side Model-View-Controller Framework:

* Model -> represents the Data
* Controller -> represents the logic (e.g. requests…)
* View -> represents the “Mockup”

**First Controller/View**

**Enabling MVC 6**

**Creating a Layout**

**Adding More Views**

**Using Tag Helpers**

**Razor Pages**

**Implementing a Contact Page**

**Model Binding**

**Using Validation**

**Adding a Service**

**Using Bootstrap to Prototype the App:**

**Building the Database with Entity Framework Core:**

**Creating the API with ASP.NET MVC:**

**ASP.NET Core Identity:**

**Brief Introduction to TypeScript:**

**Client-side Development with Angular:**

**Using Services in Angular:**

**Routing with Angular:**

**Deploying ASP.NET Core Apps:**