

INF 272

Advanced Programming



UNIVERSITEIT VAN PRETORIA
UNIVERSITY OF PRETORIA
YUNIBESITHI YA PRETORIA

**Faculty of Engineering,
Built Environment and
Information Technology**
Fakulteit Ingenieurswese, Bou-omgewing en
Inligtingtegnologie / Lefapha la Boetšenere,
Tikologo ya Kago le Theknolotši ya Tshedimošo

HTML and CSS within MVC

In today's session

HTML and CSS within MVC

CSS Box Model

Display and Position CSS properties

Forms – in MVC

Practical overview



UNIVERSITEIT VAN PRETORIA
UNIVERSITY OF PRETORIA
YUNIBESITHI YA PRETORIA

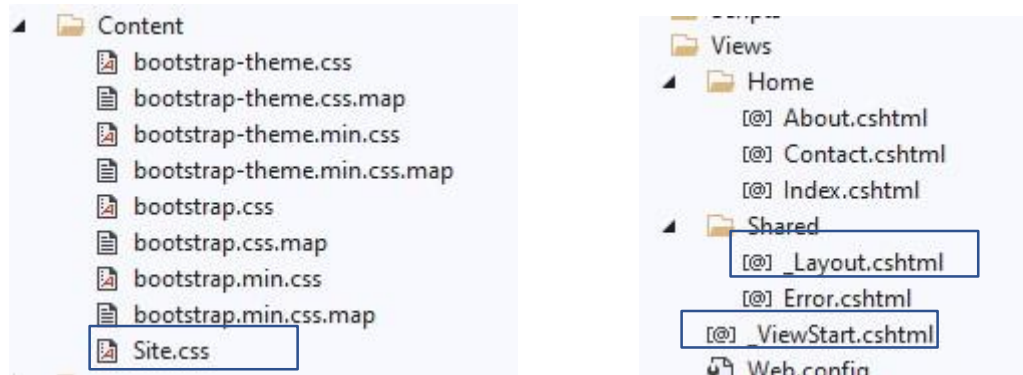
HTML and CSS in MVC

Make today matter

www.up.ac.za

Where do we find HTML and CSS in MVC?

- Consider the Solution Explorer – all files with extension .css (CSS) or .cshtml (HTML)



_ViewStart.cshtml — It is used to specify common settings for all the views under a folder and sub-folders where it is created.

Site.css - the style sheet of the application

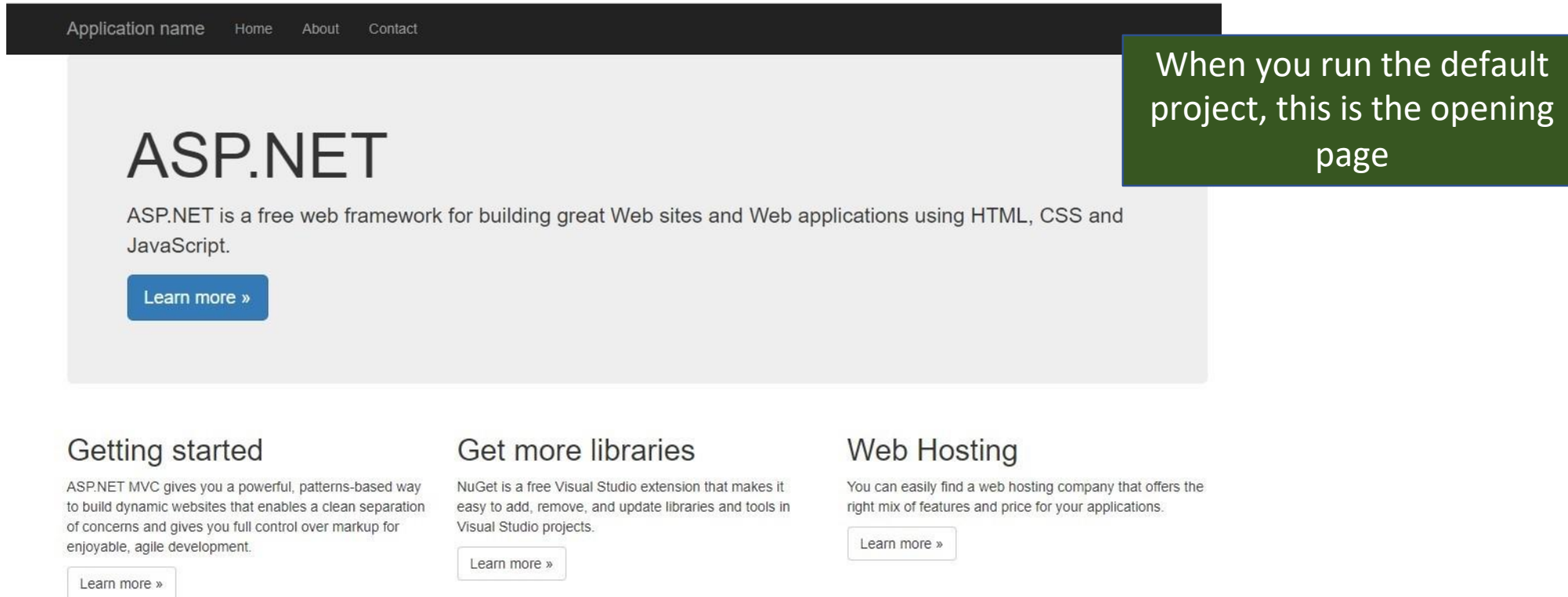
_Layout.cshtml - The file _Layout.cshtml represent the layout of each page in the application.

_ViewStart.cshtml—

```
@{Layout = "~/Views/Shared/_Layout.cshtml";}
```

Automatically added to all views – see <https://www.tutorialsteacher.com/mvc/layout-view-in-asp.net-mvc> for an excellent overview

Default style sheet and layout of the application



The Layout.cshtml page

```
1 <!DOCTYPE html>
2 <html>
3 <head>
4     <meta charset="utf-8" />
5     <meta name="viewport" content="width=device-width, initial-scale=1.0">
6     <title>@ViewBag.Title - My ASP.NET Application</title>
7     @Styles.Render("~/Content/css")
8     @Scripts.Render("~/bundles/modernizr")
9 </head>
10 <body>
11     <div class="navbar navbar-inverse navbar-fixed-top">
12         <div class="container">
13             <div class="navbar-header">
14                 <button type="button" class="navbar-toggle" data-toggle="collapse" data-target=".navbar-collapse">
15                     <span class="icon-bar"></span>
16                     <span class="icon-bar"></span>
17                     <span class="icon-bar"></span>
18                 </button>
19                 @Html.ActionLink("Application name", "Index", "Home", new { area = "" }, new { @class = "navbar-brand" })
20             </div>
21             <div class="navbar-collapse collapse">
22                 <ul class="nav navbar-nav">
23                     <li>@Html.ActionLink("Home", "Index", "Home")</li>
24                     <li>@Html.ActionLink("About", "About", "Home")</li>
25                     <li>@Html.ActionLink("Contact", "Contact", "Home")</li>
26                 </ul>
27             </div>
28         </div>
29     </div>
30     <div class="container body-content">
31         @RenderBody()
32         <hr />
33         <footer>
34             <p>&copy; @DateTime.Now.Year - My ASP.NET Application</p>
35         </footer>
36     </div>
37
38     @Scripts.Render("~/bundles/jquery")
39     @Scripts.Render("~/bundles/bootstrap")
40     @RenderSection("scripts", required: false)
41 </body>
42 </html>
43
```

```

1  <!DOCTYPE html>
2  <html>
3  <head>
4      <meta charset="utf-8" />
5      <meta name="viewport" content="width=device-width, initial-scale=1.0">
6      <title>@ViewBag.Title - My ASP.NET Application</title>
7      @Styles.Render("~/Content/css")
8      @Scripts.Render("~/bundles/modernizr")
9  </head>
10 <body>
11     <div class="navbar navbar-inverse navbar-fixed-top">
12         <div class="container">
13             <div class="navbar-header">
14                 <button type="button" class="navbar-toggle" data-toggle="collapse" data-target=".navbar-collapse">
15                     <span class="icon-bar"></span>
16                     <span class="icon-bar"></span>
17                     <span class="icon-bar"></span>
18                 </button>
19                 @Html.ActionLink("Application name", "Index", "Home", new { area = "" }, new { @class = "navbar-brand" })
20             </div>
21             <div class="navbar-collapse collapse">
22                 <ul class="nav navbar-nav">
23                     <li>@Html.ActionLink("Home", "Index", "Home")</li>
24                     <li>@Html.ActionLink("About", "About", "Home")</li>
25                     <li>@Html.ActionLink("Contact", "Contact", "Home")</li>
26                 </ul>
27             </div>
28         </div>
29     </div>
30     <div class="container body-content">
31         @RenderBody()
32         <hr />
33         <footer>
34             <p>&copy; @DateTime.Now.Year - My ASP.NET Application</p>

```

Razor – ASP.NET view engine that lets you embed server-based code (C#) into web pages.

All CSS files in the Content folder are rendered.


```

7      @Styles.Render("~/Content/css")
8      @Scripts.Render("~/bundles/modernizr")
9  </head>
10 <body>
11     <div class="navbar navbar-inverse navbar-fixed-top">
12         <div class="container">
13             <div class="navbar-header">
14                 <button type="button" class="navbar-toggle" data-toggle="collapse" data-target=".navbar-collapse">
15                     <span class="icon-bar"></span>
16                     <span class="icon-bar"></span>
17                     <span class="icon-bar"></span>
18                 </button>
19                 @Html.ActionLink("Application name", "Index", "Home", new { area = "" }, new { @class = "navbar-brand" })
20             </div>
21             <div class="navbar-collapse collapse">
22                 <ul class="nav navbar-nav">
23                     <li>@Html.ActionLink("Home", "Index", "Home")</li>
24                     <li>@Html.ActionLink("About", "About", "Home")</li>
25                     <li>@Html.ActionLink("Contact", "Contact", "Home")</li>
26                 </ul>
27             </div>
28         </div>
29     </div>
30     <div class="container body-content">
31         @RenderBody()
32         <hr />
33         <footer>
34             <p>&copy; @DateTime.Now.Year - My ASP.NET Application</p>
35         </footer>
36     </div>
37
38     @Scripts.Render("~/bundles/jquery")
39     @Scripts.Render("~/bundles/bootstrap")
40     @RenderSection("scripts", required: false)
41 </body>
42 </html>
43

```

Html helpers


```

15         <span class="icon-bar"></span>
16         <span class="icon-bar"></span>
17         <span class="icon-bar"></span>
18     </button>
19     @Html.ActionLink("Application name", "Index", "Home", new { area = "" }, new { @class = "navbar-brand"
20 </div>
21 <div class="navbar-collapse collapse">
22     <ul class="nav navbar-nav">
23         <li>@Html.ActionLink("Home", "Index", "Home")</li>
24         <li>@Html.ActionLink("About", "About", "Home")</li>
25         <li>@Html.ActionLink("Contact", "Contact", "Home")</li>
26     </ul>
27 </div>
28 </div>
29 </div>
30 <div class="container body-content">
31     @RenderBody()
32     <hr />
33     <footer>
34         <p>&copy; @DateTime.Now.Year - My ASP.NET Application</p>
35     </footer>
36 </div>
37
38 @Scripts.Render("~/bundles/jquery")
39 @Scripts.Render("~/bundles/bootstrap")
40 @RenderSection("scripts", required: false)
41 </body>
42 </html>
43

```

The views which will be displayed in a placeholder `RenderBody()` are called child views

The default Site.css file

```
1 body {
2     padding-top: 50px;
3     padding-bottom: 20px;
4 }
5
6 /* Set padding to keep content from hitting the edges */
7 .body-content {
8     padding-left: 15px;
9     padding-right: 15px;
10 }
11
12 /* Override the default bootstrap behavior where horizontal description lists
13    will truncate terms that are too long to fit in the left column
14    */
15 .dl-horizontal dt {
16     white-space: normal;
17 }
18
19 /* Set width on the form input elements since they're 100% wide by default */
20 input,
21 select,
22 textarea {
23     max-width: 280px;
24 }
25
```

Default layout and styling

- These are the default layout and styling which work perfectly well.
- You can however, change these files (Layout.cshtml and Site.css)
- You can also add a new Layout file and style sheet (e.g. myLayout.cshtml and mySite.css)
- We are now going to replace these files - the Layout.cshtml and Site.css files.

Changing the Layout.cshtml page

```
<head>
  <meta charset="utf-8" />
  <title>@ViewBag.Title</title>
  <link href="@Url.Content("~/Content/Site.css")" rel="stylesheet" type="text/css" />
  <script src="@Url.Content("~/Scripts/jquery-1.5.1.min.js")"></script>
  <script src="@Url.Content("~/Scripts/modernizr-1.7.min.js")"></script>
</head>
<body>
  <ul id="menu">
    <li>@Html.ActionLink("Home", "Index", "Home")</li>
    <li>@Html.ActionLink("Persons", "Persons", "Home")</li>
    <li>@Html.ActionLink("About", "About", "Home")</li>
  </ul>
  <section id="main">
    @RenderBody()
    <p>Copyright W3schools 2012. All Rights Reserved.</p>
  </section>
</body>
</html>
```

Only the Site.css is linked -
cutting out all the Bootstrap
resources in Content folder

... changes the appearance to ...

- [Home](#)
- [Persons](#)
- [About](#)

ASP.NET

ASP.NET is a free web framework for building great Web sites and Web applications using HTML, CSS and JavaScript.

[Learn more »](#)

Getting started

ASP.NET MVC gives you a powerful, patterns-based way to build dynamic websites that enables a clean separation of concerns and gives you

[Learn more »](#)

Get more libraries

NuGet is a free Visual Studio extension that makes it easy to add, remove, and update libraries and tools in Visual Studio projects.

[Learn more »](#)

Web Hosting

You can easily find a web hosting company that offers the right mix of features and price for your applications.

[Learn more »](#)

Index.cshtml file

```
@{
    ViewBag.Title = "Home Page";
}

<div class="jumbotron">
    <h1>ASP.NET</h1>
    <p class="lead">ASP.NET is a free web framework for building great Web sites and Web applications using HTML, CSS and JavaScript.</p>
    <p><a href="https://asp.net" class="btn btn-primary btn-lg">Learn more &raquo;</a></p>
</div>

<div class="row">
    <div class="col-md-4">
        <h2>Getting started</h2>
        <p>
            ASP.NET MVC gives you a powerful, patterns-based way to build dynamic websites that
            enables a clean separation of concerns and gives you full control over markup
            for enjoyable, agile development.
        </p>
        <p><a class="btn btn-default" href="https://go.microsoft.com/fwlink/?LinkId=301865">Learn more &raquo;</a></p>
    </div>
    <div class="col-md-4">
        <h2>Get more libraries</h2>
        <p>NuGet is a free Visual Studio extension that makes it easy to add, remove, and update libraries and tools in Visual Studio projects.</p>
        <p><a class="btn btn-default" href="https://go.microsoft.com/fwlink/?LinkId=301866">Learn more &raquo;</a></p>
    </div>
    <div class="col-md-4">
        <h2>Web Hosting</h2>
        <p>You can easily find a web hosting company that offers the right mix of features and price for your applications.</p>
        <p><a class="btn btn-default" href="https://go.microsoft.com/fwlink/?LinkId=301867">Learn more &raquo;</a></p>
    </div>
</div>
```

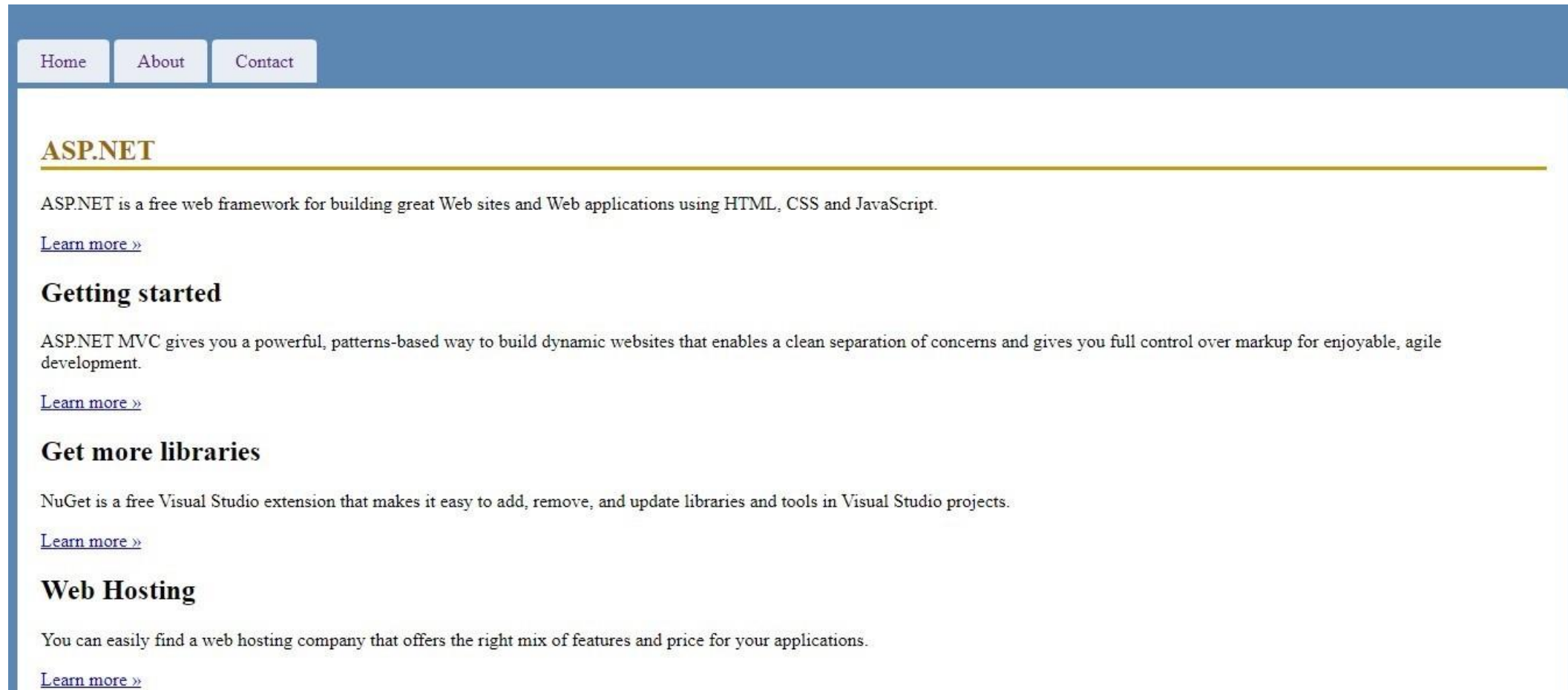
Bootstrap
predefined classes
– not rendered as
we use the
changed layout
page

Let us work on the Site.css file to make it look nicer

```
12
13 h1 {
14     border-bottom: 3px solid #cc9900;
15     font: Georgia, serif;
16     color: #996600;
17 }
18
19
20 .body-content {
21     padding-left: 15px;
22     padding-right: 15px;
23 }
24
25 #menu {
26     padding: 0px;
27     position: relative;
28     margin: 0;
29 }
30
31 #menu li {
```

```
31 #menu li {
32     display: inline;
33 }
34
35 #menu li a {
36     background-color: #e8eef4;
37     padding: 10px 20px;
38     text-decoration: none;
39     line-height: 2.8em;
40     border-radius: 4px 4px 0 0;
41 }
42
43 #menu li a:hover {
44     background-color: #ffffff;
45 }
46
47
48 #main {
49     padding: 20px;
50     background-color: #ffffff;
51     border-radius: 0 4px 4px 4px;
52 }
```

..changes the appearance to... (new styling)





UNIVERSITEIT VAN PRETORIA
UNIVERSITY OF PRETORIA
YUNIBESITHI YA PRETORIA

The CSS Box Model

Make today matter

www.up.ac.za

The CSS Box model

The CSS box model is set of rules that defines how every element in a web page is rendered. According to the CSS box model, **every element in a web page is a rectangular box.**

The content is in the middle, surrounded by optional elements such as padding, border, and margin.

The box model has five main parts/properties that determine the size of the box.

Those

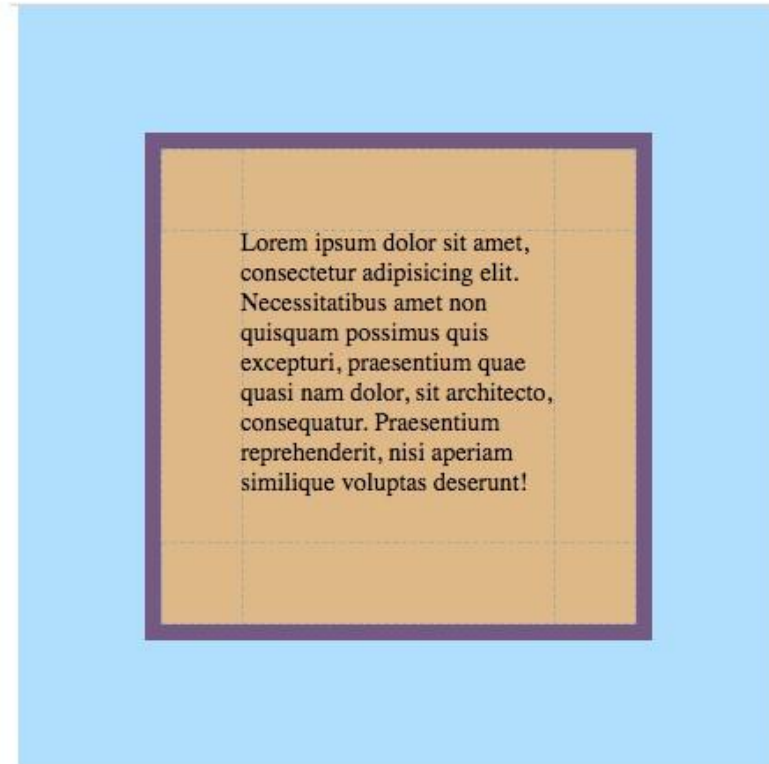
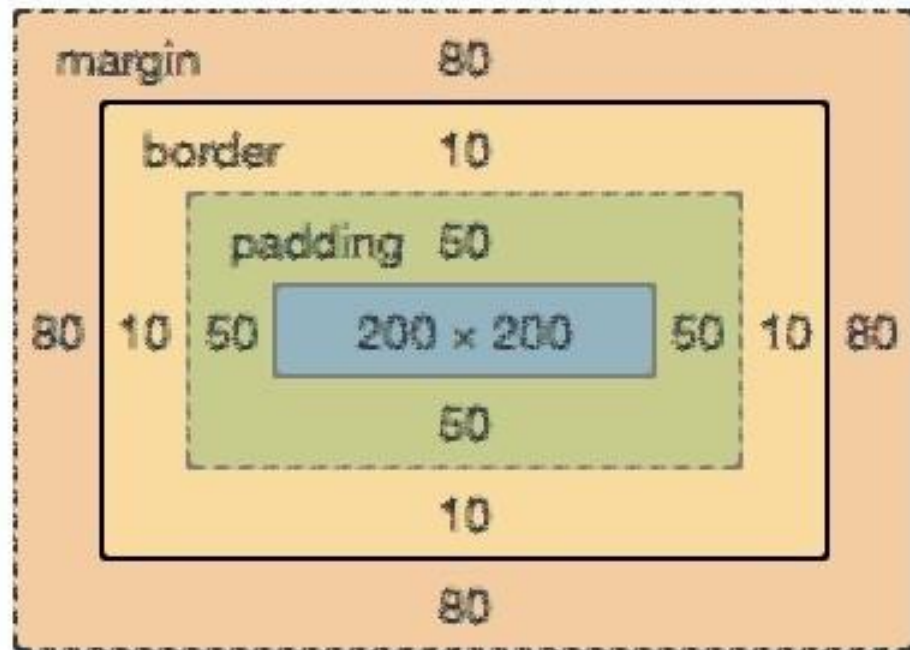
properties **the width, the height, the padding, border and margin** are. Apart from the

width, height, the other properties are optional. Width and height are mandatory whether we set them or not.

The CSS Box model continued....

- Each HTML element is wrapped by a box. Everything in CSS has a box around it, and understanding these boxes is key to being able to create layouts with CSS, or to align items with other items.
- **Content** - The content of the box, where text and images appear.
- **Padding** - Clears an area around the content. The padding is transparent
- **Border** - A border that goes around the padding and content
- **Margin** - Clears an area outside the border. The margin is transparent. Default is 0. It is important to note that margin is the external space separating boxes.

The CSS Box model



Calculation of the dimension of the box

```
p {  
width: 200px;  
padding: 10px;  
margin: 10px;  
border: 1px solid #eee;  
}
```

Total width: $\text{width} + \text{padding-left} + \text{padding-right} + \text{border-right} + \text{border-left} = 222\text{px}$

Box-sizing property

- * {box-sizing: border-box;}

The box-sizing property allows the developer to choose how to calculate the width of an element. There are three values associated to box-sizing: content-box, padding-box, and border-box.

Border-box: This value brings more flexibility and makes the calculation of the element width and height more intuitive. The width or height of an element includes padding and border.

If you set an element's width to 100 pixels, that 100 pixels will include any border or padding you added, and the content box will shrink to absorb that extra width. This typically makes it much easier to size elements. box-sizing: border-box is the default styling that browsers use for the [`<table>`](#), [`<select>`](#), and [`<button>`](#) elements, and for [`<input>`](#) elements whose type is [`radio`](#), [`checkbox`](#), [`reset`](#), [`button`](#), [`submit`](#), [`color`](#), or [`search`](#).

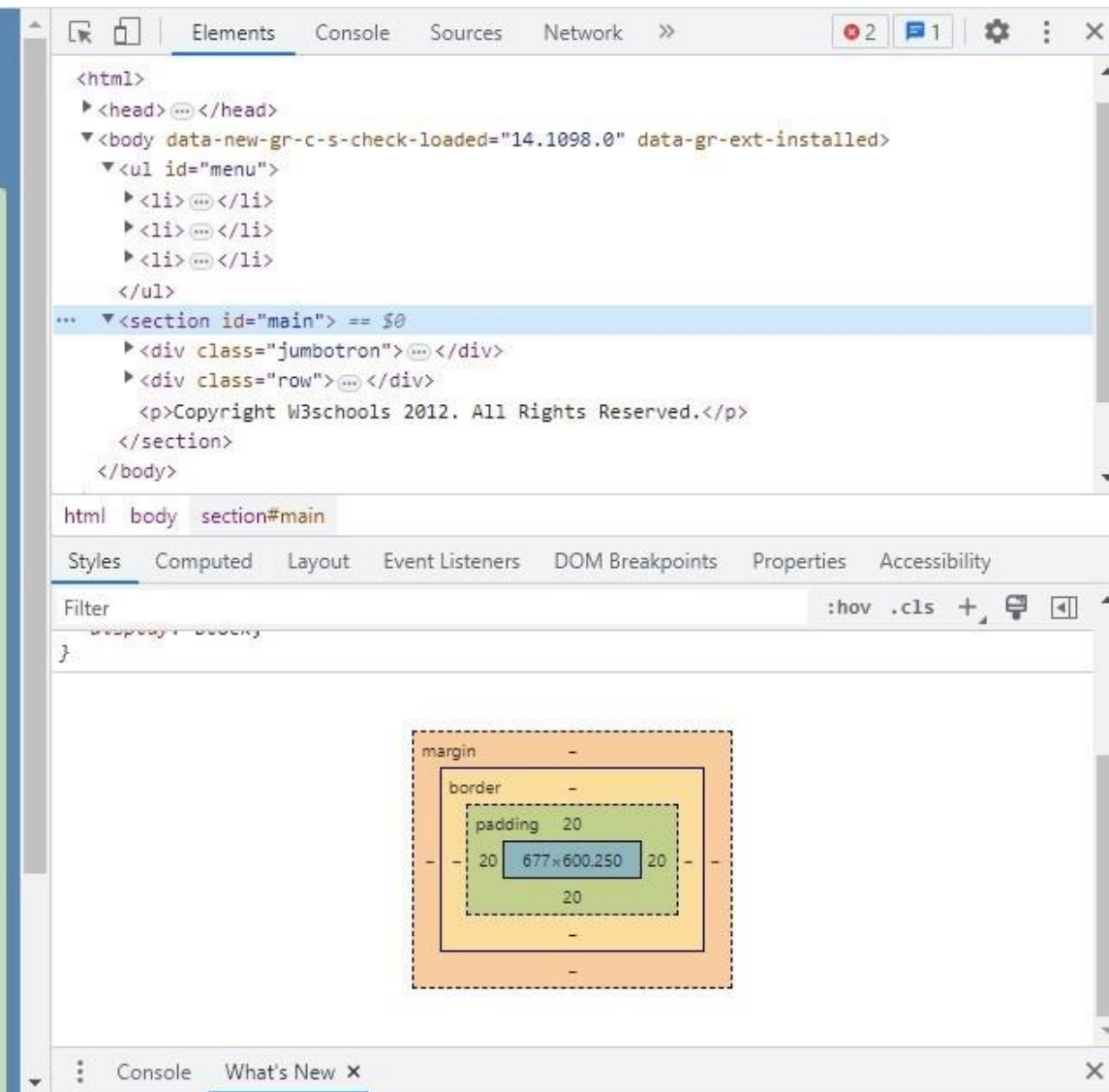
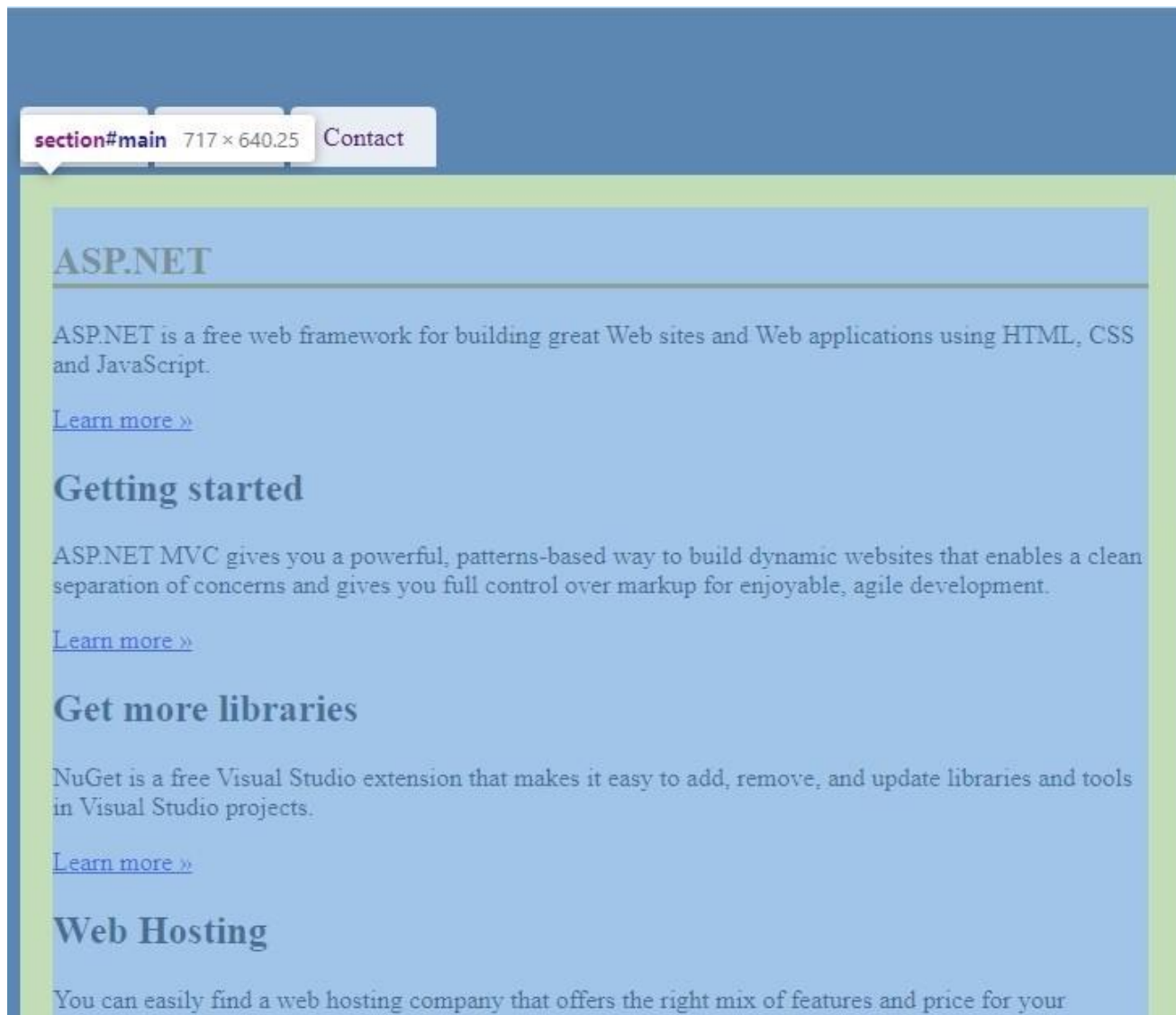
When content overflows

- By default, an element takes as much height as its content fits if we do not declare a specific height. Things get trickier when we set a specific height for an element. What happens in the specific case when the content overflow? Meaning when the height of the content is larger than the specified height of the element.

Overflow:visible/scroll/hidden/auto

- Default: visible
- Overflow:scroll – will provide scroll bar.
- Overflow:auto – will provide scroll bar if necessary

Inspect the webpage





UNIVERSITEIT VAN PRETORIA
UNIVERSITY OF PRETORIA
YUNIBESITHI YA PRETORIA

CSS Display property

Make today matter

www.up.ac.za

CSS Display options – inline

- Inline boxes are laid out , from left to right and have generally no width or height applied to them. Unlike other types of boxes, inline boxes do not fill hundred percent of their parent width. It's width is the width of the content of its element (text for example).
- Also, two or more inline elements will stick one after each other, without a line break. Well-known inline CSS properties include <a>, , , , <q>, <textarea>, etc.

Display: inline

If a box has a display type of `inline`, then:

- The box will not break onto a new line.
- The `width` and `height` properties will not apply.
- Vertical (top and bottom) padding, margins, and borders will apply but will not cause other inline boxes to move away from the box.
- Horizontal padding (left and right), margins, and borders will apply and will cause other inline boxes to move away from the box. The `<a>` element, used for links, ``, `` and `` are all examples of elements that will display inline by default.

https://developer.mozilla.org/en-US/docs/Learn/CSS/CSS_layout/

```

12
13 h1 {
14     border-bottom: 3px solid #cc9900;
15     font: Georgia, serif;
16     color: #996600;
17 }
18
19
20 .body-content {
21     padding-left: 15px;
22     padding-right: 15px;
23 }
24
25 #menu {
26     padding: 0px;
27     position: relative;
28     margin: 0;
29 }
30
31 #menu li {

```

```

31 #menu li {
32     display: inline;
33 }
34
35 #menu li a {
36     background-color: #e8eef4;
37     padding: 10px 20px;
38     text-decoration: none;
39     line-height: 2.8em;
40     border-radius: 4px 4px 0 0;
41 }
42
43 #menu li a:hover {
44     background-color: #ffffff;
45 }
46
47
48 #main {
49     padding: 20px;
50     background-color: #ffffff;
51     border-radius: 0 4px 4px 4px;
52 }

```

[Home](#)[About](#)[Contact](#)

ASP.NET

ASP.NET is a free web framework for building great Web sites and Web applications using HTML, CSS and JavaScript.

[Learn more »](#)

Getting started

ASP.NET MVC gives you a powerful, patterns-based way to build dynamic websites that enables a clean separation of concerns and gives you full control over markup for enjoyable, agile development.

[Learn more »](#)

Get more libraries

NuGet is a free Visual Studio extension that makes it easy to add, remove, and update libraries and tools in Visual Studio projects.

[Learn more »](#)

Web Hosting

You can easily find a web hosting company that offers the right mix of features and price for your

The screenshot displays the ASP.NET website in a browser with the developer tools open. The **Elements** panel shows the HTML structure, including the `<ul id="menu">` and `` elements. The **Styles** panel shows the CSS rules for the `li` element, with `display: inline;` highlighted by a red circle. The `li` element is selected in the breadcrumb trail: `html > body > ul#menu > li`.

```
<!DOCTYPE html>
<html>
  <head> ... </head>
  <body data-new-gr-c-s-check-loaded="14.1098.0" data-gr-ext-installed>
    <ul id="menu">
      <li> ... </li>
      <li> ... </li>
      <li> ... </li>
    </ul>
    <section id="main">
      <div class="jumbotron"> ... </div>
      <div class="row"> ... </div>
      <p>Copyright W3schools 2012. All Rights Reserved.</p>
    </section>
  </body>
</html>
```

Styles panel for `li`:

```
element.style {
}
#menu li {
  display: inline;
}
li {
  display: list-item;
  text-align: -webkit-match-parent;
}
Inherited from ul#menu
ul {
  list-style-type: disc;
}
Inherited from user agent stylesheet
```

[Home](#)[About](#)[Contact](#)

ASP.NET

ASP.NET is a free web framework for building great Web sites and Web applications using HTML, CSS and JavaScript.

[Learn more »](#)

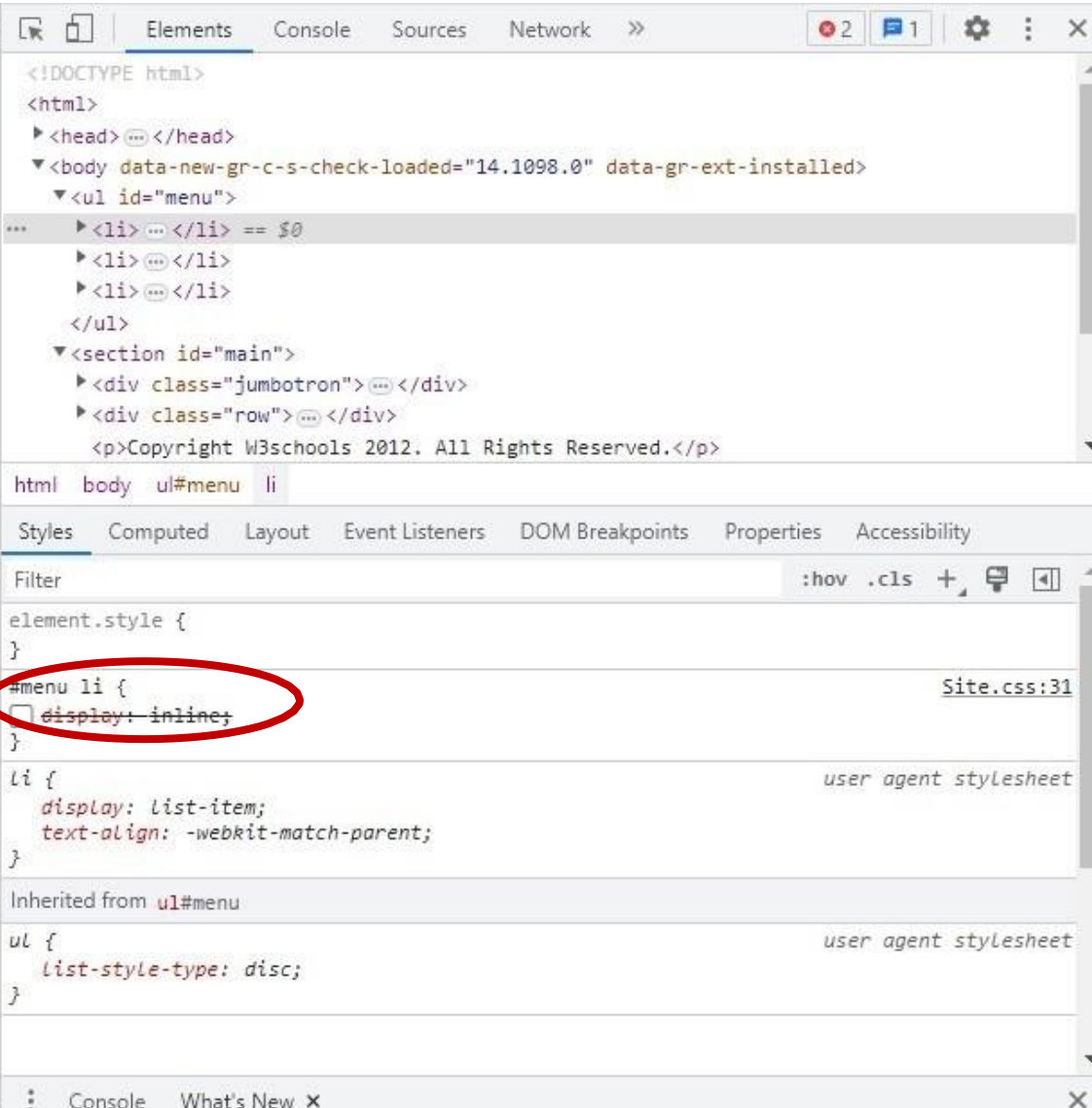
Getting started

ASP.NET MVC gives you a powerful, patterns-based way to build dynamic websites that enables a clean separation of concerns and gives you full control over markup for enjoyable, agile development.

[Learn more »](#)

Get more libraries

NuGet is a free Visual Studio extension that makes it easy to add, remove, and update libraries and tools in Visual Studio projects.



Inline replaced elements

- A replaced element is an inline element with an intrinsic width and height. Meaning that the width and height are not defined with CSS, but belongs to the element. For example, the initial width and height of an image is set outside of CSS.
- The content of a replaced element is not in the document, it is generally a link to an external content. It is the case of the image tag which links to an external file.
- In absence of the CSS loading, they will load their default width and height.
- e.g. ``, `<input>`

CSS display options: block

- Contrary to inline elements, block level elements act as if there is a line break after them so that they stack up one on another like actual boxes. Yet, there is a slight difference with actual boxes. ***Block level element occupy their space so that even if their width is less than the width of their parent container, by default, no other element will occupy the remaining vertical space.*** So they can have specific width and height, and padding, margin, and border on all four sides.
- <p>, <form>, <table>, <h>, <div>,....

Display: block

If a box is defined as a block, it will behave in the following ways:

- The box will break onto a new line.
- The box will extend in the inline direction to fill the space available in its container.
- In most cases this means that the box will become as wide as its container, filling up 100% of the space available.
- The width and height properties are respected.
- Padding, margin and border will cause other elements to be pushed away from the box

Unless we decide to change the display type to inline, elements such as headings

(e.g. `<h1>`) and `<p>` and `<form>`, `<table>`, `<div>` all use `block` as their outer display type by default.

[Home](#)[About](#)[Contact](#)

ASP.NET

ASP.NET is a free web framework for building great Web sites and Web applications using HTML, CSS and JavaScript.

[Learn more »](#)

Getting started

ASP.NET MVC gives you a powerful, patterns-based way to build dynamic websites that enables a clean separation of concerns and gives you full control over markup for enjoyable, agile development.

h2 677 × 27

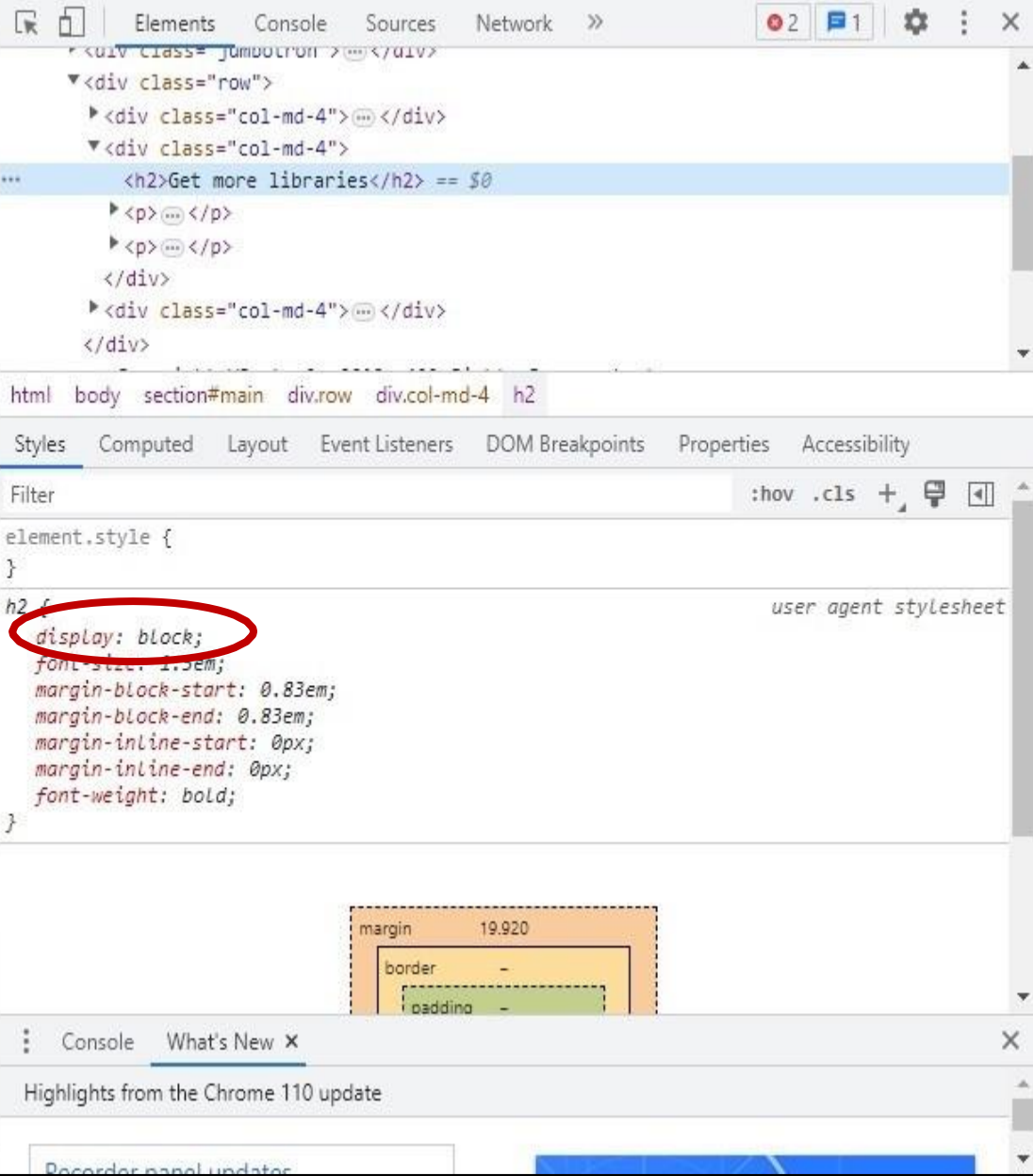
Get more libraries

NuGet is a free Visual Studio extension that makes it easy to add, remove, and update libraries and tools in Visual Studio projects.

[Learn more »](#)

Web Hosting

You can easily find a web hosting company that offers the right mix of features and price for your



Display: inline-block

Compared to `display: inline`, the major difference is that `display: inline-block` allows to set a width and height on the element.

Also, with `display: inline-block`, the top and bottom margins/paddings are respected, but with `display: inline` they are not.

Compared to `display: block`, the major difference is that `display: inline-block` does not add a line-break after the element, so the element can sit next to other elements.



UNIVERSITEIT VAN PRETORIA
UNIVERSITY OF PRETORIA
YUNIBESITHI YA PRETORIA

CSS Position property

Make today matter

www.up.ac.za

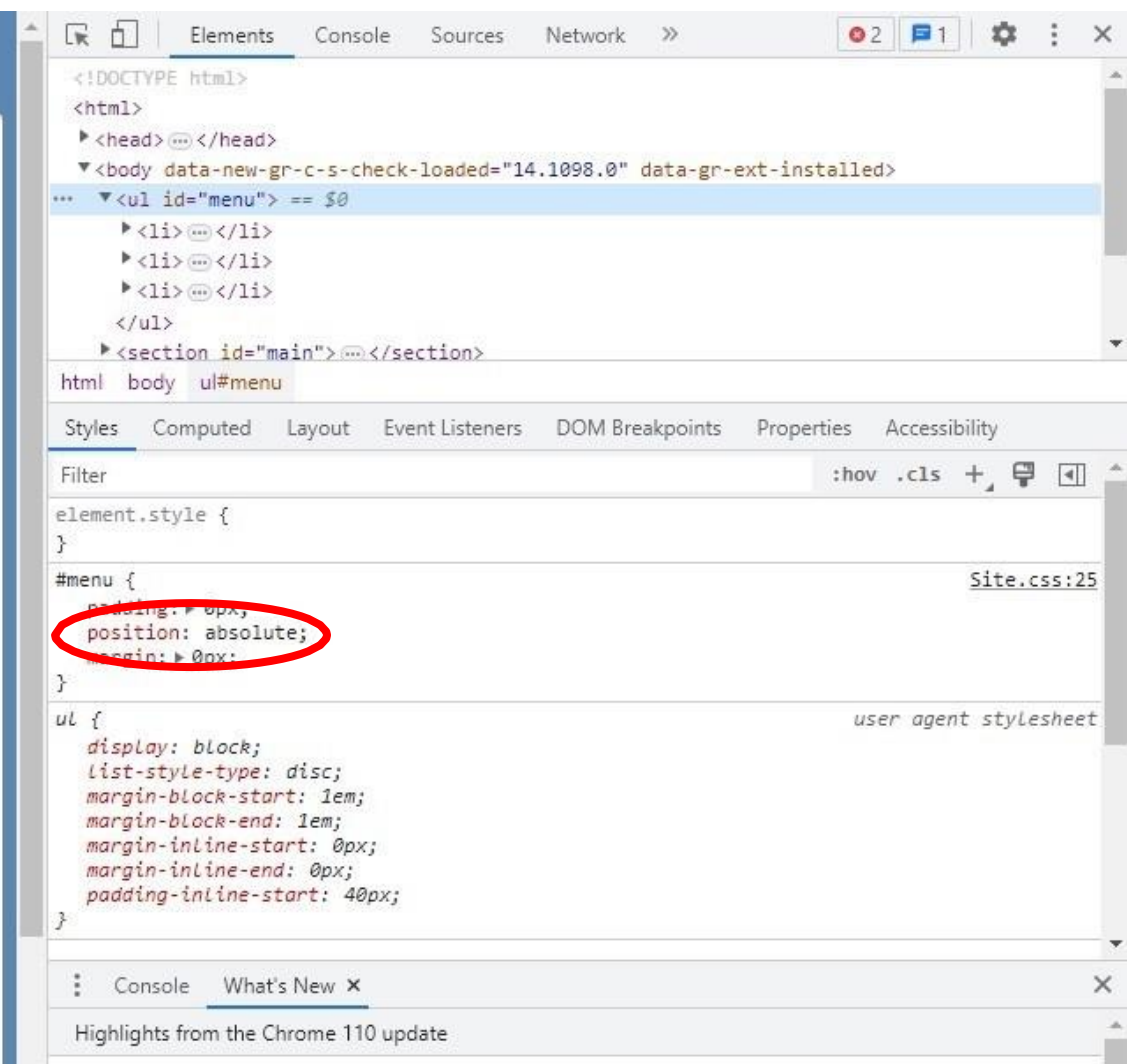
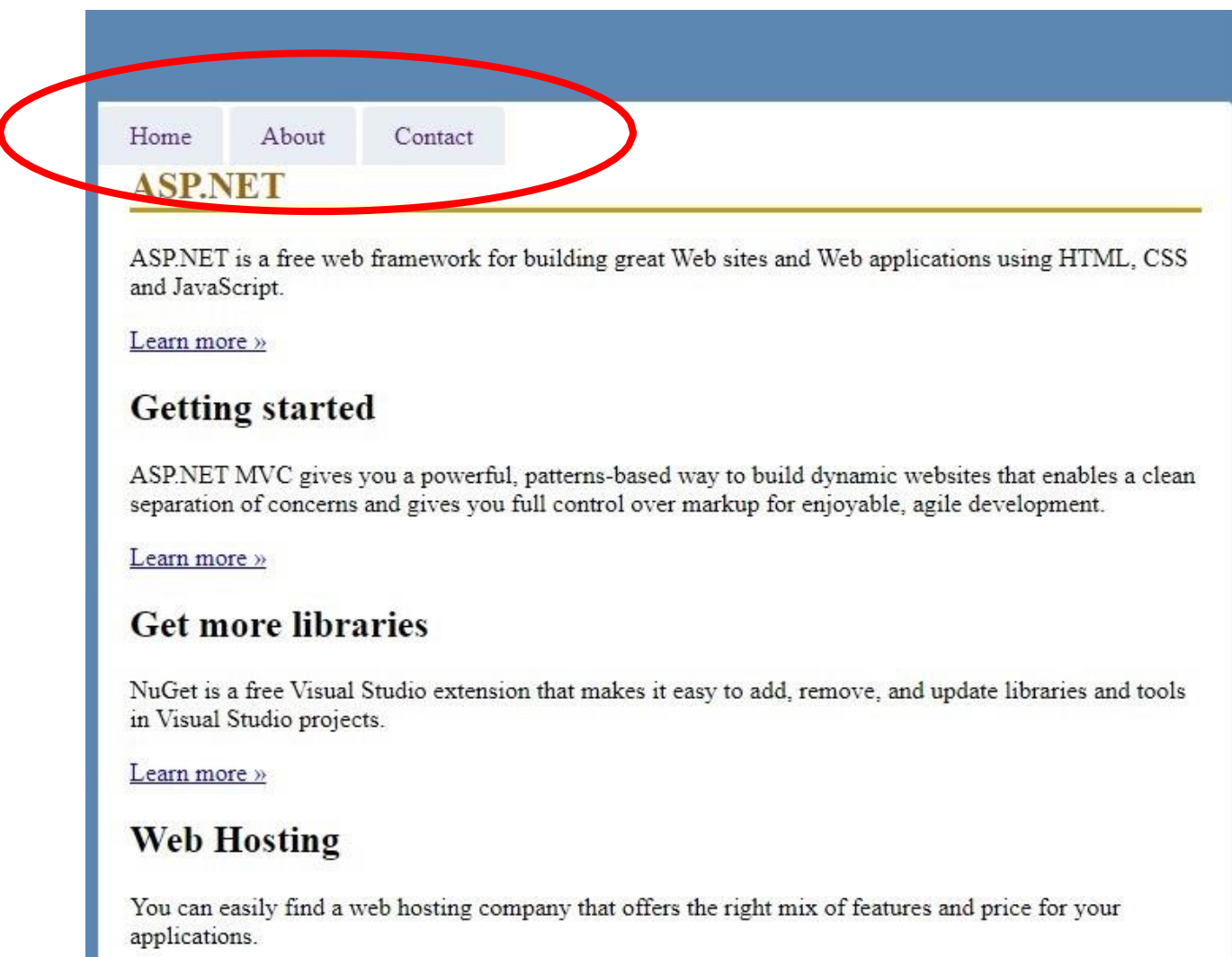
CSS position property

- The CSS position property defines the position of an element in a document. This property works with the left, right, top, bottom and z-index properties to determine the final position of an element on a page
- Positioning allows you to take elements out of the normal document layout flow, and make them behave differently, for example sitting on top of one another, or always remaining in the same place inside the browser viewport.
- https://developer.mozilla.org/en-US/docs/Learn/CSS/CSS_layout/Positioning

Position property

- Position: static
 - Has no effect on the position of the element. It will appear in the normal flow of the page. Top, bottom, left, right properties can be set but will have no effect.
- Position: relative
 - Elements with `position: relative` remain in the normal flow of the document. But, unlike static elements, the left, right, top, bottom and z-index properties affect the position of the element. An offset, based on the values of left, right, top and bottom properties, is applied to the element **relative to itself**.
- Position: absolute
 - Elements with `position: absolute` are positioned relative to their parent elements (its nearest positioned ancestor – ancestor with position value other than static). In this case, the element is removed from the normal document flow. The other elements will behave as if that element is not in the document. No space is created for the element in the page layout. The values of left, top, bottom and right determine the final position of the element. If there is no positioned ancestor element, it is positioned relative to the `<html>` element.
- Position: fixed

- Fixed position elements are similar to absolutely positioned elements. They are also removed from the normal flow of the document. But unlike absolutely positioned element, they are always positioned relative to the <html> element. One thing to note is that fixed elements are not affected by scrolling. They always stay in the same position on the screen.



More positioning

- Z-index – This property specifies the stack order of an element (e.g. z-index: -1)
- The float Property is used for positioning and formatting content, e.g. let an image float left to the text in a container. In its simplest use, the float property can be used to wrap text around images (e.g. float: right)
- https://www.w3schools.com/css/css_float.asp

Pseudo classes

A pseudo-class is used to define a special state of an element. For example, it can be used to:

Style an element when a user mouses over it

Style visited and unvisited links differently

Style an element when it gets focus

In this example, the pseudo class is `:hover`

There are MANY more

```
#menu li a {  
    background-color: #e8eef4;  
    padding: 10px 20px;  
    text-decoration: none;  
    line-height: 2.8em;  
    border-radius: 4px 4px 0 0;  
}  
  
#menu li a:hover {  
    background-color: #ffffff;  
}
```



UNIVERSITEIT VAN PRETORIA
UNIVERSITY OF PRETORIA
YUNIBESITHI YA PRETORIA

Forms

Make today matter

www.up.ac.za

Forms

```
<form action = "Script URL" method = "GET|POST">  
    form elements like input, textarea etc.  
</form>
```

- https://www.tutorialspoint.com/html/html_forms.htm

Forms – in Views

I replaced the Contact View Content with this:

```
@{
    ViewBag.Title = "Contact";
}
<h4 style="color:purple">
    <b>First Name:</b> @ViewBag.FirstName <br />
    <b>Last Name:</b> @ViewBag.LastName <br />
    <b>Age:</b> @ViewBag.Age <br />
    <b>Is Alive:</b> @ViewBag.IsAlive
</h4>
<hr />
<h3><b>Forms: Weakly Typed</b></h3>

<form action="form1" method="post">

    <label for="fname">First name:</label><br>
    <input type="text" id="fname" name="fname" value="John"><br>

    <label for="lname">Last name:</label><br>
    <input type="text" id="lname" name="lname" value="Doe"><br><br>

    <label for="age">Age:</label><br>
    <input type="number" id="age" name="age"><br><br>

    <label for="isAlive">Is Alive:</label><br>
    <input type="checkbox" name="isAlive" /><br >

    <input type="submit" value="Submit Form" />

</form>
```



```
<h4 style="color:purple">
  <b>First Name:</b> @ViewBag.FirstName <br />
  <b>Last Name:</b> @ViewBag.LastName <br />
  <b>Age:</b> @ViewBag.Age <br />
  <b>Is Alive:</b> @ViewBag.IsAlive
</h4>
<hr />
<h3><b>Forms: Weakly Typed</b></h3>
```

```
<form action="form1" method="post">

  <label for="fname">First name:</label><br>
  <input type="text" id="fname" name="fname" value="John"><br>

  <label for="lname">Last name:</label><br>
  <input type="text" id="lname" name="lname" value="Doe"><br><br>

  <label for="age">Age:</label><br>
  <input type="number" id="age" name="age"><br><br>

  <label for="isAlive">Is Alive:</label><br>
  <input type="checkbox" name="isAlive" /><br >

  <input type="submit" value="Submit Form" />

</form>
```

```
@{
    ViewBag.Title = "Contact";
}
<h4 style="color:purple">
    <b>First Name:</b> @ViewBag.FirstName <br />
    <b>Last Name:</b> @ViewBag.LastName <br />
    <b>Age:</b> @ViewBag.Age <br />
    <b>Is Alive:</b> @ViewBag.IsAlive
</h4>
<hr />
<h3><b>Forms: Weakly Typed</b></h3>
```

```
<form action="form1" method="post">

    <label for="fname">First name:</label><br>
    <input type="text" id="fname" name="fname" value="John"><br>

    <label for="lname">Last name:</label><br>
    <input type="text" id="lname" name="lname" value="Doe"><br><br>
```

form1 is Action Method that gets executed when forms sends data to **HomeController** using post method

In the HomeController...

form1 Action method,

- 1) receives these parameters,
- 2) assigns it to ViewBag objects
- 3) returns Contact View to
- 4) display these values.

The ViewBag in ASP.NET MVC is used to transfer temporary data (which is not included in the model) from the controller to the view.

References

```
public ActionResult form1(string fName, string lName, int age, string isAlive)
{
    ViewBag.FirstName = fName;
    ViewBag.LastName = lName;
    ViewBag.Age = age;

    if (isAlive != null)
    {
        ViewBag.IsAlive = "Alive";
    }
    else
    {
        ViewBag.IsAlive = "Not Alive";
    }

    return View("Contact");
}
```

When rendered...

First Name:
Last Name:
Age:
Is Alive:

Forms: Weakly Typed

First name:

Last name:

Age:

Is Alive:



First Name: Mary
Last Name: Alexander
Age: 20
Is Alive: Alive

Forms: Weakly Typed

First name:

Last name:

Age:

Is Alive:



Let us reflect....

- In this example, each item was sent as a parameter.
- In MVC, data is conceptualized as models.

```
namespace INF272Lecture4v1.Models
{
    References
    public class PersonModel
    {
        References
        public string FirstName { get; set; }
        References
        public string LastName { get; set; }
        References
        public int Age { get; set; } = 0;

        References
        public bool IsAlive { get; set; } = true;
    }
}
```


Forms – sending objects

- With this new form, we will send *objects* instead of sending each item as parameter.

```
@model INF272Lecture4v1.Models.PersonModel
@{
    ViewBag.Title = "Contact";
}
<h4 style="color:purple">
    <b>First Name:</b> @ViewBag.FirstName <br />
    <b>Last Name:</b> @ViewBag.LastName <br />
    <b>Age:</b> @ViewBag.Age <br />
    <b>Is Alive:</b> @ViewBag.IsAlive
</h4>
<hr />
<h3><b>Forms: Strongly Typed</b></h3>

@using (Html.BeginForm("Form2", "Home", FormMethod.Post))
{
    @Html.LabelFor(m => m.FirstName)
    @Html.EditorFor(m => m.FirstName)

    @Html.LabelFor(m => m.LastName)
    @Html.TextBoxFor(m => m.LastName)

    @Html.LabelFor(m => m.Age)
    @Html.EditorFor(m => m.Age)

    @Html.LabelFor(m => m.IsAlive)|
    @Html.CheckBoxFor(m => m.IsAlive) <br />

    <input type="submit" value="Submit Form" />
}
}
```

Contact View

```
@model INF272Lecture4v1.Models.PersonModel
@{
    ViewBag.Title = "Contact";
}
```

```
<h4 style="color:purple">
    <b>First Name:</b> @ViewBag.FirstName <br />
    <b>Last Name:</b> @ViewBag.LastName <br />
    <b>Age:</b> @ViewBag.Age <br />
    <b>Is Alive:</b> @ViewBag.IsAlive
</h4>
<hr />
<h3><b>Forms: Strongly Typed</b></h3>
```

Accessing the
PersonModel

```
@using (Html.BeginForm("Form2", "Home", FormMethod.Post))
```

```
{
    @Html.LabelFor(m => m.FirstName)
    @Html.EditorFor(m => m.FirstName)
```

```
    @Html.LabelFor(m => m.LastName)
    @Html.TextBoxFor(m => m.LastName)
```

```
    @Html.LabelFor(m => m.Age)
    @Html.EditorFor(m => m.Age)
```

```
    @Html.LabelFor(m => m.IsAlive)
    @Html.CheckBoxFor(m => m.IsAlive) <br />
```

```
    <input type="submit" value="Submit Form" />
}
```

i. **Form2**: It is Action Method Name

ii. **Home**: It is Controller Name

iii. **FormMethod.Post**: It denotes that all the data will be submitted to controller using Post method.

HTML
helpers

In the HomeController.cs

```
[HttpPost]
```

0 references

```
public ActionResult form2(models.PersonModel pm)
{
    ViewBag.FirstName = pm.FirstName;
    ViewBag.LastName = pm.LastName;
    ViewBag.Age = pm.Age;

    if (pm.IsAlive == true)
        ViewBag.IsAlive = "Alive";
    else
        ViewBag.IsAlive = "Not Alive";

    return View("Contact");
}
```

When rendered....

First Name: Mary
Last Name: Alexander
Age: 2
Is Alive: Alive

Forms: Strongly Typed

First Name

Last Name

Current Age

Living Status



The layout of the form also improved after adding this CSS

```
form {  
  padding-left: 12px;  
}  
  
form label {  
  display: block;  
  padding: 4px;  
}  
  
input[type="text"], input[type="number"] {  
  width: 300px;  
}  
  
input[type="submit"] {  
  padding: 4px;  
}
```

HTML helpers

The `HtmlHelper` class renders HTML controls in the razor view. It binds the model object to HTML controls to display the value of model properties into those controls and also assigns the value of the controls to the model properties while submitting a web form. So always use the `HtmlHelper` class in razor view instead of writing HTML tags manually.

See more: <https://www.tutorialsteacher.com/mvc/html-helpers>



UNIVERSITEIT VAN PRETORIA
UNIVERSITY OF PRETORIA
YUNIBESITHI YA PRETORIA

Practical overview

Make today matter

www.up.ac.za

This is a simple brief overview of the practical work that will be completed in the practical sessions this week.

Main tasks

- Use practical 1 to work from.
- Change the model by adding another property named myLink
- Add 5 HTML pages to the project containing the person info from every person in your study group including yourself.
- The View which accesses the List of people contains Links to the personal pages

List of People

Student number	First Name	Last Name	Email address	Link to personal page
u0001	Name 1	Surname 1	person1@tuks.co.za	LINK
u0002	Name 2	Surname 2	person2@tuks.co.za	LINK
u0003	Name 3	Surname 3	person3@tuks.co.za	LINK
u0004	Name 4	Surname 4	person4@tuks.co.za	LINK
u0005	Name 5	Surname 5	person5@tuks.co.za	LINK

© 2023 - INF272 Samples

- The personal pages each have 8 menu items as follows.



- The personal pages have the following content.

MAIN	LIST	Person 01	Person 02	Person 03	Person 04	Person 05	BACK
------	------	-----------	-----------	-----------	-----------	-----------	------

Biography and Background

In this paragraph you have to give the following information about the person in your study group: 1) From where, 2) the degree he/she studies and 3) the reason why he/she studies that degree. For example: Mary Alexander was born in Polokwane, and went to school in Haenertsburg. She studies BIT(I)S with Statistics as other elective because she believes Data Science is the future. Add a photograph of your friend. Make sure that the photo is situated at the left of the text and does not flow into the hobbies section.

Hobbies

In this paragraph you have to give a description of the hobbies of your friend. Insert a photo related to the hobby(ies) of your friend. The photo must be situated at the right of the text.

Modules

In this section you have to provide the first and second semester modules your friend is enrolled for - in two columns.

Modules Semester 1

Module 11
Module 12
Module xx


Modules Semester 2

Module 21
Module 22
Module yy

When the user hovers over the image, the mirror-image is displayed

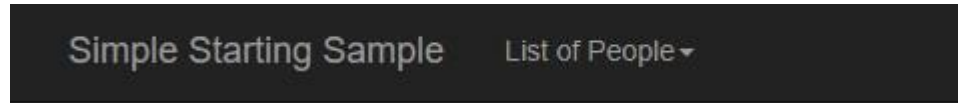
Biography and Background

In this paragraph you have to give the following information about the person in your study group: 1) From where, 2) the degree he/she studies and 3) the reason why he/she studies that degree. For example: Mary Alexander was born in Polokwane, and went to school in Haenertsburg. She studies BIT(IS) with Statistics as other elective because she believes Data Science is the future. Add a photograph of your friend. Make sure that the photo is situated at the left of the text and does not flow into the hobbies section.



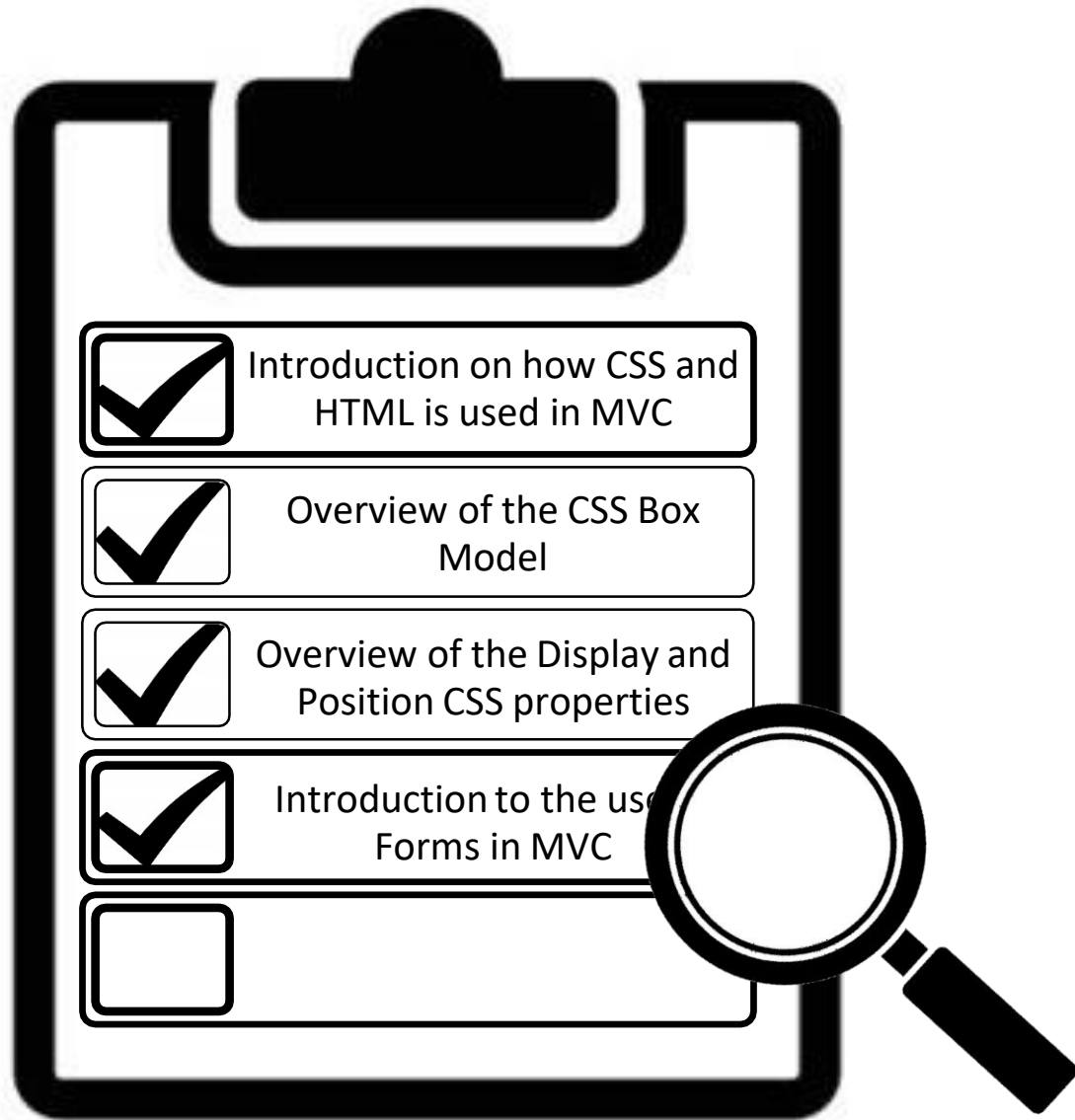
Main tasks

- The main menu has two items.



- The List of People item has sub-items.





ANY
Questions?

Good Bye

Until next time

