Web Technology Note (UNITE 1-3)

WWW: The World Wide Web is a hypermedia system. It has largely achieved the goal of Tim Berners-Lee, its British inventor. Tim Berners- Lee invented the World Wide Web in October 1994

It is a system of interlinked hypertext documents accessed via the Internet.

The World Wide Web, or simply Web, is a way of accessing information over the medium of the Internet. It is an information- sharing model that is built on top of the Internet. The Web uses the HTTP protocol, only one of the languages spoken over the Internet, to transmit data. Web services, which use HTTP to allow applications to communicate in order to exchange business logic, use the Web to share information. The web also utilizes browsers, such as Internet Explorer or Firefox, to access Web documents called Web pages that are linked to each other via hyperlinks. Web documents also contain graphics, sounds, text and video.

WEB PAGE: A web page is a document commonly written in Hyper Text Markup Language (HTML) that is accessible through the Internet network using a browser. A web page is accessed by entering a URL address and may contain text, graphics, and hyperlinks to other web pages and files.

WEB SITE: A connected group of pages on the World Wide Web regarded as a single entity, usually maintained by one person or organization and devoted to a single topic or several closely related topics.

URI: Uniform Resource Identifier (URI) is a string of characters used to identify the name of a resource. Such identification enables interaction with representations of the resource over a network, typically the World Wide Web, using specific protocols.

URL: Web browsers request pages from web servers by using a URL.

The URL is the address of a web page, like: www.texassifal.com This provides

location and protocol which is http part.

WEB SERVER: Web servers are computers that deliver (serves) Web pages. Every Web server has an IP address and possibly a domain name. A Web server is a program that uses HTTP (Hypertext Transfer Protocol) to serve the files and form pages to users, it responses to their requests, which are forwarded by their computers' HTTP clients. Dedicated computers and appliances may be referred to as Web servers as well.

Web client: It typically refers to the Web browser in the user's machine.

Web Browser: Browsers are software programs that allow you to search and view the many different kinds of information that's available on the World Wide Web. The information could be web sites, video or audio information.

SMTP: Simple Mail Transfer Protocol (SMTP), a protocol for sending e-mail messages between servers. Most e- mail systems that send mail over the Internet use SMTP to send messages from one server to another; the messages can then be retrieved with an e-mail client using either POP or IMAP. In addition, SMTP is generally used to send messages from a mail client to a mail server. This is why you need to specify both the POP or IMAP

(Internet Message Access Protocol**)** server and the SMTP server when you configure your e-mail application. SMTP by default uses TCP port.

POP: Post Office Protocol (POP) is an application-layer Internet standard protocol used by local e-mail clients to retrieve e-mail from a remote server over a TCP/IP connection.

Introduction to HTML

HTML is a language for describing web pages. It is not a programming language. A markup language specifies the layout and style of a document. The browser does not display the HTML tags, but uses the tags to interpret the content of the page.

- HTML stands for Hyper Text Markup Language
- A markup language is a set of markup tags
- The tags describe document content
- HTML documents contain HTML tags and plain text
- HTML documents are also called web pages

HTML Tags

- HTML tags are keywords (tag names) surrounded by angular brackets like <html>
- HTML tags normally come in pairs like and
- The first tag in a pair is the start tag, the second tag is the end tag
- The end tag is written like the start tag, with a forward slash before the tag name
- Start and end tags are also called opening tags and closing tags

HTML Images <img> Tag and the Src Attribute

Syntax for defining an image: <img src="*url*" alt="*some_text*">

HTML Tables:

Tag	Description
<table>	Defines a table
<th>	Defines a header cell in a table
<tr>	Defines a row in a table
<td>	Defines a cell in a table
<thead>	Groups the header content in a table
<tbody>	Groups the body content in a table

HTML Lists:

The most common HTML lists are ordered and unordered lists:

List properties:

 - An unordered list. This will list items using plain bullets.
 - An ordered list. This will use different schemes of numbers to list your items.

```
<html>
<head>
<title>HTML Unordered List</title>
</head>
<body>
  <ul type="square">
  <li>Beetroot</li>
  <li>Ginger</li>
  <li>Potato</li>
  <li>Radish</li>
  </ul>
</body>
</html>
```

This will produce following result:

- Beetroot
- Ginger
- Potato
- Radish

HTML Ordered Lists

If you are required to put your items in a numbered list instead of bulleted then HTML ordered list will be used.

Example

```
<html>
<head>
<title>HTML Ordered List</title>
</head>
<body>
<ol>
<li>Beetroot</li>
<li>Ginger</li>
<li>Potato</li>
<li>Radish</li>
</ol>
</body>
</html>
```

This will produce following result:

1. Beetroot
2. Ginger
3. Potato
4. Radish

HTML Forms:

HTML forms are used to pass data to a server. An HTML form can contain input elements like text fields, checkboxes, radio-buttons, submit buttons and more.

A form has two duties: to collect information from the user and to send that information to a separate web page for processing. For example, whenever you submit personal information to a web, you are using a form. Or whenever you type the keyword into your search engine, you are using a form. Forms are the heart and soul of the World wide web.

HTML Forms - The Input Element

Attribute value	Description
Text	A text field
Password	A password text field where each keystroke appears as an *
Button	A new button other than submit and reset button
Checkbox	A checkbox
Radio	A radio button
Reset	A reset button
Submit	A submit button
Select	A selection list
TextArea	A multiline text entry field

Creating Forms:

A form is created using a <form> tag

<form method=POST action="page.php">

Method:

The method tag can be set either GET or POST

GET: GET is used to request data from a specified resource. With this method, the form data will be encoded in a URL. It is composed of the name of the page or script to be loaded

POST: POST method sends the data captured by form element back to the web server as a separate bit stream of data. When there is a large amount of data to be send back to the web server, this method is used.

Action:

The action tag specifies what page will process the information entered by the user. The server side program that process this data can be written in any scripting language that web server understand.

Cascading Style Sheet (CSS)

CSS was first proposed by **Hakon Wium Lie** on October 10, 1994. At the time, Lie was working with Tim Berners-Lee (father of Html) at CERN. The European Organization for Nuclear Research is known as CERN. Hakon wium lie is known as father of css.

CSS was proposed in 1994 as a web styling language, to solve some of the problems of Html 4.

There were other styling languages proposed at this time, such as Style Sheets for Html and JSSS but CSS won.

Include properties in CSS2

CSS level 2 specification was developed by the W3C and published as a recommendation in May 1998. CSS 2 includes a number of new capabilities like below;

- ✓ **absolute**
- ✓ **relative**
- ✓ **fixed**
- ✓ **positioning**
- ✓ **z-index**
- ✓ **concept of media type**
- ✓ **bidirectional text**
- ✓ **new font properties such as shadows**.

CSS3 was started in 1998 but it has never been completed. Some parts are still being developed and some components work on some browsers. It published in June 1999. CSS 3 is divided into several separate documents called "modules". Each module adds new capabilities or extends features defined in CSS 2.

Function of CSS:

CSS is the language for describing the presentation of Web pages, including colors, layout, and fonts. It allows one to adapt the presentation to different types of devices, such as large screens, small screens, or printers.

CSS is independent of HTML and can be used with any XML-based markup language. The separation of HTML from CSS makes it easier to maintain sites, share style sheets across pages, and modify pages to different environments.

Types of CSS (Cascading Style Sheet)

Cascading Style Sheet (CSS) is used to set the style in web pages that contain HTML elements.

There are three types of CSS which are given below:

- ✓ Inline CSS
- ✓ Internal or Embedded CSS
- ✓ External CSS

Inline CSS: Inline CSS contains the CSS property in the body section attached with element is known as inline CSS.

```
<html>
   <head>
      <title>Inline</title>
   </head>
   <body><p style = "color:#009900; font-size:50px;
            font-style:italic; text-align:center;">

                        Welcome page
      </p>

   </body>
</html>
```

Internal or Embedded CSS: This can be used when a single HTML document must be styled uniquely. The CSS rule set should be within the HTML file in the head section,

```
<html>
   <head>
      <title>Internal CSS</title>
      <style>
         .main {
            text-align:center;
         }
         .info {
            color:#009900;
            font-size:50px;
            font-weight:bold;
         }
         .sec {
            font-style:bold;
            font-size:20px;
         }
      </style>
   </head>
   <body>
      <div class = "main">
         <div class ="info">Hi How are you</div>

         <div class ="sec">
            A computer science portal for students
         </div>
      </div>
   </body>
</html>
```

External CSS: External CSS contains separate CSS file which contains only style property with the help of tag attributes.

```
body {
   background-color:powderblue;
}
.main {
   text-align:center;
}
.info {
   color:#009900;
   font-size:50px;
   font-weight:bold;
}
#sec {
   font-style:bold;
   font-size:20px;
}
```

- ✓ **link** tag is used to link the external style sheet with the html webpage.
- ✓ **href** attribute is used to specify the location of the external style sheet file.

```
<html>
   <head>
      <link rel="stylesheet" href="file.css"/>
   </head>

   <body>
      <div class = "main">
         <div class ="info">WELCOME</div>
         <div id ="sec">
            A computer science portal for students
         </div>
      </div>
   </body>
</html>
```

Bootstrap:

Bootstrap is a front-end framework used to create modern websites and web apps. It's open-source and free to use, it helps to create UI elements such as buttons and forms. It also supports JavaScript extensions.

Basically, Software engineers use Bootstrap for a number of different reasons. It is easy to set up and master, it has a lot of components, a good grid system, styling for many HTML elements

It is great for creating layouts. its responsive to conform to different devices. It can be ensured consistency, eliminate cross-browser issues, and so on.

WEB APLICATION:

Most modern web applications are a collection of both **static** and **dynamic** web pages.

A static website contains simple HTML pages and supporting files hosted on a web server. When a site visitor requests a static page by clicking a link, the web server sends the page directly to the web browser without modifying the final contents.

A dynamic page displays different content for different users while retaining the same layout and design. Such pages, usually written in PHP, JAVA or ASP.NET, take more time to load than simple static pages.

Dynamic pages usually contain application programs for different services and require server-side resources like databases. A database allows the page creator to separate the website's design from the content to be displayed to users. Once they upload content into the database, it is retrieved by the website in response to a user request.

Difference between Static and Dynamic Web Pages:

Static Page	Dynamic Page
In static web pages, Pages will remain same until someone changes it manually.	In dynamic web pages, Content of pages are different for different visitors.
Static Web Pages are simple in terms with less complexity.	Dynamic web pages are complicated.
In static web pages, Information changes rarely.	In dynamic web page, Information changes frequently.
Static Web Page takes less time for loading than dynamic web page.	Dynamic web page takes more time for loading.
In Static Web Pages, database is not used.	In dynamic web pages, database is used.
Static web pages are written in languages such as: HTML, JavaScript, CSS, etc.	Dynamic web pages are written in languages such as: PHP, JAVA, ASP, ASP.NET, etc.
Static web pages require less work and cost in designing.	Dynamic web pages require comparatively more work and cost in designing them.

A **web server** stores and delivers the content for a website such as text, images, video, and application data. The most common type of client is a web browser program, which requests data from your website when a user clicks on a link or downloads a document on a page displayed in the browser.

A web server communicates with a web browser using the Hypertext Transfer Protocol (HTTP). The content of most web pages is encoded in Hypertext Markup Language (HTML). To deliver dynamic content, most web servers support server-side scripting languages to encode business logic into the communication. Commonly supported languages include Active Server Pages (ASP.Net), PHP, Python, and Java and so on.

Web Site VS Web Server: Web site and Web server are different: A Web site consists of a collection of Web pages associated with a particular hostname. A Web server is a program to satisfy client requests for Web resources.

Types of Web Servers:

1. Apache Web Server
2. IIS Server
3. Xampp Server
4. WAMP Server

Apache Web Server:

Introduction: Apache Web server is the most commonly used http server today. About 80% of all websites and Intranets use Apache web server to deliver their content to requesting Browsers.

Server-side programming languages such as PHP, asp.net, Python, Java and many others

Apache, an open-source Web server created by American software developer Robert McCool. Apache was released in 1995 and quickly gained a majority hold on the Web server market. Apache provides servers for Internet giants such as Google and Wikimedia projects such as Wikipedia.

In the early 21st century, Apache servers deployed more than 50 percent of the Internet's content.

As a Web server, Apache is responsible for accepting directory (HTTP) requests from Internet users and sending them their desired information in the form of files and Web pages. Much of the Web's software and code is designed to work along with Apache's features. Programmers working on Web applications typically make use of a home version of Apache to preview and test code. Apache also has a safe and secure file-sharing feature, allowing users to put files into the root directory of their Apache software and share them with other users. The Apache server's impact on the open-source software community is partly explained by the unique license through which software from the Apache Software Foundation is distributed.

Apache was originally known as the NCSA HTTPd Web server and was written by McCool when he was an undergraduate at the National Center for Supercomputing Applications (NCSA) at the University of Illinois at Urbana-Champaign. Apache is maintained and developed by a large community of volunteers and developers from the Apache Software Foundation, as well as by contributions from users worldwide. **[UNIT 1-3 ONLY]**

**********END**********

JavaScript

JavaScript is a lightweight, cross-platform, and interpreted scripting language. It is well-known for the development of web pages; many non-browser environments also use it. JavaScript can be used for **Client-side** developments as well as **Server-side** developments. JavaScript contains a standard library of objects, like **Array**, **Date**, and **Math**, and a core set of language elements like **operators**, **control structures**, and **statements.**

- ✓ **Client-side:** It supplies objects to control a browser and its Document Object Model (DOM). Like if client-side extensions allow an application to place elements on an HTML form and respond to user events such as **mouse clicks**, **form input**, and **page navigation**. Useful libraries for the client-side are **AngularJS**, **ReactJS**, **VueJS** and so many others.
- ✓ **Server-side:** It supplies objects relevant to running JavaScript on a server. Like if the server-side extensions allow an application to communicate with a database, and provide continuity of information from one invocation to another of the application, or perform file manipulations on a server. The useful framework which is the most famous these days is **node.js**.

- ✓ JavaScript can be added to your HTML file in **two ways:**
- ✓ **Internal JS:** We can add JavaScript directly to our HTML file by writing the code inside the <script> tag. The <script> tag can either be placed inside the <head> or the <body> tag according to the requirement.
- ✓ **External JS:** We can write JavaScript code in other file having an extension .js and then link this file inside the <head> tag of the HTML file in which we want to add this code.

History of JavaScript:

It was created in 1995 by Brendan Eich while he was an engineer at Netscape. It was originally going to be named LiveScript but was renamed. Unlike most programming languages, the JavaScript language has no concept of input or output. It is designed to run as a scripting language in a host environment, and it is up to the host environment to provide mechanisms for communicating with the outside world. The most common host environment is the browser.

Features of JavaScript: According to a recent survey conducted by **Stack Overflow**, JavaScript is the most popular language on earth.
With advances in browser technology and JavaScript having moved into the server with Node.js and other frameworks, JavaScript is capable of so much more.

Here are a few things that we can do with JavaScript:

- ✓ JavaScript was created in the first place for DOM manipulation. Earlier websites were mostly static, after JS was created dynamic Web sites were made.
- ✓ Functions in JS are objects. They may have properties and methods just like another object. They can be passed as arguments in other functions.
- ✓ Can handle date and time.
- ✓ Performs Form Validation although the forms are created using HTML.
- ✓ No compiler needed.
- ✓

Applications of JavaScript:

- ✓ **Web Development:** Adding interactivity and behavior to static sites JavaScript was invented to do this in 1995. By using AngularJS that can be achieved so easily.
- ✓ **Web Applications:** With technology, browsers have improved to the extent that a language was required to create robust web applications. When we explore a map in Google Maps then we only need to click and drag the mouse. All detailed view is just a click away, and this is possible only because of JavaScript. It uses Application Programming Interfaces (APIs) that provide extra power to the code. The Electron and React is helpful in this department.
- ✓ **Server Applications:** With the help of Node.js, JavaScript made its way from client to server and node.js is the most powerful in the server-side.
- ✓ **Games:** Not only in websites, but JavaScript also helps in creating games for leisure. The combination of JavaScript and HTML 5 makes JavaScript popular in game development as well. It provides the EaseJS library which provides solutions for working with rich graphics.
- ✓ **Smartwatches:** JavaScript is being used in all possible devices and applications. It provides a library PebbleJS which is used in smartwatch applications. This framework works for applications that require the internet for its functioning.
- ✓ **Art:** Artists and designers can create whatever they want using JavaScript to draw on HTML 5 canvas, make the sound more effective also can be used p5.js library.

- ✓ **Machine Learning:** This JavaScript ml5.js library can be used in web development by using machine learning.

Where to put JavaScript in an HTML Document?

JavaScript can be placed inside the body or the head section of an HTML page or inside both head and body.

JavaScript in head

```
<html>
   <head>
      <script>
         function gfg() {
           document.getElementById("demo").innerHTML = "Geeks For Geeks";
         }
      </script>
   </head>
   <body>
      <h2>JavaScript in Head</h2>
      <p id="demo" style="color:green;">geeksforgeeks.</p>
      <button type="button" onclick="gfg()">click it</button>
   </body>
</html>
```

JavaScript in body

```
<html>
   <center>
      <body>
         <h2>JavaScript in Body</h2>
         <p id="demo">geeksforgeeks.</p>
         <button type="button" onclick="gfg()">Try it</button>
         <script>
            function gfg() {
              document.getElementById("demo").innerHTML = "Geeks For Geeks";
            }
         </script>
      </body>
   </center>
</html>
```

External JavaScript

```
<html>
   <center>
   <body>
      <h2>External JavaScript</h2>
      <p id="demo">Geeks For Geeks.</p>
      <button type="button" onclick="myFunction()">Try it</button>
      <script src="myScript.js"></script>
   </body>
   <center>
</html>
```

JavaScript | Statements

The programming instructions written in a program in a programming language are known as statements. Order of execution of Statements is the same as they are written.

Ex:1

```
<html>

<body>
    <h2>Welcome</h2>

    <p id="dev"></p>

    <script>

        var a, b, c;


        a = 2;


        b = 3;


        c = a + b;

 document.getElementById("dev").innerHTML = "The value of c is " + c + ".";

    </script>

</body>

</html>
```

Ex:2

```
<html>
<body>

<h2>JavaScript Variables</h2>

<p>In this example, x, y, and z are variables.</p>

<p id="demo"></p>

<script>
var x = 5;
var y = 6;
var z = x + y;
document.getElementById("demo").innerHTML =
"The value of z is: " + z;
</script>
</body>
</html>
```

Code Blocks:

JavaScript statements can be grouped together inside curly brackets. Such groups are known as **code.**

Ex:1

```
<html>
<body>
    <p>Welcome</p>
    <button type="button"
            onclick="myFunction()">
      Click Me!
  </button>

    <p id="dev1"></p>
    <p id="dev2"></p>

    <script>
        function myFunction() {

document.getElementById("dev1").innerHTML = "Hello";

document.getElementById("dev2").innerHTML = "How are you?";
        }
    </script>

</body>

</html>
```

JavaScript Variables: A JavaScript variable is the simple name of storage location where data to be stored. There are two types of variables in JavaScript which are listed below:

- ✓ **Local variables:** Declare a variable inside of block or function.
- ✓ **Global variables:** Declare a variable outside function or with window object.

```
<script>

// Gloabal variable declaration
var Name="Apple";

function MyFunction() {

    // Local variable declaration
    var num = 45;

    // Display the value of Gloabal variable
    document.writeln(Name);

    // Display the value of local variable
    document.writeln("<br>" + num );
}

MyFunction();

</script>
```

JavaScript Operator:

JavaScript operators are symbols that are used to compute the value or in other word we can perform operations on operands. Arithmetic operators (+, -, *, /) are used to compute the value and Assignment operator (=, +=, %=) are used to assign the values to variables.

Ex:1

```
<script>

var x, y, sum;
x = 3;
y = 23;

sum = x + y;

document.write(sum);

</script>
```

JavaScript | Output

JavaScript Output defines the ways to display the output of a given code. The output can be display by using four different ways which are listed below:

- ✓ **innerHTML:** It is used to access an element. It defines the HTML content

JS: document.getElementById(id)

Ex:1

```
<html>

<head>
    <title>
        JavaScript Output using innerHTML
    </title>
</head>

<body>
    <h1>Welcome to Javascript </h1>

    <h2>
        JavaScript Display Possibilities
        Using innerHTML
    </h2>

    <p id="DN"></p>

    <!-- Script to uses innerHTML -->
    <script>
        document.getElementById("DN").innerHTML = 10 * 2;
    </script>
</body>

</html>
```

EX:2

```
<html>

<head>
    <title>
        JavaScript Output using window.alert()
    </title>
</head>

<body>
    <h1>Welcome window.alert</h1>

    <h2>
        JavaScript Display Using window.alert()
    </h2>

    <p id="DN"></p>

    <script>
        window.alert(10 * 2);
    </script>
</body>

</html>
```

BASIC FORM OPERATORS IN JAVASCRIPT FUNCTION

USING VARIABLES AND OPERATOR

EX:1

```
<html>

<head>
    <title>
        JavaScript Comments
    </title>

    <script>


        function add() {


            var x, y, z;


            x = Number(document.getElementById("num1").value );

            y = Number(document.getElementById("num2").value );


            z= x +y;


            document.getElementById("sum").value = z;
        }
    </script>
</head>

<body>
    First number: <input id="num1"><br><br>
    Second number: <input id="num2"><br><br>

    <button onclick="add()">Sum</button>

    <input id="sum">
</body>

</html>
```

JavaScript | Events

JavaScript has events to provide a dynamic interface to a webpage. These events are hooked to elements in the Document Object Model (DOM).

There are some JavaScript events:

1) onclick events: This is a mouse event and provokes any logic defined if the user clicks on the element, it is

EX:1

```
<html>
  <head>
    <script>
      function hiThere() {
        alert('Hi there!');
      }
    </script>
  </head>
  <body>
    <button type="button" onclick="hiThere()">Click me event</button>
  </body>
</html>
```

2) onkeyup event: This event is a keyboard event and executes instructions whenever a key is released after pressing.

EX:2

```
<html>
  <head>
    <script>
      var a = 0;
      var b = 0;
      var c = 0;
      function changeBackground() {
        var x = document.getElementById('bg');
        bg.style.backgroundColor = 'rgb('+a+', '+b+', '+c+')';
        a += 1;
        b += a + 1;
        c += b + 1;
        if (a > 255) a = a - b;
        if (b > 255) b = a;
        if (c > 255) c = b;
      }
    </script>
  </head>
  <body>
    <input id="bg" onkeyup="changeBackground()"
      placeholder="write something" style="color:#fff">
  </body>
</html>
```

3) onmouseover event: This event corresponds to hovering the mouse pointer over the element and its children, to which it is bound to.

EX:3

```
<html>
  <head>
    <script>
       function hov() {
         var e = document.getElementById('hover');
         e.style.display = 'none';
      }
    </script>
  </head>
  <body>
    <div id="hover" onmouseover="hov()"
      style="background-color:green;height:200px;width:200px;">
    </div>
  </body>
</html>
```

4) onmouseout event: Whenever the mouse cursor leaves the element which handles a mouseout event, a function associated with it is executed.

```
<html>
  <head>
    <script>
      function out() {
        var e = document.getElementById('hover');
        e.style.display = 'none';
      }
    </script>
  </head>
  <body>
    <div id="hover" onmouseout="out()"
      style="background-color:green;height:200px;width:200px;">
    </div>
  </body>
</html>
```

5) onchange event: This event detects the change in value of any element listing to this event.

EX:1

```
<html>
  <head></head>
  <body>
    <input onchange="alert(this.value)" type="number">
  </body>
</html>
```

6) onload event: When an element is loaded completely, this event is evoked.

```
<html>

  <head></head>

  <body>

    <img onload="alert('Image completely loaded')"

      alt="Logo"

      src="AddImage">

  </body>

</html>
```

7) onfocus event: An element listing to this event executes instructions whenever it receives focus.

```
<html>
  <head>
    <script>
      function focused() {
        var e = document.getElementById('inp');
        if (confirm('Got it?')) {
          e.blur();
        }
      }
    </script>
  </head>
  <body>
    <p >Take the focus into the input box below:</p>

    <input id="inp" onfocus="focused()">
  </body>
</html>
```

8) onblur event: This event is evoked when an element loses focus.

```
<html>
  <head></head>
  <body>

<p>Write something in the input box and then click elsewhere
       in the document body.</p>

    <input onblur="alert(this.value)">
  </body>
</html>
```

JavaScript Data Types

Data types basically specify what kind of data can be stored and manipulated within a program.

There are **six basic** data types in JavaScript which can be divided into three main categories:

primitive (primary**), composite** (reference) and **special** data types.

String, Number and Boolean are **primitive** data types.

Object, Array, and Function are **composite** data types.

Whereas Undefined and Null are **special** data types.

Primitive data types can hold only one value at a time, whereas composite data types can hold collections of values and more complex entities.

The null type is the second primitive data type that also has only one value: **null**. JavaScript defines that null is equal to undefined.

JavaScript was originally developed as LiveScript by Netscape in the mid-1990s. It was later renamed to JavaScript in 1995 and became an ECMA standard in 1997. Now JavaScript is the standard client-side scripting language for web-based applications, and it is supported by virtually all web browsers available today, such as Google Chrome, Mozilla Firefox.

JavaScript is an object-oriented language, and it also has some similarities in syntax to Java programming language. But JavaScript is not related to Java in any way.

JavaScript is officially maintained by ECMA (European Computer Manufacturers Association).

(ES6) is the latest major version of the ECMAScript standard.

Except VAR One of the features that came with ES6 is the addition of LET and CONST, which can be used for variable declaration as well.

Data Types

In programming, data types are an important concept. To be able to operate on variables, it is important to know something about the types:

STRING DATA:

```
<body>

<h2>JavaScript Strings</h2>

<p>Strings are written </p>

<p id="demo"></p>

<script>

let carName1 = "Volvo XC60";

let carName2 = 'Volvo XC60';

document.getElementById("demo").innerHTML =

carName1 + "<br>" +

carName2;

</script>

</body>

</html>
```

NUMBER DATA TYPE

Ex:2

```
<html>
<body>
<h2>JavaScript Numbers</h2>
<p>Numbers can be written </p>
<p id="demo"></p>
<script>
let x1 = 34.00;
let x2 = 34;
let x3 = 3.14;
document.getElementById("demo").innerHTML =
x1 + "<br>" + x2 + "<br>" + x3;
</script>
</body>
</html>
```

Booleans Data Type

Ex:3

```
<html>
<body>
<h2>JavaScript Booleans</h2>
<p>Booleans can have two values: true or false:</p>
<p id="demo"></p>
<script>
let x = 5;
let y = 5;
let z = 6;
document.getElementById("demo").innerHTML =
(x == y) + "<br>" + (x == z);
</script>
</body>
</html>
```

Array Data Type:

Ex: 4

```
<html>
<body>

<h2>JavaScript Arrays</h2>
<p>Array indexes are zero-based, which means the first item is [0].</p>
<p id="demo"></p>
<script>
const cars = ["Saab","Volvo","BMW"];
document.getElementById("demo").innerHTML = cars[0];
</script>
</body>
</html>
```

Object Data Type:

Ex:5

```
<html>
<body>
<h2>JavaScript Objects</h2>
<p id="demo"></p>
<script>
const person = {
  firstName : "Durganand",
  lastName  : "Panjiyar",
  age     : 39,
  eyeColor  : "Royal Blue"
};
document.getElementById("demo").innerHTML =
person.firstName + " is " + person.age + " years old.";
</script>
</body>
</html>
```

Undefined

```
<html>

<body>


<h2>JavaScript</h2>


<p>The value of a variable with no value is <b>undefined</b>.</p>


<p id="demo"></p>


<script>

let car;

document.getElementById("demo").innerHTML =

car + "<br>" + typeof car;

</script>


</body>

</html>
```

**********END**********

INTRODUCTION TO PHP

INTRODUCTION TO PHP

PHP was originally created by **Rasmus Lerdorf** in 1994. PHP is one of the most widely used server-side scripting language for web development. Popular websites like Facebook, Yahoo are developed using PHP. PHP stands for **Hypertext Pre-Processor**. it is a scripting language used to develop static and dynamic webpages and web applications. Here are a few important things you must know about PHP:

1. PHP is an Interpreted language; it doesn't need a compiler.
2. To run and execute PHP code, we need a Web server.
3. PHP is a server-side scripting language, which means that PHP is executed on the server and the result is sent to the browser.
4. PHP is open source and free.

PHP VARIAVLE

Variables are used to store data, like string, Text and numbers. Variable values can change over the course of a script.

Important things to know about variables:

1. All variables in PHP are signified with dollar sign ($).
2. The value of a variable is the value of its assignment.
3. Variables are assigned with the = operator, with the variable on the left-hand side and the expression on the right.
4. Variables used before they are assigned have default values.
5. PHP does a good job of automatically converting types from one to another when necessary

Ex:1

```
<?php
$txt = "Hello World!";
$number = 10;
echo $txt;
echo $number;
?>
```

PHP OPERATORS:

Operators are used to perform operations on some values. In other words, we can describe operators as something that takes some values, performs some operation on them and gives a result. From example: 1 + 2 = 3 this expression is an operator.

Just like any other programming language, PHP also supports various types of operations like the arithmetic operations (addition, subtraction, others), logical operations (AND, OR), Increment/Decrement Operations. PHP provides us with many operators to perform such operations on various operands and values.

Below are the groups of operators:

- Arithmetic Operators
- Logical Operators
- Comparison Operators
- Conditional Operators
- Assignment Operators
- Array Operators
- Increment/Decrement Operators
- String Operators
- Spaceship Operators (Introduced in PHP 7)

Arithmetic Operators

Operator	Description	Example
+	Addition	$x + $y
-	Subtraction	$x - $y
*	Multiplication	$x * $y
/	Division	$x / $y
%	Modulus	$x % $y

Ex:1

```
<?php
error_reporting();
$x = $_POST['fst'];
$y = $_POST['snd'];
$a = $x + $y;
$b = $x - $y;
$c = $x * $y;
$d = $x / $y;
$e = $x % $y;
?>
<html>
<head>
<title>Operators</title>
<style>
input[type=text] {
  width: 100%;
  padding: 12px 20px;
  margin: 8px 0;
```

```
  display: inline-block;
  border: 1px solid #ccc;
  border-radius: 4px;
  box-sizing: border-box;
}
input[type=submit] {
  width: 20%;
  background-color: #4CAF50;
  color: white;
  padding: 14px 20px;
  margin: 8px 0;
  border: none;
  border-radius: 4px;
  cursor: pointer;
  float:right;
}
</style>
</head>
<body>
<table style="width:40%; margin:0 auto; background-color:brown; margin-top:40px;
color:#fff;">
<form action="" method="POST">

<tr>
<td><h3>Enter No1</h3></td>
<td><input type="text" name="fst" /></td>
</tr>
<tr>
<td><h3>Enter No2</h3></td>
<td><input type="text" name="snd" /></td>
</tr>
```

```
<tr>
<td><h1>Result</h1></td>

<td>
<?php echo "Add:$a <br />"; ?>
<?php echo "Sub:$b <br />"; ?>
<?php echo "Mul:$c <br />"; ?>
<?php echo "Div:$d <br />"; ?>
<?php echo "Moduler:$e <br />"; ?>
</td>
</tr>
<tr>
<td colspan="2">
<input type="submit" value="RESULT">
</td>
</tr>
</form>
</table>
</body>
</html>
```

PHP Assignment Operators

The assignment operators are used to assign values to variables.

Operator	Description	Example
=	Assign	$x = $y
+=	Add and assign	$x += $y
-=	Subtract and assign	$x -= $y
*=	Multiply and assign	$x *= $y
/=	Divide and assign quotient	$x /= $y
%=	Divide and assign modulus	$x %= $y

EX:1

```
<html>

  <head>
    <title>Assignment Operators</title>
  </head>

  <body>

    <?php

       $a = 20;
       $b = 10;
        echo "<br />";
                  echo "Value A : 20 <br />";
                  echo "Value B : 10 <br />";
       $c = $a + $b;
       echo "total: $c <br/><br/>";

       $c += $a;
       echo "Total + a: $c <br/>";

       $c -= $a;
       echo "Total -a : $c <br/>";

       $c *= $a;
       echo "Multiple a*: $c <br/>";
```

```
    $c /= $a;
    echo "Division a/: $c <br/>";

    $c %= $a;
    echo "Modulus a%: $c <br/>";
  ?>

 </body>
</html>
```

PHP Comparison Operators

The comparison operators are used to compare two values in a Boolean fashion.

Operator	Name	Example
==	Equal	$x == $y
===	Identical	$x === $y
!=	Not equal	$x != $y
<>	Not equal	$x <> $y
!==	Not identical	$x !== $y
<	Less than	$x < $y
>	Greater than	$x > $y
>=	Greater than or equal to	$x >= $y
<=	Less than or equal to	$x <= $y

EX:1

```
<html>
  <head>
    <title>Comparison Operators</title>
  </head>

  <body>

    <?php

       $a = 42;
       $b = 20;

           echo "A value:$a <br />";
           echo "B value:$b <br />";

       if( $a == $b ) {
         echo " a==b: is equal to b<br/>";
       }else {
         echo " a==b: is not equal to b<br/>";
       }

       if( $a > $b ) {
         echo " a>b: is greater than  b<br/>";
       }else {
         echo "a>b: is not greater than b<br/>";
       }

       if( $a < $b ) {
         echo "a<b: is less than  b<br/>";
       }else {
         echo "a<b: a is not less than b<br/>";
       }

       if( $a != $b ) {
         echo " a!=b : a is not equal to b<br/>";
       }else {
         echo " a!=b : a is equal to b<br/>";
       }

       if( $a >= $b ) {
         echo " a>=b: a is either greater than or equal to b<br/>";
       }else {
         echo "a>=b : a is neither greater than nor equal to b<br/>";
```

```
      }

      if( $a <= $b ) {
        echo "a<=b : a is either less than or equal to b<br/>";
      }else {
        echo "a<=b : a is neither less than nor equal to b<br/>";
      }
    ?>

  </body>
</html>
```

PHP Incrementing and Decrementing Operators

There are two types of incrementing and decrementing operators, post incrementing/decrementing operators and pre incrementing /decrementing operators. Post incrementing operators are placed after a variable name, and pre incrementing /decrementing operators are placed before the name.

Operator	Name
++$x	Pre-increment
$x++	Post-increment
--$x	Pre-decrement
$x--	Post-decrement

Incrementing and Decrementing Operators

Examples:

Post Incrementing

Ex:1

```
<?php
$a = 8;
$b = $a++;
echo "$b <br />";
echo $a;
?>
```

Post Decrementing

Ex:2

```
<?php
$a = 10;
$b = $a--;
echo "$b <br />";
echo $a;
?>
```

Pre-Incrementing

Ex:3

```
<?php
$a = 8;
$b = ++$a;
echo "$b <br />";
echo $a;
?>
```

Pre-Decrementing
Ex:4

```
<?php
$a = 10;
$b = --$a;
echo "$b <br />";
echo $a;
?>
```

PHP Logical Operators

The logical operators are typically used to combine conditional statements.

Operator	Name	Example
and	And	$x and $y
or	Or	$x or $y
xor	Xor	$x xor $y
&&	And	$x && $y
||	Or	$x || $y
!	Not	!$x

AND

Ex:1

```
<html>
<body>
<?php
$x = 100;
$y = 50;

if ($x == 100 and $y == 50) {
   echo "Both and different No:";
}
?>

</body>
</html>
```

NOR

Ex: 1

```
<html>
<body>

<?php
$x = 100;
$y = 50;

if ($x == 100 xor $y == 80) {
   echo "Hello world!";
}
?>

</body>
</html>
```

NOT

Ex :1

```
<html>
<body>

<?php
$x = 100;

if ($x !== 100) {
   echo "Hello world!";
}
?>
```

PHP String Operators

These operators are tokens of a programming language that allow some operation on variables. Strings in any programming language can be combined to a variable. PHP String operators are (.) dot operator and (.=) concatenation assignment operator .

Operator	Description	Example
.	Concatenation	$str1. $str2
.=	Concatenation assignment	$str1. = $str2

Dot Operator (.)

Ex:1

```
<?php
$my_hello='Hello ';
$my_world='World ';
$my_welcome='Welcome ';
$my_php= 'Come and learn ';
$my_site= "String Operator ";

echo $my_hello.$my_world.$my_welcome.$my_php.$my_site.'<br>';
?>
```

Concatenation Assignment Operator (.=)

Concatenation Assignment operator behaves like an assignment operator that combines the string on the right-hand side with the variable value on the left-hand side and updates the variable's value with concatenated string

Ex: 1

```
<?php
$my_msg="";
$hello='Hello ';
$world='World ';

$welcome='Welcome ';

$php= 'PHP programming ';
```

```
$site= "Application ";

$my_msg.=$hello;
echo $my_msg.'<br>';
$my_msg.=$world;
echo $my_msg.'<br>';
$my_msg.=$welcome;
echo $my_msg.'<br>';
$my_msg.=$site.'<br>';
echo $my_msg.'<br>';
?>
```

PHP Array Operators

Arrays is a type of data structure that allows us to store multiple elements of similar data type under a single variable.

. The arrays are helpful to create a list of elements of similar types, which can be accessed using their index or key.

Suppose we want to store five names and print them accordingly. This can be easily done by the use of five different string variables. But if instead of five, the number rises to a hundred, then it would be really difficult for the developer to create so many different variables.

Here array comes into play and helps us to store every element within a single variable and also allows easy access using an index or a key. An array is created using an **array()** function in PHP.

There are basically three types of arrays in PHP:

- **Indexed or Numeric Arrays:** An array with a numeric index where values are stored linearly.
- **Associative Arrays:** An array with a string index where instead of linear storage, each value can be assigned a specific key.
- **Multidimensional Arrays:** An array which contains single or multiple array within it and can be accessed via multiple indices.

The array operators are used to compare arrays:

Operator	Name	Example
+	Union	$x + $y
==	Equality	$x == $y
===	Identity	$x === $y
!=	Inequality	$x != $y
<>	Inequality	$x <> $y
!==	Non-identity	$x !== $y

Ex:1

```
<?php
 $arr_variable = array(
 1 => "PHP",
 2 => "MySQL",
 3 => "Java",
$pro = "Learning",
 );
 foreach( $arr_variable as $arritem )
 {
 echo "$arritem <br />";
 }
 ?>
```

Numeric Array

These arrays can store numbers, strings and any object but their index will be represented by numbers

Ex:1

```
<html>
  <body>
  <?php
     $num = array( 1, 2, 3, 4, 5);
       foreach( $num as $value ) {
         echo "$value <br />";
       }
       $num[0] = "one";
       $num[1] = "two";
       $num[2] = "three";
       $num[3] = "four";
      $num[4] = "five";
        foreach( $num as $value ) {
         echo "Value carry: $value <br />";
       }
    ?>
 </body>
</html>
```

Associative Arrays

The associative arrays are very similar to numeric arrays in term of functionality but they are different in terms of their index. Associative array will have their index as string so that you can establish a strong association between key and values.

Ex:1

```
<html>
<body>
<?php
      /* First method to associate create array. */
      $salaries = array("admin" => 2000, "acc" => 1000, "teach" => 500);
        echo "Salary  ". $salaries['admin'] . "<br />";
      echo "Salary  ".  $salaries['acc']. "<br />";
      echo "Salary ".  $salaries['teach']. "<br />";
       /* Second method to create array. */
      $salaries['admin'] = "high";
      $salaries['acc'] = "medium";
      $salaries['teach'] = "low";
      echo "Salary Admin ". $salaries['admin'] . "<br />";
      echo "Salary Acc ".  $salaries['acc']. "<br />";
      echo "Salary Teach ".  $salaries['teach']. "<br />";
   ?>
 </body>
</html>
```

Multidimensional Arrays

A multi-dimensional array each element in the main array can also be an array. And each element in the sub-array can be an array, and so on. Values in the multi-dimensional array are accessed using multiple index.

Ex:1

```
<html>
  <body>

    <?php
      $marks = array(
        "student1" => array (
          "physics" => 35,
          "maths" => 30,
          "chemistry" => 39
        ),

        "student2" => array (
          "physics" => 30,
          "maths" => 32,
          "chemistry" => 29
        ),

        "student3" => array (
          "physics" => 31,
          "maths" => 22,
```

```
            "chemistry" => 39
         )
      );

      /* Accessing multi-dimensional array values */
      echo "Marks for student1 in physics : " ;
      echo $marks['student1']['physics'] . "<br />";

      echo "Marks for student2 in maths : ";
      echo $marks['student2']['maths'] . "<br />";

      echo "Marks for student3 in chemistry : " ;
      echo $marks['student3']['chemistry'] . "<br />";
   ?>

  </body>
</html>
```

PHP PROGRAMMING (INSERT DATA INTO DATABASE)

Create Table

```
CREATE TABLE tbldemo (
id INT(6) AUTO_INCREMENT PRIMARY KEY,
firstname VARCHAR(30) NOT NULL,
lastname VARCHAR(30) NOT NULL,
email VARCHAR(50),
reg_date TIMESTAMP DEFAULT CURRENT_TIMESTAMP ON UPDATE
CURRENT_TIMESTAMP
)
```

PHP BACKEND PROGRAMMING

```
<?php
$servername = "localhost";
$username = "root";
$password = "";
$dbname = "demo";
$conn = mysqli_connect($servername, $username, $password, $dbname);
if (!$conn) {
  echo "Failed Connection";
}
$sql = "INSERT INTO tbdemo(firstname, lastname, email)
VALUES ('Dev', 'Panjiyar', 'dn@gmail.com.com')";
$res = mysqli_query($conn,$sql);
if(!$res)
{        echo "failed to send data";}
else
{echo "Success";}
?>
```

FRONTEND IN PHP

```
<html>
        <head>
<title>wt</title>
</head>
        <body>
        <h1>INSERT DATA INTO</h1>
        <form action="back.php" method="POST">
        <table><tr>
        <td>Enter Firstname</td>
        <td><input type="text" name="f"></td>
        </tr><tr>
        <td>Enter Lastname</td>
        <td><input type="text" name="l"></td>
        </tr>
        <tr>
        <td>Enter E-mail</td>
        <td><input type="text" name="e"></td>
        </tr>
        <tr>
        <td></td>
        <td><input type="submit" value="INSERT"></td>
        </tr>
        </td>
        </tr>
        </table>
</form>
</body>
</html>
```

PHP DISPLAY RECORDS FROM DATABASE

```
<?php
$servername = "localhost";
$username = "root";
$password = "";
$dbname = "demo";

$conn = mysqli_connect($servername, $username, $password, $dbname);

if (!$conn) {
  echo "no connected!";
}

$sql = "SELECT id, firstname, lastname FROM tbdemo";
$result = mysqli_query($conn, $sql);

if (mysqli_num_rows($result) > 0) {

  while($row = mysqli_fetch_array($result)) {
    echo "id:" . $row["id"]. "Name:" . $row["firstname"]. " " . $row["lastname"]. "<br>";
  }
}
 else
{
  echo "0 results";
}

mysqli_close($conn);
?>
```

Role Based Access Control

Role-Based Access Control (RBAC) is a security model whereby users are granted access to resources based on their role in the company

.

The basic principle of Role-Based Access Control is simple: The Finance department can't see HR data.

When implemented correctly, RBAC will be transparent to the users. Role assignment happens behind the scenes, and each user has access to the applications and data that they need to do their job.

RBAC Example: The Basics

1. **Role assignment:** A subject can exercise a permission only if the subject has selected or been assigned a role.
2. **Role authorization:** A subject's active role must be authorized. That is, I can't just assign myself to a role. I need authorization.
3. **Permission authorization:** A subject can exercise a permission only if the permission is authorized for the subject's active role. With rules 1 and 2, this rule ensures that users can only exercise permissions for which they are authorized.

Benefits Of RBAC

1. RBAC helps maximize operational efficiency, protects your data from being leaked or stolen, reduces admin and IT support work, and makes it easier to meet audit requirements.
2. RBAC also reduces IT and administrative load across the organization and increases the productivity of the users. IT doesn't have to manage personalized permissions for every user, and it's easier for the right users to get to the right data.
3. Managing new users and guest users without time-consuming and difficult

4. companies can implement RBAC systems to meet the regulatory and statutory requirements for confidentiality and privacy because executives and IT departments can more effectively manage how the data is accessed and used

XML

XML stands for Extensible Markup Language. It is a text-based markup language derived from Standard Generalized Markup Language (SGML). XML tags identify the data and are used to store and organize the data, rather than specifying how to display it like HTML tags, which are used to display the data. XML is not going to replace HTML in the near future, but it introduces new possibilities by adopting many successful features of HTML.

There are three important characteristics of XML that make it useful in a variety of systems and solutions:

- XML is extensible: XML allows you to create your own self-descriptive tags or language, that suits your application.
- XML carries the data, does not present it: XML allows you to store the data irrespective of how it will be presented.
- XML is a public standard: XML was developed by an organization called the World Wide Web Consortium (W3C) and is available as an open standard.

XML Usage

- XML can work behind the scene to simplify the creation of HTML documents for large web sites.
- XML can be used to exchange the information between organizations and systems.
- XML can be used for offloading and reloading of databases.
- XML can be used to store and arrange the data, which can customize your data handling needs.
- XML can easily be merged with style sheets to create almost any desired output.
- Virtually, any type of data can be expressed as an XML document.

An XML document is a basic unit of XML information composed of elements and other markup in an orderly package. An XML document can contain a wide variety of data. For

example, database of numbers, numbers representing molecular structure or a mathematical equation.

XML Document Example

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<note>
  <to>To Dev</to>
  <from>You are </from>
  <heading>Remindered</heading>
  <body>Don't forget me this weekend! </body>
</note>
```

Static websites

A static website consists of a series of HTML files, each one representing a physical page of a website. So, on static sites, each page is a separate HTML file. When you visit the homepage, you are viewing the actual homepage file.

Even if two pages contain a chunk of identical content, they both contain two versions. So, if you want to update the footer, you must do so twice, once on each page.

Dynamic Websites

A dynamic website uses server technologies (such as PHP and MYSQL) to dynamically build a webpage right when a user visits the page.

Basically, what happens is the user goes to a certain web address and the server finds a bunch of different pieces of information that it writes into a single consistent web page

The benefits of dynamic websites

Easier design updates

Since each part of a web page is separate, it is much simpler to update something across many pages, all at once.

More flexible data

Because a dynamic site pulls together a bunch of page bits to make a whole page, you can store your content in a database.

Easier content updates

Dynamic websites with a content management system (CMS) make it simple for a non-technical person to create and update the content of the site.

Client/Server Architecture Mean

Client server architecture is a computing model in which multiple components work in strictly defined roles to communicate. The server hosts, delivers and manages most of the resources and services to be consumed by the client. This type of shared resources architecture has one or more client computers connected to a central server over a network or internet connection.

Client/server architecture is also known as a networking computing model or client/server network because all the requests and services are delivered over a network. It's considered a form of distributed computing system because the components are doing their work independently of one another.

Web Application

A web application is an application that is capable of working directly from a suitable web browser. The web application can run on several platforms at the same time and perform the entire task assigned by the client in a systematic format.

Parameter of Comparison	**Client Server Application**	**Web Application**
Architecture	It is made up of two tire architectures	It is made up of multi tire architecture
Operation	It requires pre-installation in the device	It can run directly on suitable web browse
Efficiency	May get overloaded by tasks decreasing the performance efficiency	Capable of working over multiple platforms in the same time with quick performance
Cookies	Cookies are not required	Cookies are needed
security	Less risk as there are fewer users	High risk comparatively as more number of users

Markup language is a series of markings that tells web servers the style and structure of a document.

HTML is not considered a programming language because it cannot create dynamic functionality.

some of the common uses for HTML:

- **Web development.** Developers use HTML code to design how a browser displays web page elements, such as text, hyperlinks, and media files.
- **Internet navigation.** Users can easily navigate and insert links between related pages and websites as HTML is heavily used to embed hyperlinks.
- **Web documentation.** HTML makes it possible to organize and format documents, similarly to Microsoft Word.

CSS

CSS is used for defining the styles for web pages. It is easier to make the web pages presentable using it.

We recommend to use CSS because the HTML attributes are compatible with future browsers, it is good to apply CSS in HTML pages.

There are several uses of CSS

Solves a big problem

CSS, tags like font, color, background style, element alignments, border and size had to be repeated on every web page. This was a very long process in HTML, But CSS has made it easier now.

Saves a lot of time

CSS style definitions are saved in external CSS files, so it is possible to change the entire website by changing just one file.

Provide more attributes

CSS provides more detailed attributes than plain HTML to define the look and feel of the website.

Pages load faster

CSS does not require the writing of HTML tag attributes every time. So using CSS, there is less code, which means faster downloading.

Easier Website maintenance

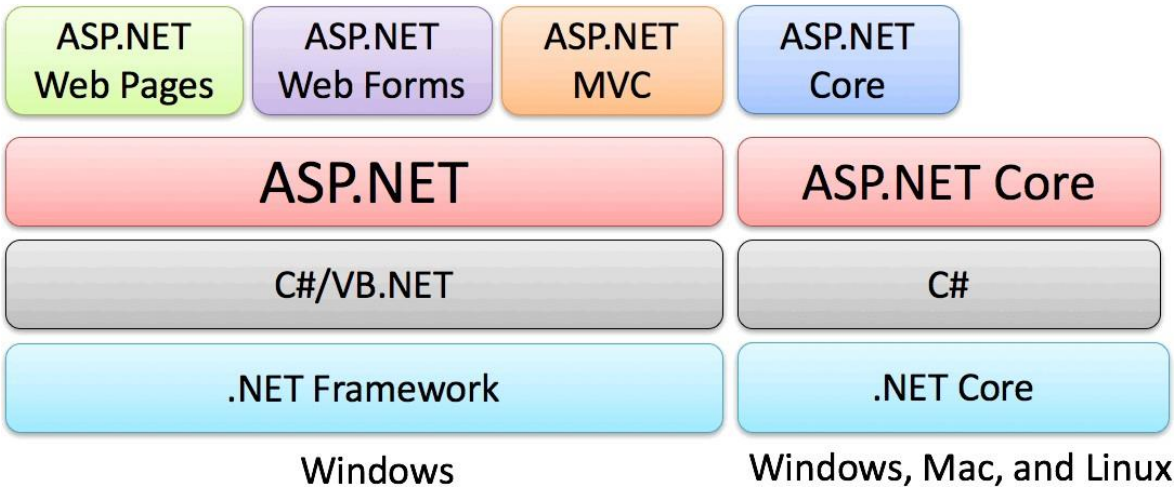
CSS makes the maintenance of the website easier. It plays an essential role in website maintenance.

Multiple device compatibility

CSS make compatible with different devices and platforms

ASP.NET

Web Programming



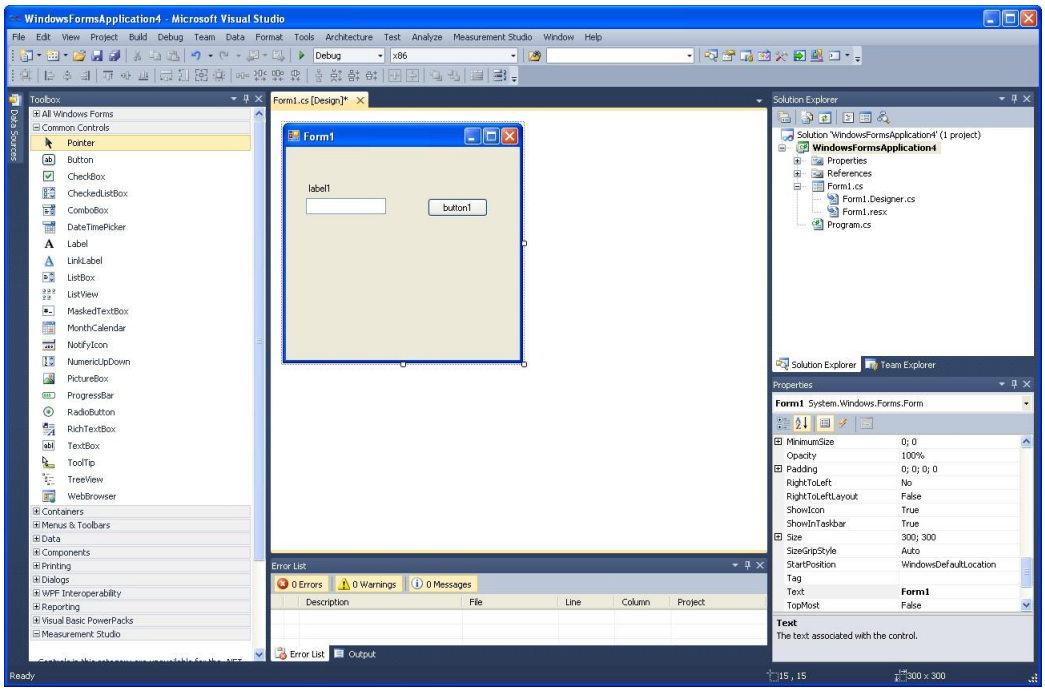
Introduction

ASP.NET is a framework for creating web sites, apps and services with HTML, CSS and JavaScript.

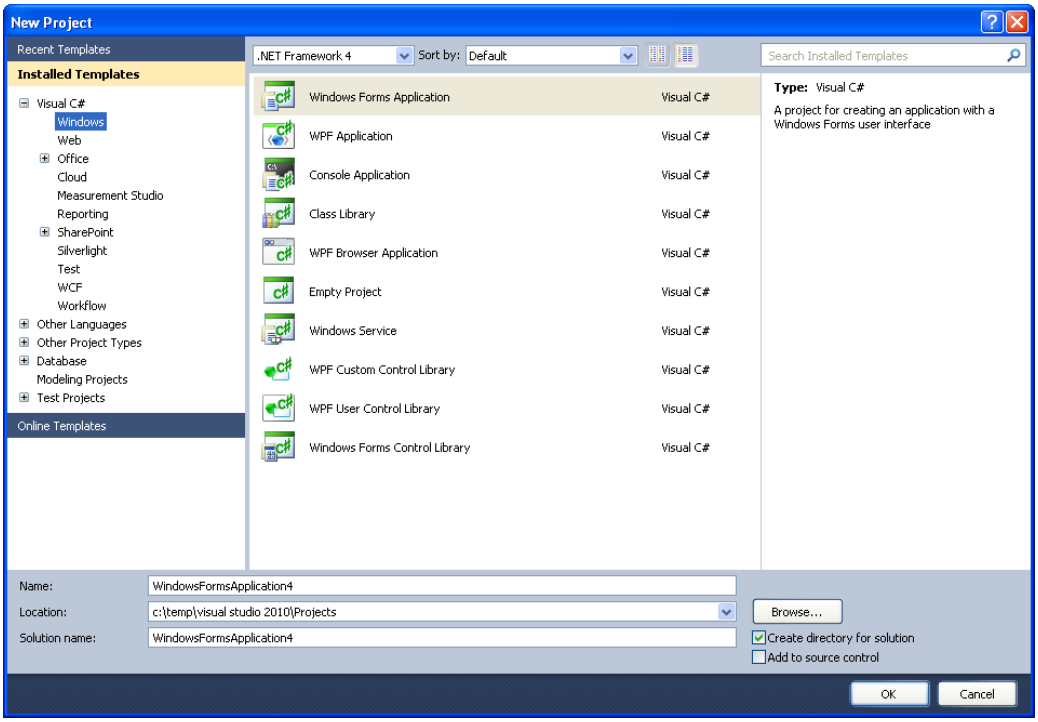
Visual Studio

Microsoft Visual Studio is an integrated development environment (IDE) from Microsoft. It can be used to develop console and graphical user interface applications along with Windows Forms applications, web sites, web applications, and web services in both native code together with managed code for all platforms supported by Microsoft Windows, Windows Phone, Windows CE, .NET Framework, .NET Compact Framework and Microsoft Silverlight.

Below we see the integrated development environment (IDE) in Visual Studio:



New projects are created from the “New Project” window:



C#

C# is pronounced “see sharp”. C# is an object-oriented programming language and part of the .NET family from Microsoft. C# is very similar to C++ and Java. C# is developed by Microsoft and works only on the Windows platform.

1.1 .NET Framework

The .NET Framework (pronounced “dot net”) is a software framework that runs primarily on Microsoft Windows. It includes a large library and supports several programming languages which allow language interoperability (each language can use code written in other languages). The .NET library is available to all the programming languages that .NET supports. Programs written for the .NET Framework execute in a software environment, known as the Common Language Runtime (CLR), an application virtual machine that provides important services such as security, memory management, and exception handling. The class library and the CLR together constitute the .NET Framework.

1.2 Object-Oriented Programming (OOP)

Object-oriented programming (OOP) is a programming language model organized around "objects" rather than "actions" and data rather than logic. Historically, a program has been viewed as a logical procedure that takes input data, processes it, and produces output data.

The first step in OOP is to identify all the objects you want to manipulate and how they relate to each other, an exercise often known as data modeling. Once you've identified an object, you generalize it as a class of objects and define the kind of data it contains and any logic sequences that can manipulate it. Each distinct logic sequence is known as a method. A real instance of a class is called an “object” or an “instance of a class”. The object or class instance is what you run in the computer. Its methods provide computer instructions and the class object characteristics provide relevant data. You communicate with objects - and they communicate with each other.

Important features with OOP are:

- Classes and Objects
- Inheritance
- Polymorphism
- Encapsulation

Simula was the first object-oriented programming language. Simula was developed in the 1960s by Kristen Nygaard from Norway.

Java, Python, C++, Visual Basic .NET and C# are popular OOP languages today.

Since Simula-type objects are reimplemented in C++, Java and C# the influence of Simula is often understated. The creator of C++ (1979), Bjarne Stroustrup (from Denmark), has acknowledged that Simula was the greatest influence on him to develop C++.

2 Visual Studio

2.1 Getting Started

2.1.1 Integrated Development Environment (IDE)

The Visual Studio product family shares a single integrated development environment (IDE) that is composed of several elements: the Menu bar, Standard toolbar, various tool windows docked or auto-hidden on the left, bottom, and right sides, as well as the editor space. The tool windows, menus, and toolbars available depend on the type of project or file you are working in.

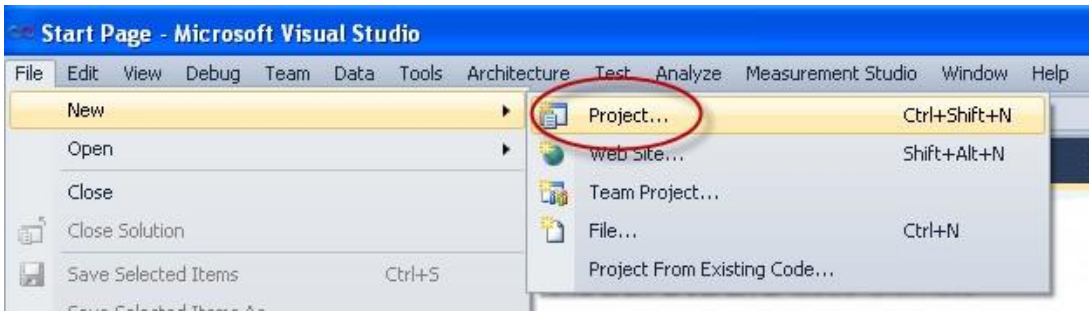
Below we see the Visual Studio IDE (Integrated Development Environment):

2.1.2 New Project

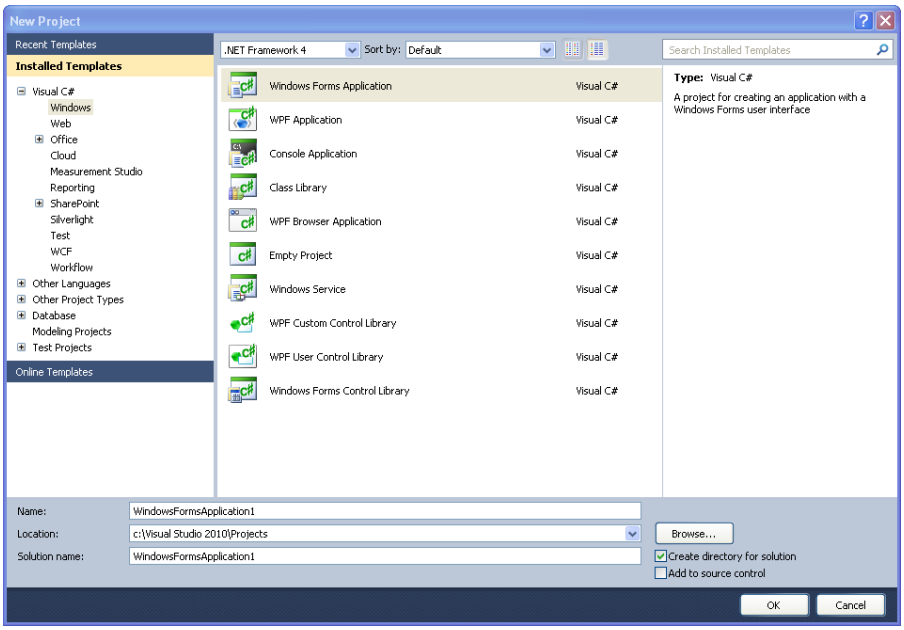
The first thing you do when you want to create a new application is to create a new project.

This can be done from the Start Page:

Or from the File menu:



Then the "New Project" window appears:



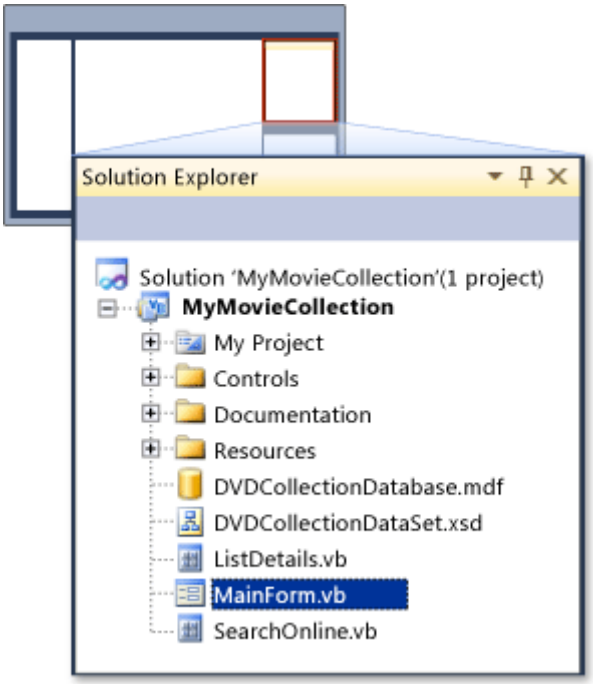
In this window you will select an appropriate template based on what kind of application you want to create, and a name and location for your project and solution.

The most common applications are:

- Windows Form Application
- Console Application
- WPF Application
- ASP.NET Web Application
- Silverlight Application

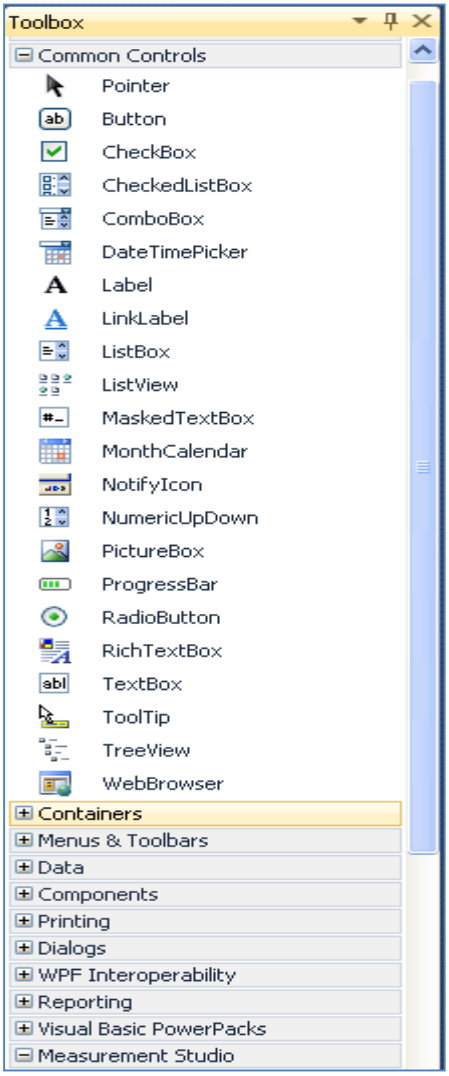
2.1.3 Solution Explorer

Solutions and projects contain items that represent the references, data connections, folders, and files that you need to create your application. A solution container can contain multiple projects and a project container typically contains multiple items.



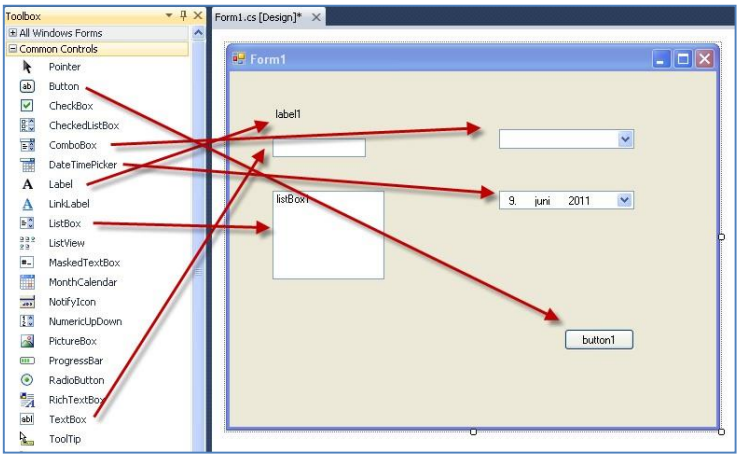
2.1.4 Toolbox

The Toolbox contains all the necessary controls, etc. you need to create your user interface. See Figure below.



The Toolbox contains all the controls, etc. we can use in our user interface.

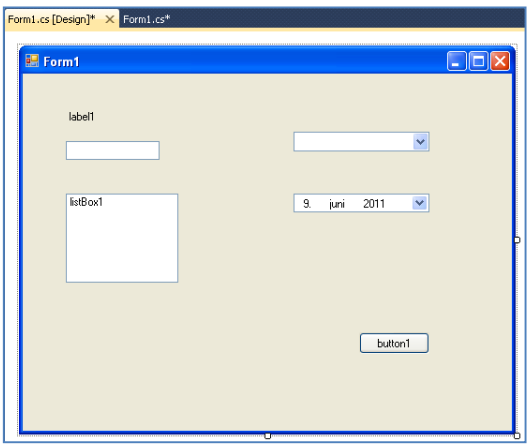
In order to use them in our user interface, we just drag and drop them to the "Form", as shown below:



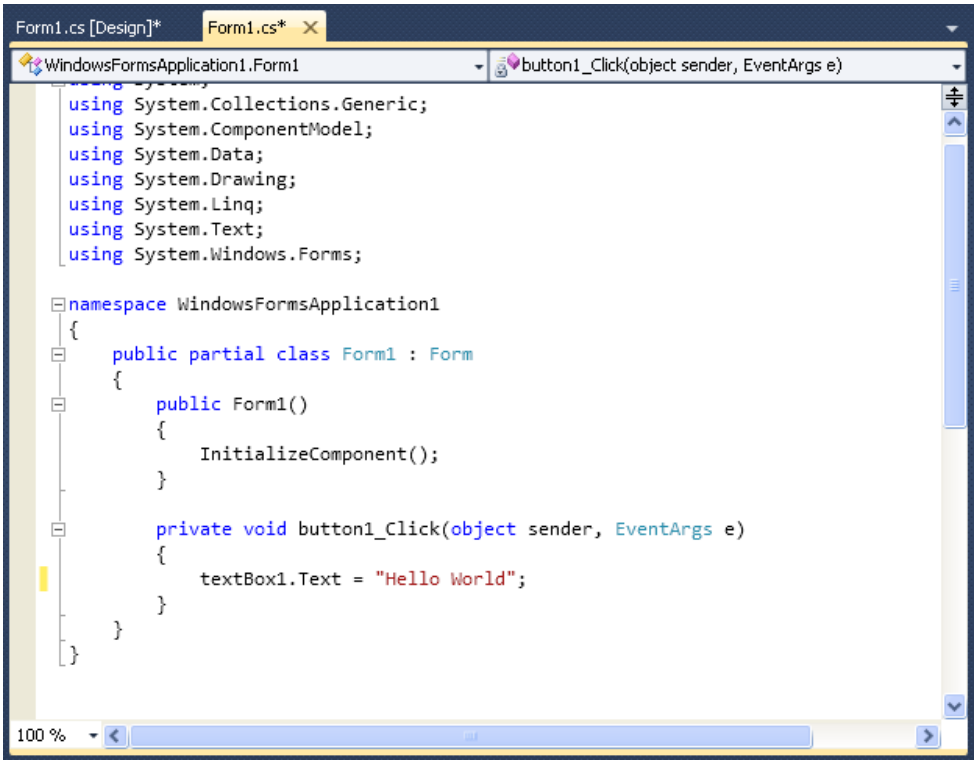
2.1.5 Editors and Designers

Visual Studio has different editors and design tools.

Graphical User Interface Designer:



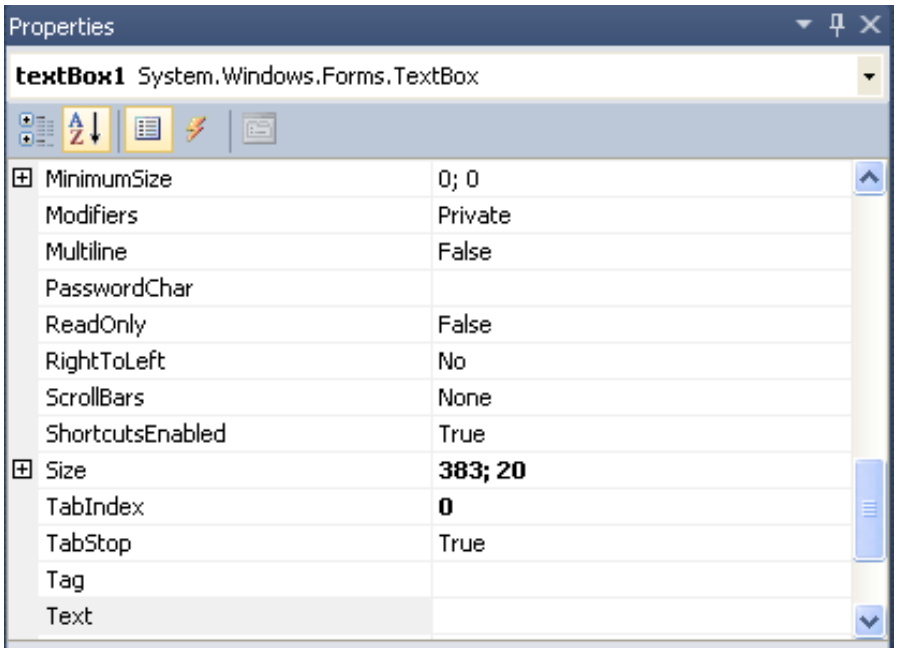
Code Editor:



2.1.6 Properties window

Each control we have on our user interface has lots of Properties we can set.

This is done in the Properties window:

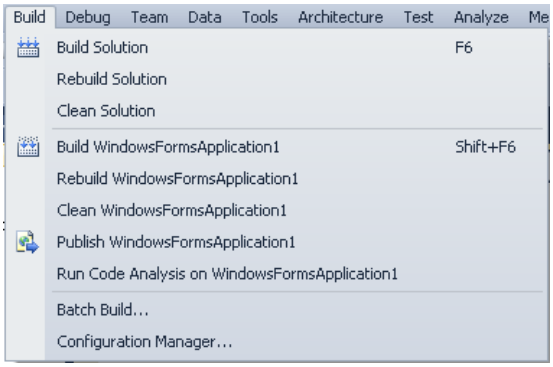


2.1.7 Build and Debug Tools

In Visual Studio we have lots of Build and Debugging Tools.

Build menu:

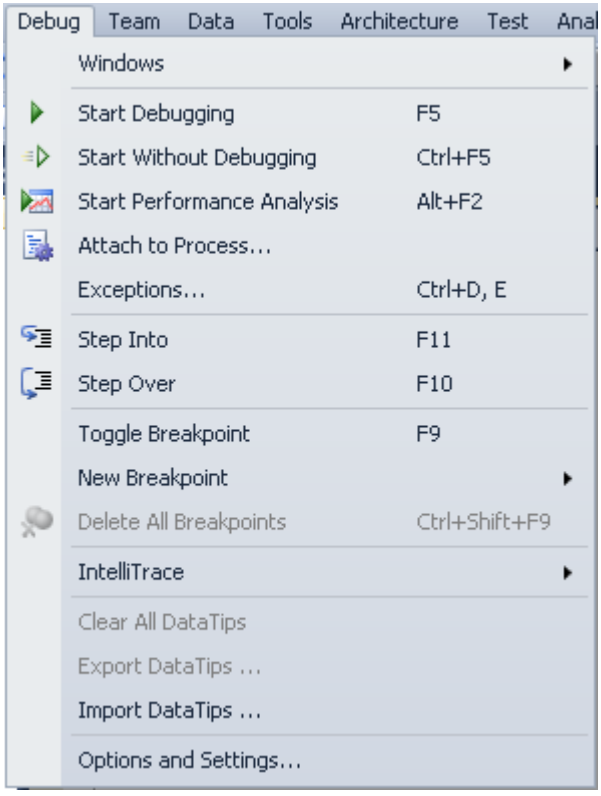
Below we see the **Build** menu:



The most used tool is "**Build Solution**" (Shortcut Key: F6).

Debug menu:

Below we see the **Debug** menu:



The most used tool is "**Start Debugging**" (Shortcut Key: F5).

3 Introduction to Web Programming

3.1 Introduction

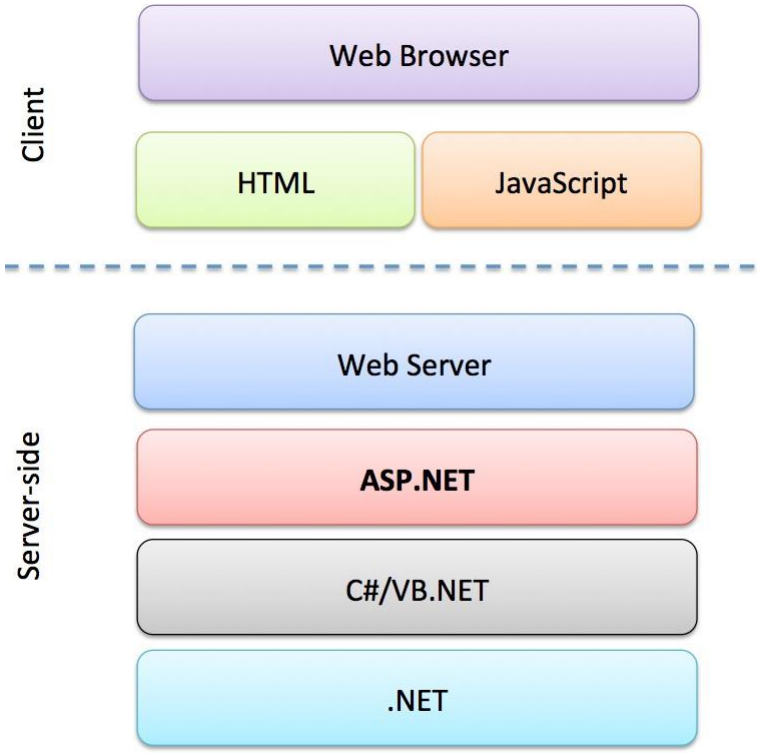
Today most of the applications focus on the Internet, where the applications can be viewed in a standard Web Browser. Static Web pages are based on HTML and CSS. In order to create more advanced applications, we need more powerful tools.

Important frameworks and tools for creating dynamic web pages:

- ASP.NET
- AJAX/ ASP.NET AJAX
- JavaScript

- Silverlight

These frameworks and tools will be explained below.



3.2 Web Browser

The purpose of a web browser is to read HTML documents and compose them into visual or audible web pages. The browser does not display the HTML tags, but uses the tags to interpret the content of the page.

Today we have the following major Web Browsers:

- Internet Explorer (by Microsoft)
- Firefox (by Mozilla)
- Chrome (by Google)
- Safari (by Apple)
- Opera (by Opera from Norway)

3.3 HTML

HTML, which stands for HyperText Markup Language, is the predominant markup language for web pages. HTML is the basic building-blocks of webpages.

HTML is written in the form of HTML elements consisting of tags, enclosed in angle brackets (like <html>), within the web page content. HTML tags normally come in pairs like <h1> and </h1>. The first tag in a pair is the start tag, the second tag is the end tag (they are also called opening tags and closing tags). In between these tags web designers can add text, tables, images, etc.

Below we see a simple a HTML web page:

```
<!DOCTYPE html>
<html>
<body>

<h1>My First Heading</h1>

<p>My first paragraph.</p>

</body>
</html>
```

A useful web site for learning more about HTML: http://www.w3schools.com/html

3.4 CSS

Web browsers can also refer to Cascading Style Sheets (CSS) to define the appearance and layout of text and other material.

The W3C, maintainer of both the HTML and the CSS standards

A useful web site for learning more about CSS: http://www.w3schools.com/css

3.5 JavaScript

JavaScript is an object-oriented scripting language basically used to create dynamic web pages. JavaScript is primarily used in the form of client-side JavaScript, implemented as part of a web browser in order to provide enhanced user interfaces and dynamic websites.

Below we see a simple a HTML web page with JavaScript:

```
<!DOCTYPE html>
<html>
<body>

<h1>My First JavaScript</h1>
<p>Click the button to display the date.</p>
<p id="demo"></p>

<button type="button" onclick="myFunction()">Try it</button>

<script>
function myFunction()
{
  document.getElementById("demo").innerHTML = Date();
}
```

```
</script>

</body>
</html>
```

3.6 ASP.NET

ASP.NET is a web application framework developed by Microsoft to allow programmers to build dynamic web sites, web applications and web services.

ASP.NET is part of the Visual Studio package.

It was first released in January 2002 with version 1.0 of the .NET Framework, and is the successor to Microsoft's Active Server Pages (ASP) technology. ASP.NET is built on the Common Language Runtime (CLR), allowing programmers to write ASP.NET code using any supported .NET language, such as C# and VB.NET.

ASP.NET web pages or webpage, known officially as Web Forms], are the main building block for application development. Web forms are contained in files with an ".aspx" extension.

3.7 AJAX/ ASP.NET AJAX

AJAX is an acronym for Asynchronous JavaScript and XML. AJAX is a group of interrelated web development methods used on the client-side to create interactive web applications. With Ajax, web applications can send data to, and retrieve data from, a server asynchronously (in the background) without interfering with the display and behavior of the existing page.

ASP.NET AJAX is a set of extensions to ASP.NET developed by Microsoft for implementing AJAX functionality.

3.8 Silverlight

Microsoft Silverlight is an application framework for writing and running browser plug-ins or other rich internet applications, with features and purposes similar to those of Adobe Flash. The run-time environment for Silverlight is available as a plug-in for most web browsers. Silverlight is also one of the two application development platforms for Windows Phone 7/8.

The latest version is Silverlight 5.0.

Silverlight is based on WPF, so in Silverlight applications, user interfaces are declared in Extensible Application Markup Language (XAML) and programmed using a subset of the .NET Framework.

4 Introduction to ASP.NET

4.1 Introduction

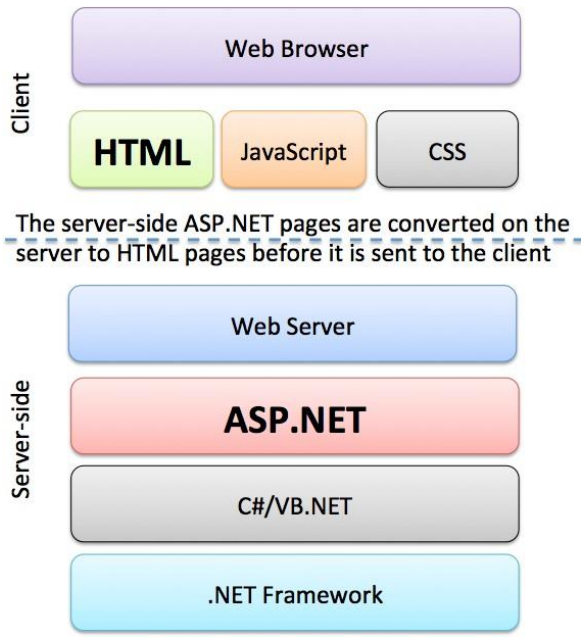
ASP.NET is a web application framework developed by Microsoft to allow programmers to build dynamic web sites, web applications and web services.

ASP.NET is a framework fro creating web sites, apps and services with HTML, CSS and JavaScript.

ASP.NET supports different approaches for making web sites or web pages.

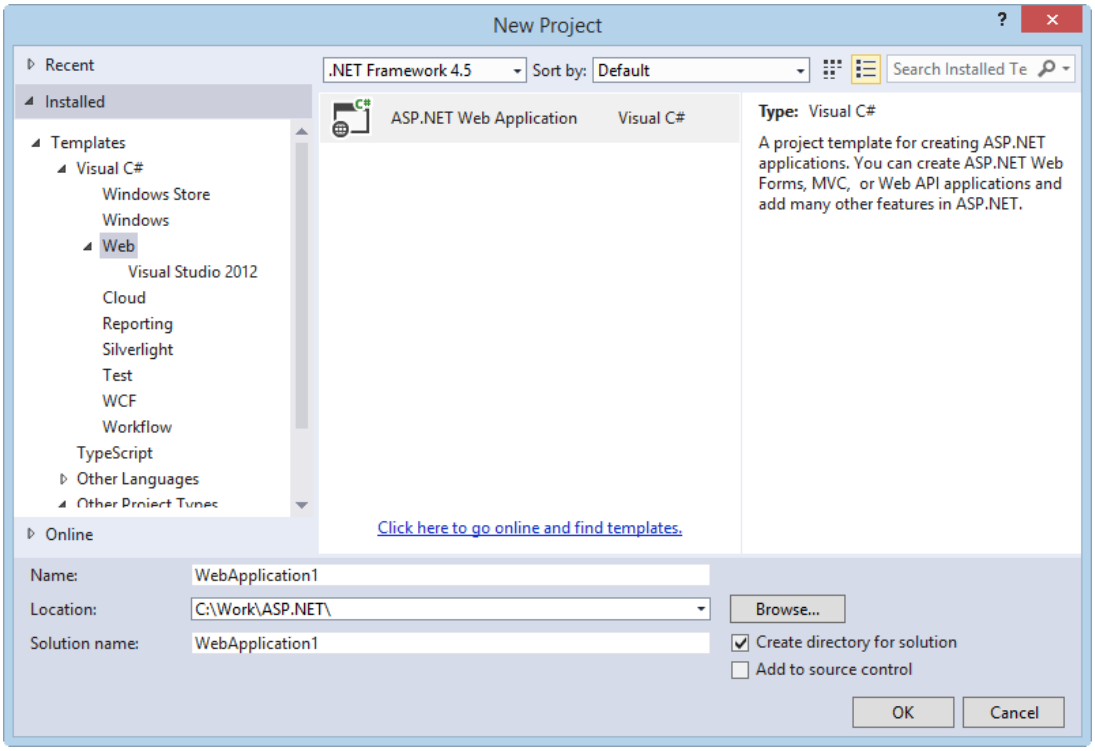
- ASP.NET Web Pages uses a single page model that mixes code and HTML markup.
- ASP.NET SPA (Single Page Application)
- ASP.NET Web Forms uses controls and an event-model for component-based development.
- ASP.NET MVC (Model View Controller) values separation of concerns and enables easier test-driven development.

You can mix and match these techniques within one application depending on your needs.

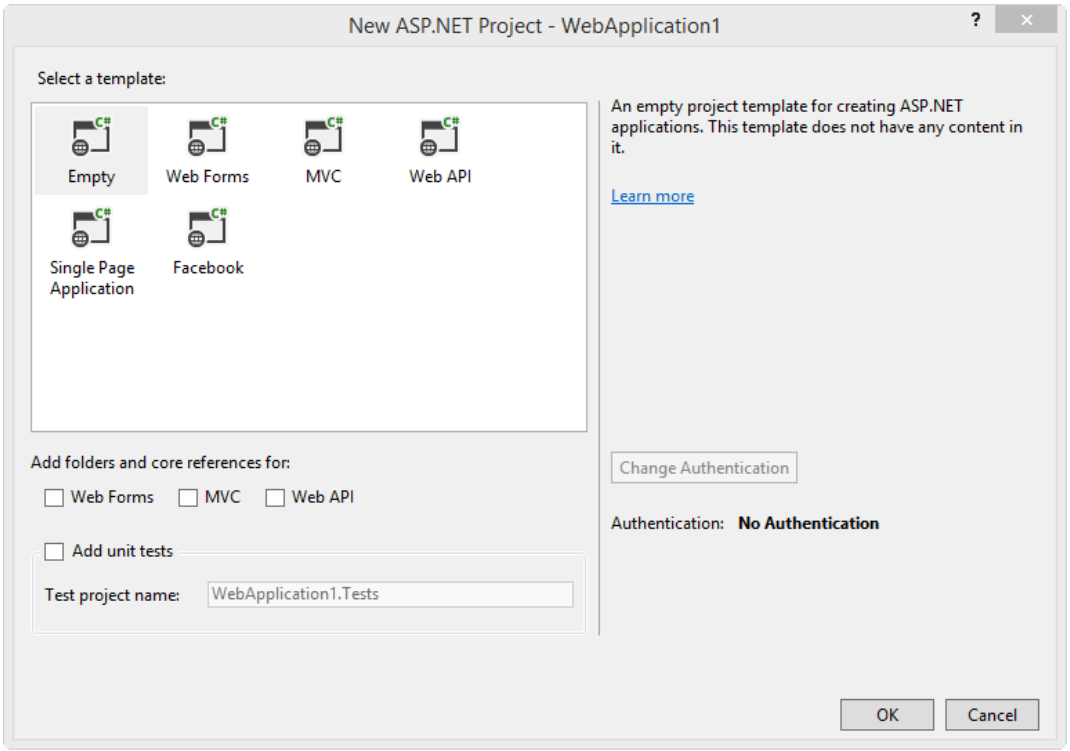


New Web Project Dialog:

Select Project Name, Loction and Solution Name.



Next, seelct the template you want to use.



4.2 ASP.NET Web Pages

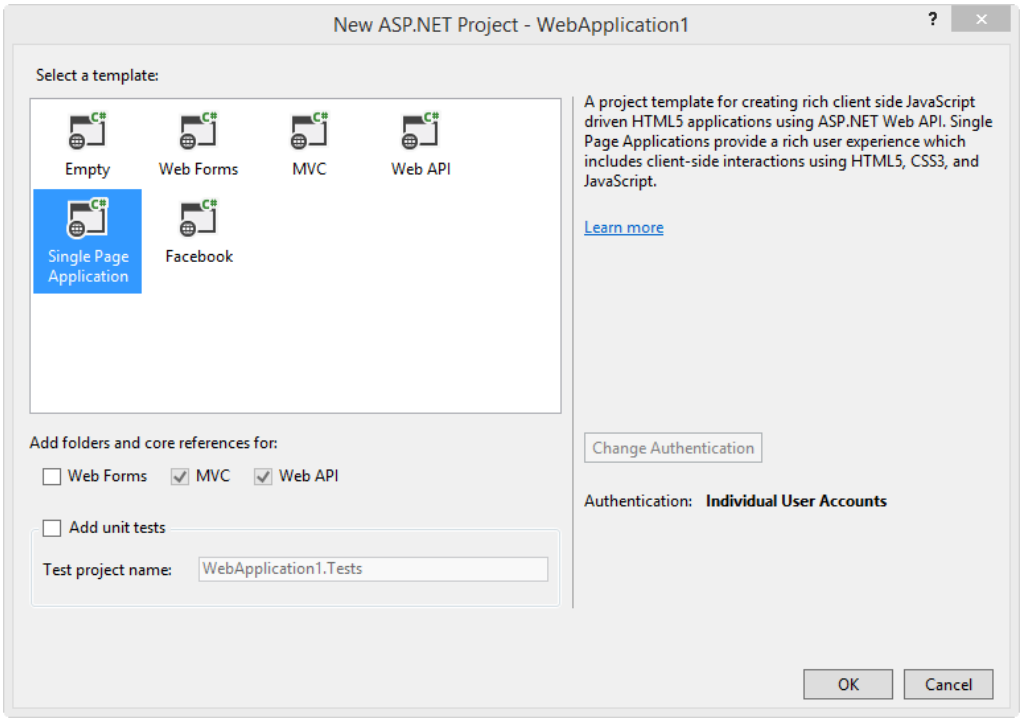
In ASP.NET Web Pages you include server code directly into the HTML syntax, similar to PHP and Classic ASP.

ASP.NET Web Pages uses Singe Page Model (The server-side code is mixed in between the HTML). It uses the the **Razor** syntax (.cshtml files)

A tool called Microsoft WebMatrix is optimized for this development model, but you can also use Visual Studio if you want to.

4.3 ASP.NET

SPA – Single Page Application



4.4 ASP.NET MVC

MVC – Model - View – Controller

A new development model where you split your development into 3 parts/components: Models for Data, Views for Display and Controllers for Input.

4.5 ASP.NET Web Forms

ASP.NET is similiar to the desktop development model used in WPF. GUI (*.aspx files) and Code (*.aspx.cs) is separated.

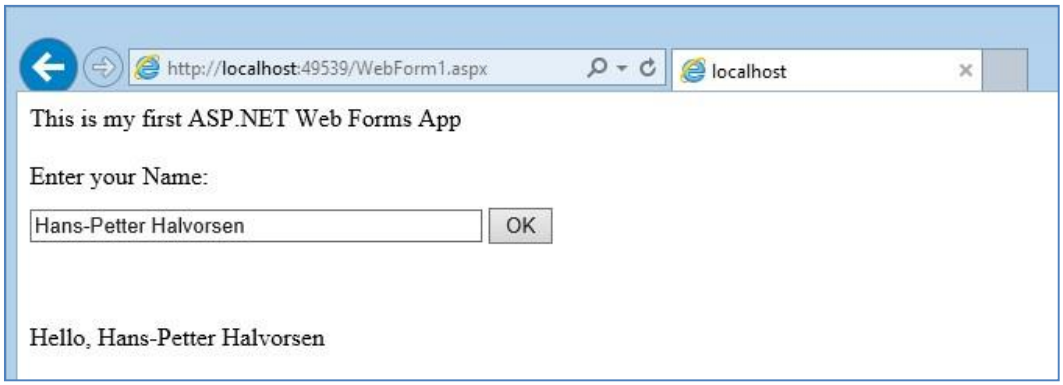
In this document we will focus on ASP.NET Web Forms.

5 ASP.NET Web Forms

The main focus in this tutorial will be ASP.NET Web Forms.

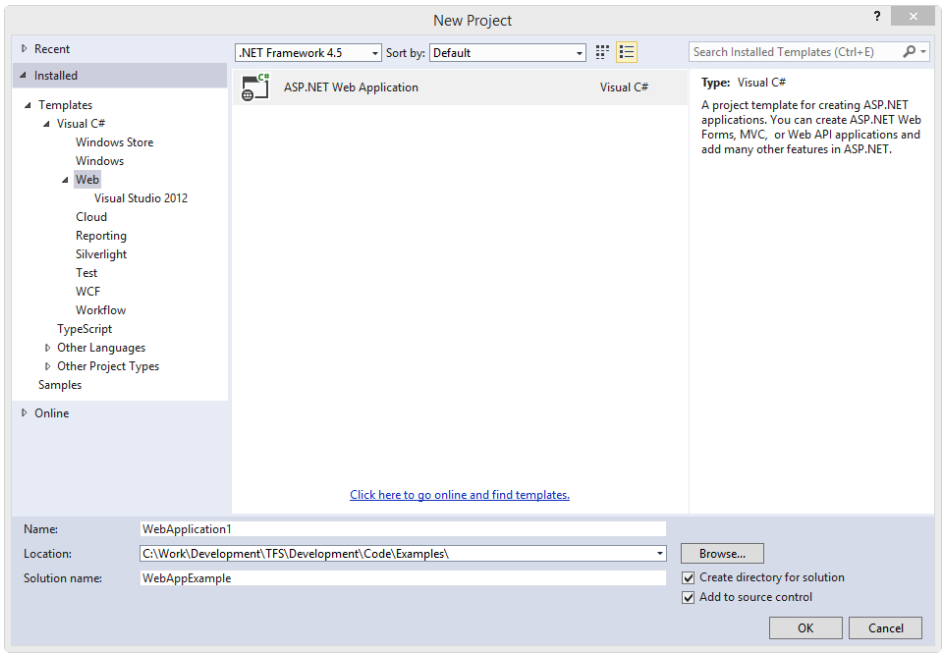
5.1 Hello World Example

When you are finished, your Web App should look something like this:

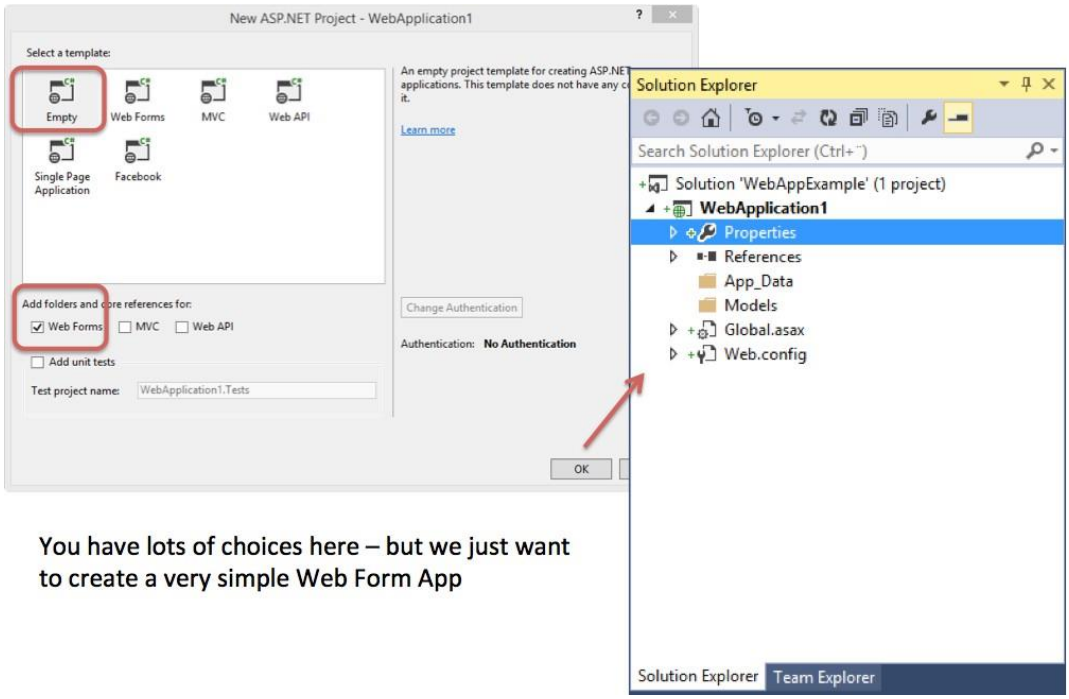


When you enter your Name in the TextBox and click the OK Button, the program should respond with a greetings.

Seelct Create New Project:

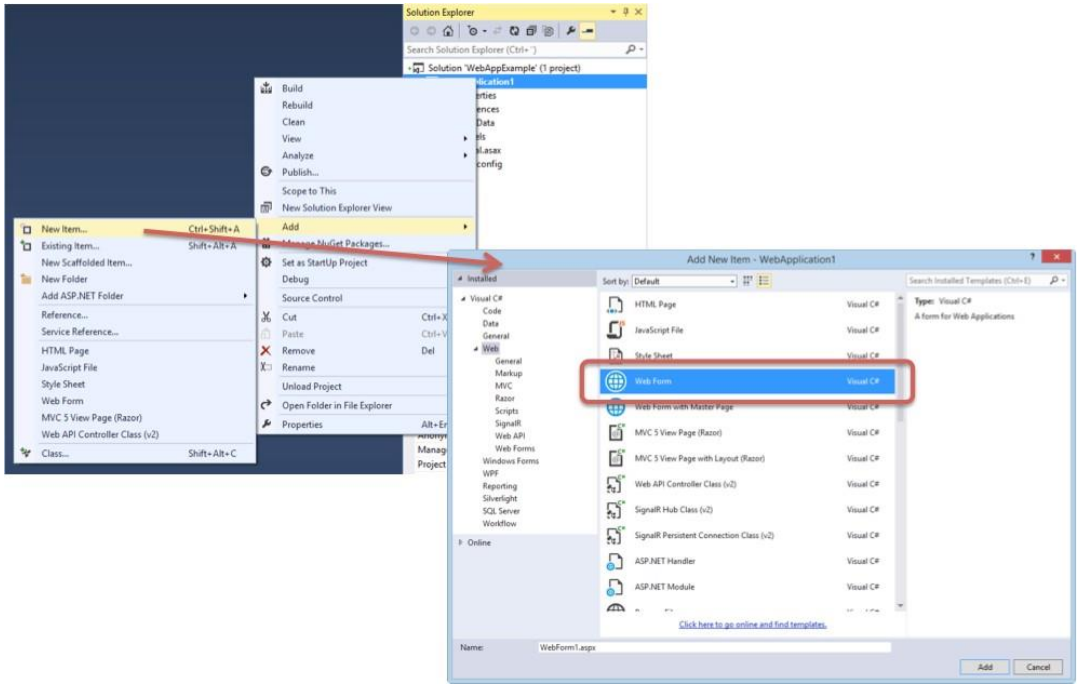


You have lots of choices here – but we just want to create a very simple Web Form App.

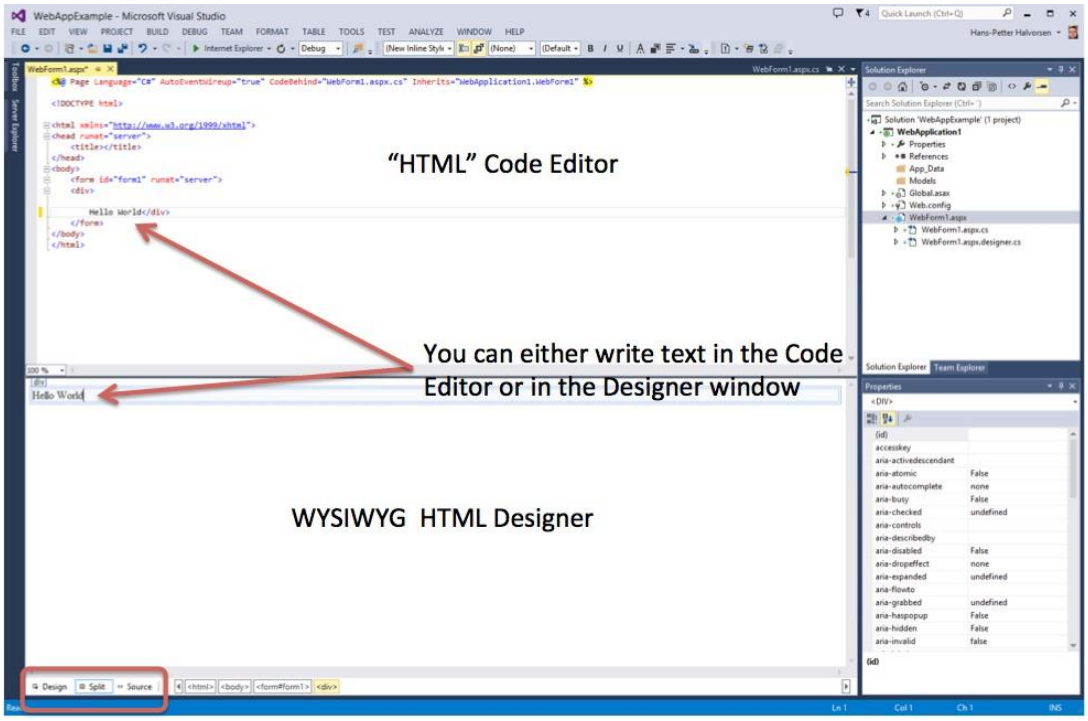


You have lots of choices here – but we just want to create a very simple Web Form App

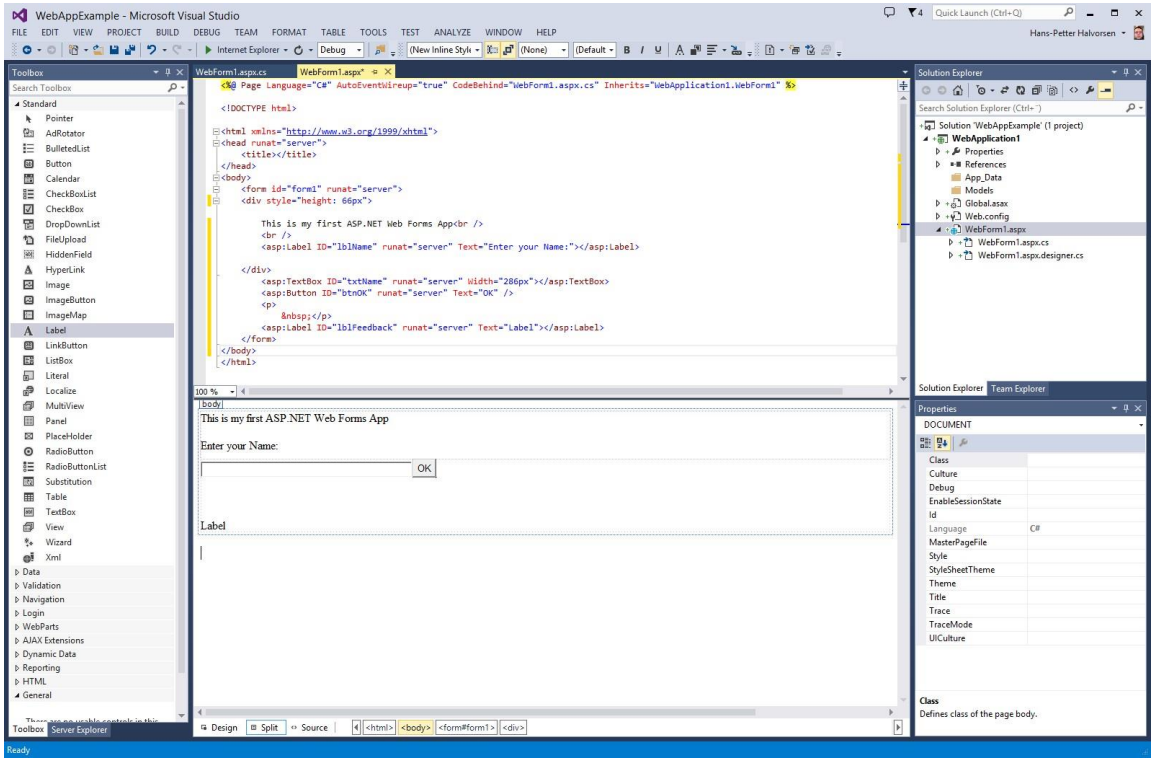
Create New Web Form:



WinForm1.aspx Page: Used to create the GUI/HMI



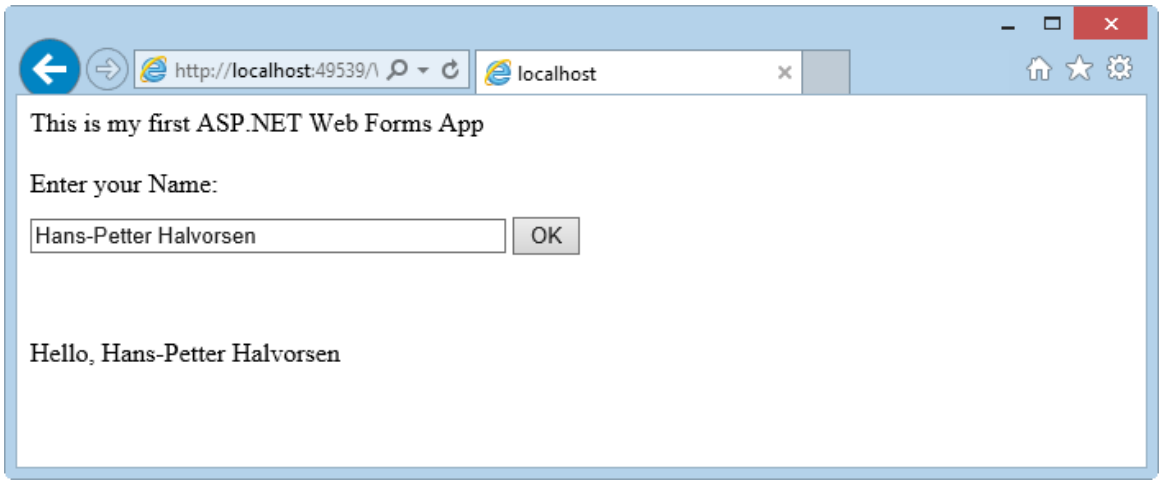
Hello world Example – GUI:



WebForm1.aspx.cs:

```
WebForm1.aspx.cs | WebForm1.aspx
WebApplication1.WebForm1 | Page_Load(object sender, EventArgs e)

using System;
using System.Collections.Generic;
using System.Linq;
using System.Web;
using System.Web.UI;
using System.Web.UI.WebControls;

namespace WebApplication1
{
    public partial class WebForm1 : System.Web.UI.Page
    {
        protected void Page_Load(object sender, EventArgs e)
        {

        }

        protected void btnOK_Click(object sender, EventArgs e)
        {
            lblFeedback.Text = "Hello, " + txtName.Text;
        }
    }
}
```

Running the Application (F5):



Congratulations, you have created your first ASP.NET Web Forms application.

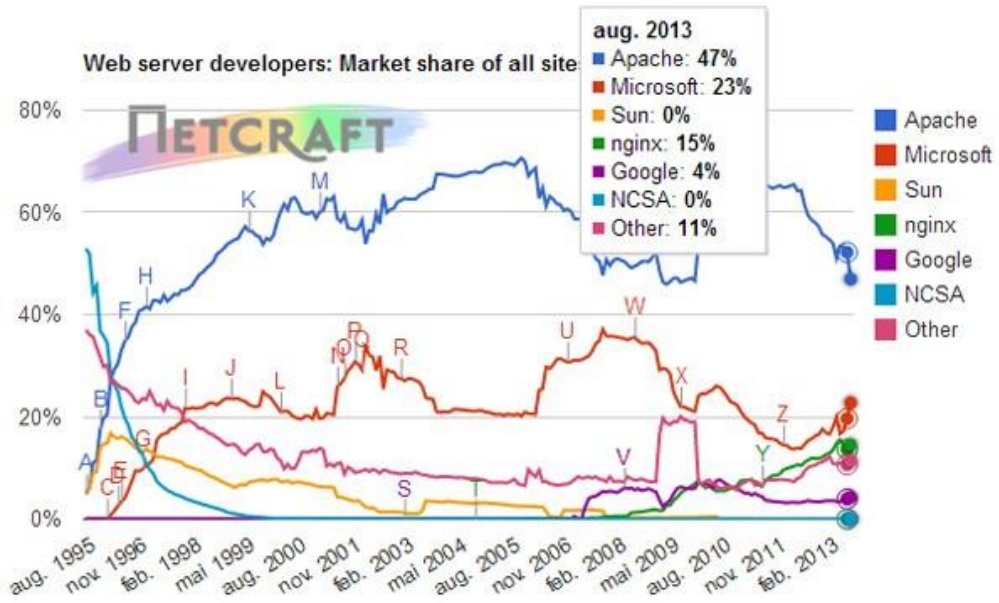
6 Internet Information Services (IIS)

6.1 Web Server

The term web server can refer to either the hardware (the computer) or the software (the computer application) that helps to deliver web content that can be accessed through the Internet.

The most common use of web servers is to host websites, but there are other uses such as gaming, data storage or running enterprise applications.

- IIS - Internet Information Services, Microsoft Windows
- Apache Web Server, Open Source, Cross-platform: UNIX, Linux, OS X, Windows, ...
- Nginx (pronounced "engine x") - Has become very popular latly
- GWS (Google Web Server)
- etc.



7 Working with Databases in ASP.NET

Today, most of the web sites and web pages uses a database to store data that is visible on the web page, e.g. Facebook, Instagram, Twitter, etc.

7.1 Database Systems

There are lots of different database systems, or DBMS – Database Management Systems, such as:

- Microsoft SQL Server
  - Enterprise, Developer versions, etc.
  - Express version is free of charge
- Oracle
- MySQL (Oracle, previously Sun Microsystems) - MySQL can be used free of charge (open source license), Web sites that use MySQL: YouTube, Wikipedia, Facebook
- MariaDB ("The New MySQL")
- Microsoft Access
- IBM DB2
- Sybase
- … lots of other systems

7.1.1 Mirosoft SQL Server

Microsoft is the vendor of SQL Server. We have different editions of SQL Server, where SQL Server Express is free to download and use. We will focus on SQL Server in this tutorial.

7.2 SQL

SQL (Structured Query Language) is a database computer language designed for managing data in relational database management systems (RDBMS).

SQL, is a standardized computer language that was originally developed by IBM for querying, altering and defining relational databases, using declarative statements.

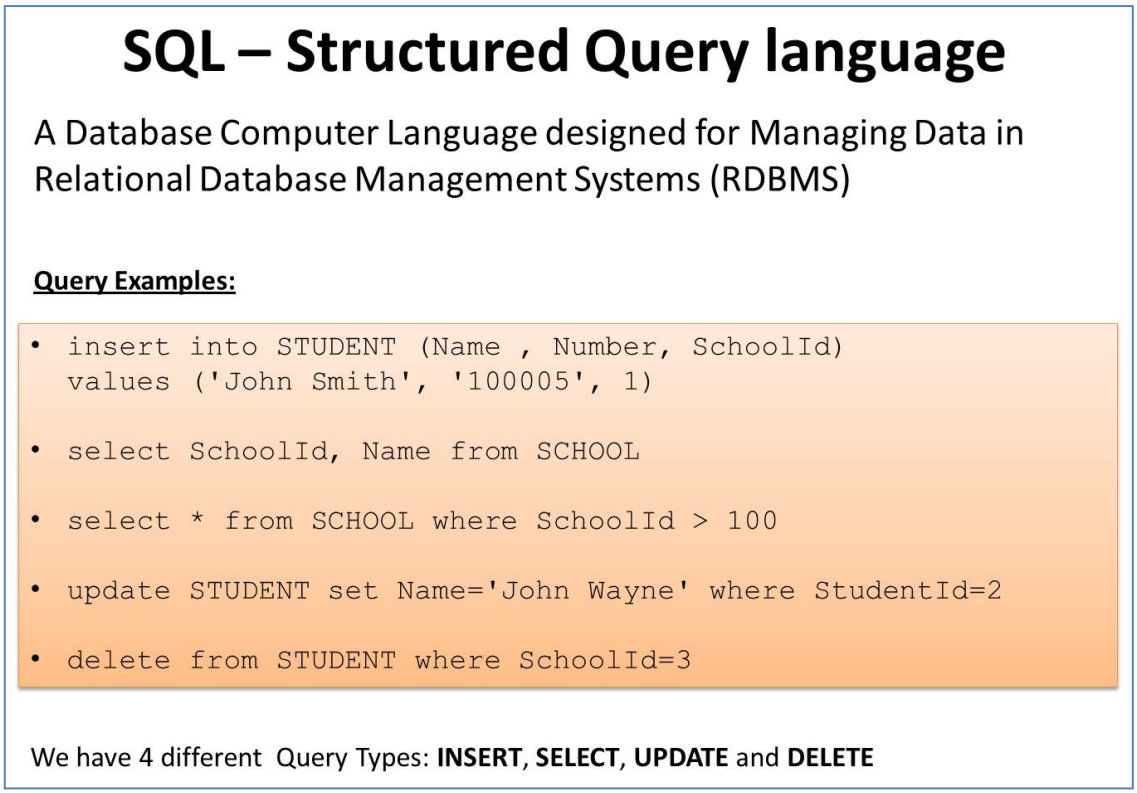
SQL is pronounced /ˌɛs kjuː ˈɛl/ (letter by letter) or /ˈsiːkwəl/ (as a word).

What can SQL do?

- SQL can execute queries against a database
- SQL can retrieve data from a database
- SQL can insert records in a database
- SQL can update records in a database
- SQL can delete records from a database
- SQL can create new databases
- SQL can create new tables in a database
- SQL can create stored procedures in a database
- SQL can create views in a database
- SQL can set permissions on tables, procedures, and views

Even if SQL is a standard, many of the database systems that exist today implement their own version of the SQL language. In this document we will use the Microsoft SQL Server as an example.

Here are some examples of SQL queries:



In this Tutorial we will focus on Microsoft SQL Server. SQL Server uses T-SQL (Transact-SQL). T-SQL is Microsoft's proprietary extension to SQL. T-SQL is very similar to standard SQL, but in addition it supports some extra functionality, built-in functions, etc.

In order to use databases in our applications we need to know **Structured Query language (SQL)**. For more information about SQL, see the following Tutorial: Structured Query Language (SQL).

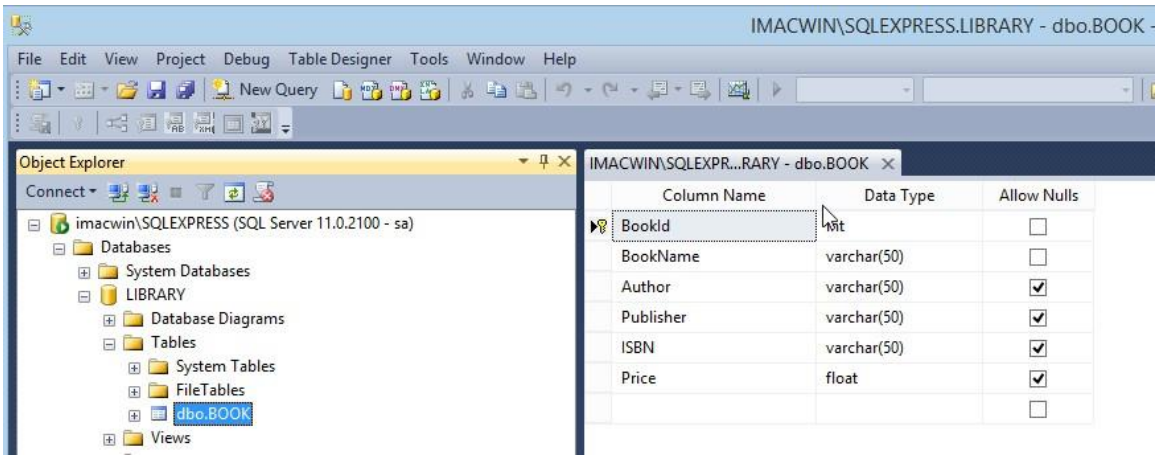
The Tutorials are available from: https://www.halvorsen.blog

7.3 SQL Server + ASP.NET

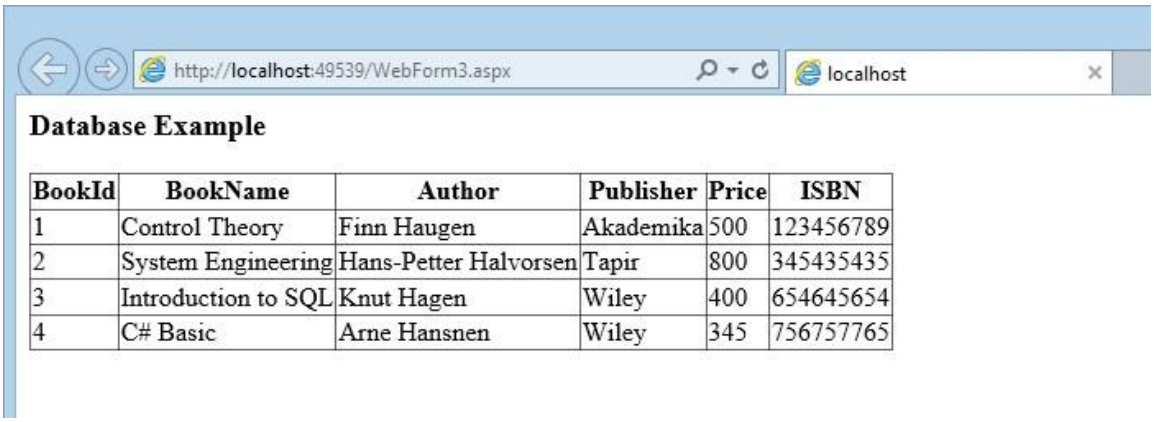
SQL Server + ASP.NET = Powerful Web Applications

7.3.1 Simple Database Examplee

Create a Database called LIBRARY and a BOOK table in SQL Server. Enter some Test data into the BOOK table.

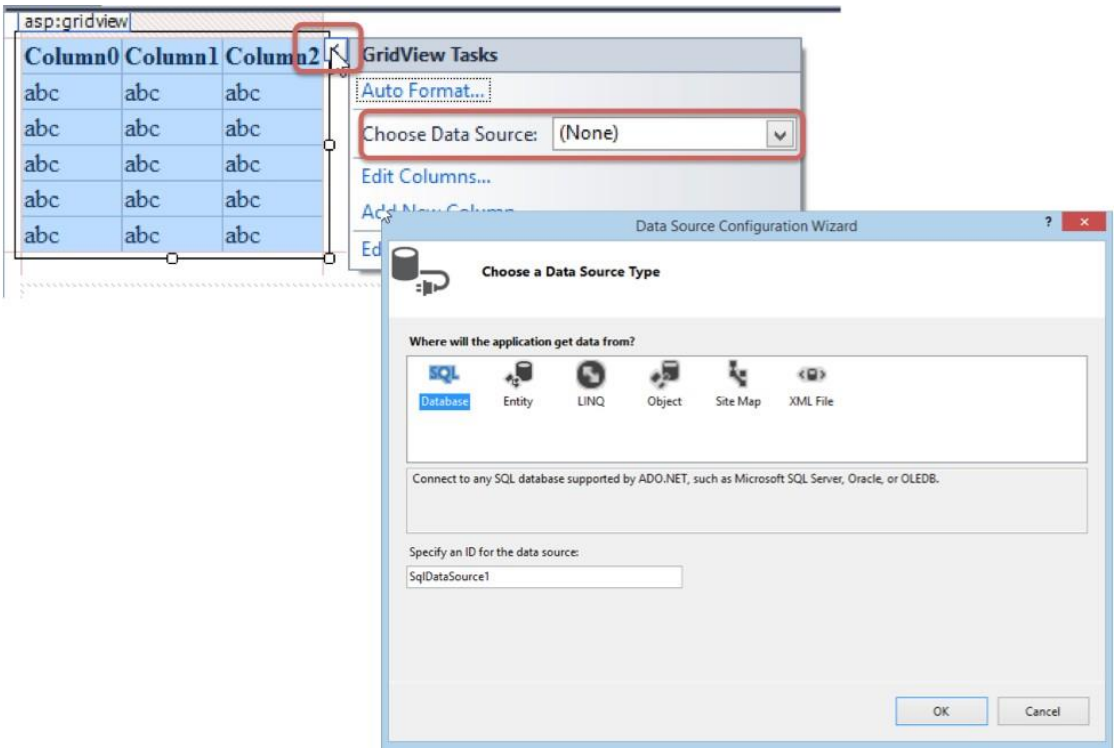


Fill a “GridView” with data from the Database Table (BOOK), as shown below

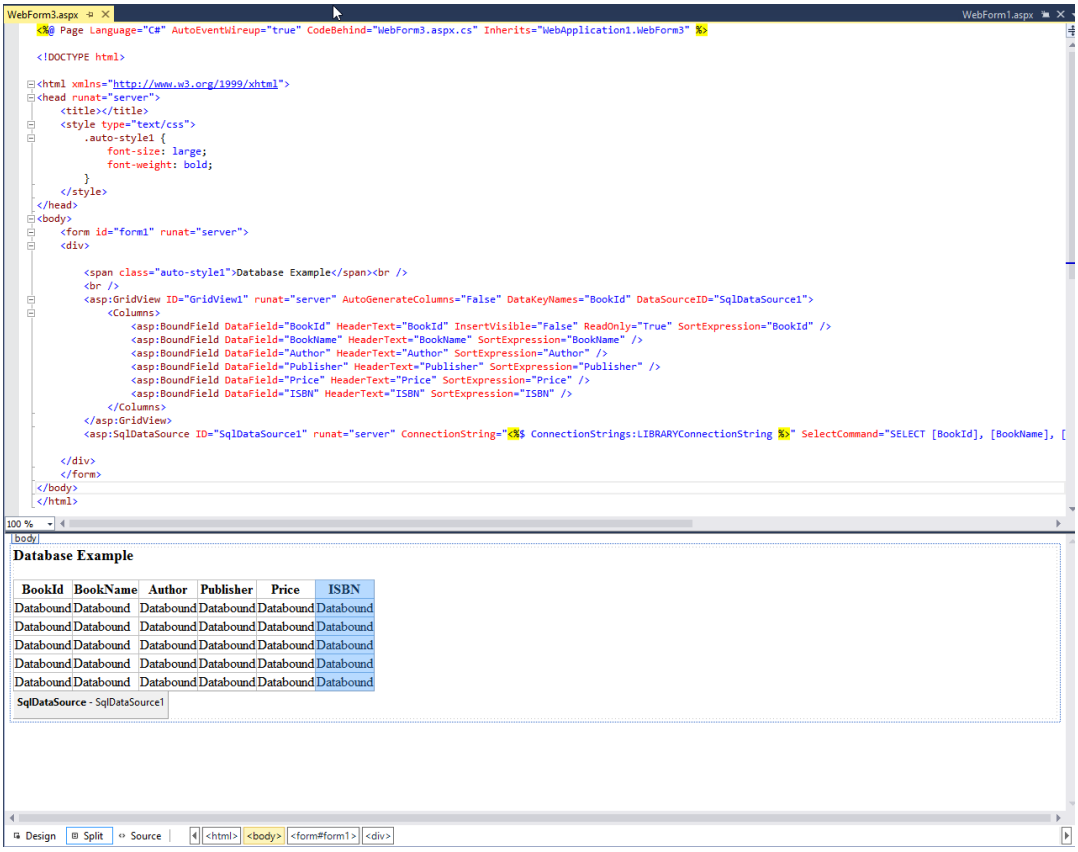


BookId	BookName	Author	Publisher	Price	ISBN
1	Control Theory	Finn Haugen	Akademika	500	123456789
2	System Engineering	Hans-Petter Halvorsen	Tapir	800	345435435
3	Introduction to SQL	Knut Hagen	Wiley	400	654645654
4	C# Basic	Arne Hansnen	Wiley	345	756757765

Use the “Wizard” in order to create all the “magic” you need in your application:



When finishing the “Wizard”, your .aspx page should look something like this



As you see - No Code needed to be written in this example ☺

```
WebForm3.aspx.cs  WebForm3.aspx
WebApplication1.WebForm3                    Page_Load(object sender, EventArgs e)
using System;
using System.Collections.Generic;
using System.Linq;
using System.Web;
using System.Web.UI;
using System.Web.UI.WebControls;

namespace WebApplication1
{
    public partial class WebForm3 : System.Web.UI.Page
    {
        protected void Page_Load(object sender, EventArgs e)
        {

        }
    }
}
```

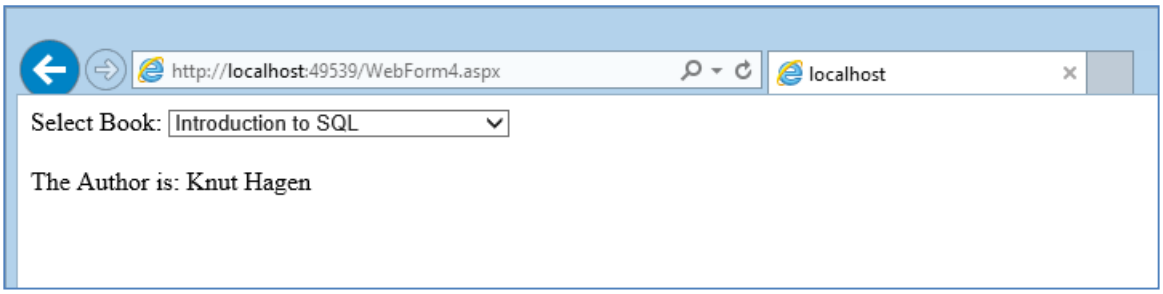
This is OK for quick demos – but for professional applications, you need to use some hardcore ADO.NET or similiar frameworks.

7.4 ADO.NET

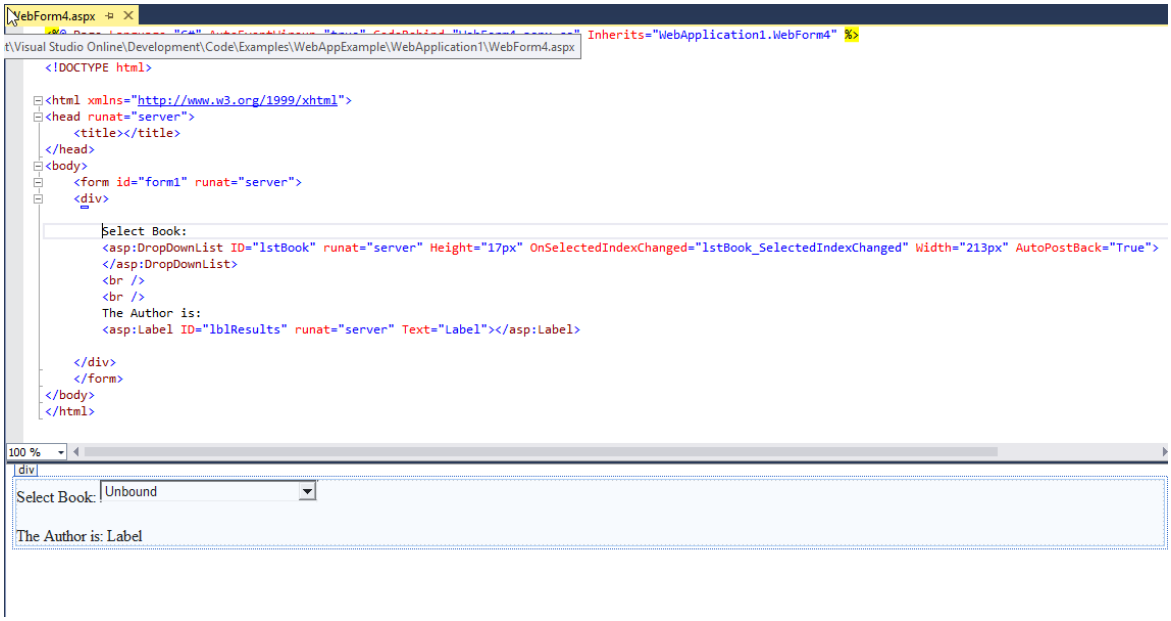
ADO.NET (ActiveX Data Object for .NET) is a set of computer software components that programmers can use to access data and data services. It is a part of the base class library that is included with the Microsoft .NET Framework. It is commonly used by programmers to access and modify data stored in relational database systems, though it can also access data in non-relational sources.

7.4.1 Example

This time: Create everything from scratch using C# code and ADO.NET. Fill a "DropDownList" with Book Names from the Database Print the Author Name based on the selected BookName on the screen, see below.



Your .aspx page should look something like this:



Your .aspx.cs page should look something like this:

```
namespace WebApplication1
{
    public partial class WebForm4 : System.Web.UI.Page
    {

        private string connectionString = WebConfigurationManager.ConnectionStrings["LIBRARYConnectionString"].ConnectionString;

        protected void Page_Load(object sender, EventArgs e)
        {
            if (!IsPostBack)
               {
                    FillBookList();
               }
        }

        protected void lstBook_SelectedIndexChanged(object sender, EventArgs e)
        {

            // Create a Select statement that searches for a record
            // matching the specific author ID from the Value property.
            string selectSQL;
            selectSQL = "SELECT * FROM BOOK ";
            selectSQL += "WHERE BookId='" + lstBook.SelectedItem.Value + "'";

            // Define the ADO.NET objects.
            SqlConnection con = new SqlConnection(connectionString);
            SqlCommand cmd = new SqlCommand(selectSQL, con);
            SqlDataReader reader;

            // Try to open database and read information.
            try
            {
                con.Open();
                reader = cmd.ExecuteReader();
                reader.Read();

                lblResults.Text = reader["Author"].ToString();
                reader.Close();
            }
            catch (Exception err)
            {
                lblResults.Text = "Error getting Data. ";
                lblResults.Text += err.Message;
            }
            finally
            {
                con.Close();
            }
        }
```

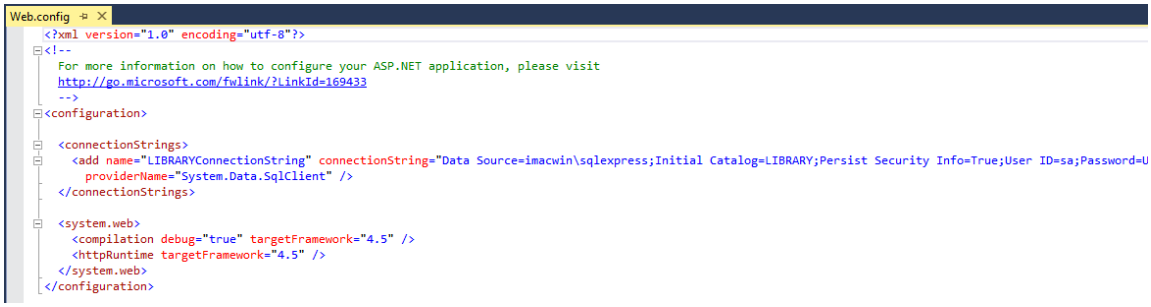
The Page_load() method is executed when the Web Page is loaded. In this simple example we put all the code into the Event Handler for the DropDownList. The code could be improved by creating a separate Class where you put this code into a Method.

The **FillBookList** method is as follows:

```
private void FillBookList()
{
    lstBook.Items.Clear();

    string selectSQL = "SELECT BookId, BookName FROM BOOK";

    // Define the ADO.NET objects.
    SqlConnection con = new SqlConnection(connectionString);
    SqlCommand cmd = new SqlCommand(selectSQL, con);
    SqlDataReader reader;

    // Try to open database and read information.
    try
    {
        con.Open();
        reader = cmd.ExecuteReader();

        while (reader.Read())
        {
            ListItem newItem = new ListItem();
            newItem.Text = reader["BookName"].ToString();
            newItem.Value = reader["BookId"].ToString();
            lstBook.Items.Add(newItem);
        }
        reader.Close();
    }
    catch (Exception err)
    {
        lblResults.Text = "Error reading list of names. ";
        lblResults.Text += err.Message;
    }
    finally
    {
        con.Close();
    }
}
```

The code could be improved by creating a separate Class where you create this Method.

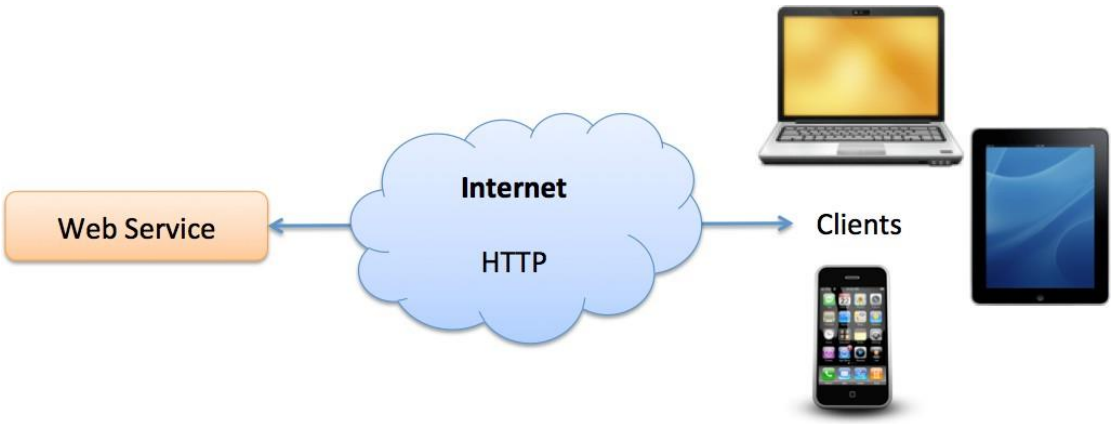
All kind og configuration should be stored in the Web.config file, including Database Connection Strings, etc.



8 Web Services

In order to share data between devices over Internet, Web Services is the solution.

In ASP.NET there are several alternatives for creating such Web Services.



See the Tutorial “Web Services with Examples” for more details.

Visual Studio and ASP.NET has powerful features for creating Web Services:

- ASMX Web Service (Traditional Web Wervice using SOAP)
- ASP.NET Web API (The modern Web Service using REST, Web 2.0)