

HTML and CSS

Labs and Solutions

LAB NOTES.....	2
MODULE 1 LAB – HTML.....	3
MODULE 1 LAB SOLUTION	9
MODULE 2 LAB – CORE HTML ELEMENTS.....	11
MODULE 2 LAB SOLUTION	19
MODULE 3 LAB – CASCADING STYLE SHEETS	21
MODULE 3 LAB SOLUTION	31
MODULE 4 LAB – FONTS AND TEXT	34
MODULE 4 LAB SOLUTION	39
MODULE 5 LAB – COLOR AND BACKGROUNDS	41
MODULE 5 LAB SOLUTION	46
MODULE 6 LAB – ANCHORS AND HYPERLINKS	48
MODULE 6 LAB SOLUTION	51
MODULE 7 LAB – PAGE LAYOUT	53
MODULE 7 LAB SOLUTION	57
MODULE 8 LAB – IMAGES.....	58
MODULE 8 LAB SOLUTION	63
MODULE 9 LAB – HTML FORMS	67
MODULE 9 LAB SOLUTION	70
MODULE 10 MULTIMEDIA	72
MODULE 10 LAB SOLUTION	76

COPYRIGHT

© 2017 MAX Technical Training, Inc. All Rights Reserved. This manual and any training materials supplied with it are copyrighted with all rights reserved. No part of this publication may be reproduced, transmitted, transcribed, stored in a retrieval system or translated into any language in any form by any means without the written permission of MAX Technical Training, Inc.

TRADEMARKS

Brand names, company names and product names used herein are trademarks or registered trademarks of their respective companies.

MAX Technical Training, Inc.
4900 Parkway Drive, Suite 160
Mason, OH 45040
513.322.8888

MA-2001 A0 03/27/17

Lab Notes

Try each lab from the basic steps first, and then review the lab solutions if you need some help.

Your Environment Might Not Look the Same

The screens and features depicted in this courseware and in the lab environment may differ due to:

- Drive letters or network paths used for lab files.
- Version of the editor being used.
- Browser versions.

Your instructor should provide you with the following, if needed:

URL to your server: _____

User Name: _____

Password: _____

Path to sample files: _____

Module 1 Lab – HTML

Questions

- ☐ Which is the latest version of HTML?
 - ☐ HTML 4.01
 - ☐ HTML 5
 - ☐ HTML 5.1
 - ☐ XHTML

- ☐ Which of the following browsers support HTML5 and CSS3?
 - ☐ Microsoft Edge
 - ☐ Internet Explorer 6 and 7
 - ☐ Internet Explorer 8 and 9
 - ☐ Internet Explorer 10 and 11
 - ☐ Google Chrome
 - ☐ Mozilla Firefox
 - ☐ Apple Safari

- ☐ HTML stands for:
 - ☐ Hypertext Meta Language
 - ☐ Harvard Text Markup Language
 - ☐ Hypertext Markup Language
 - ☐ Hot Typesetting Markup Language

- ☐ CSS Stands for:
 - ☐ Cyclic Style Sheets
 - ☐ Cascading Style Sheets
 - ☐ Common Style System
 - ☐ Cascading Scripting Styles

- ☐ Shims and Shivs:
 - ☐ Add new features to web browsers.
 - ☐ Add new features to HTML and CSS.
 - ☐ Supply missing functionality to older browsers without impacting newer browsers.
 - ☐ Make a CSS or HTML feature in all recent browsers behave exactly the same way.

- ☐ Polyfills:
 - ☐ Add new features to web browsers.
 - ☐ Add new features to HTML and CSS.


- ☐ Supply missing functionality to older browsers without impacting newer browsers.
 - ☐ Make a CSS or HTML feature in all recent browsers behave exactly the same way.
-
- ☐ Which of the following URLs are not valid?
 - ☐ http://www.example.com
 - ☐ http://example.com
 - ☐ http://example.com/home.html
 - ☐ http://www.example.com:81
 - ☐ http://www.example.com.81
 - ☐ http://www.example.com/81
 - ☐ httpx://www.example.com
 - ☐ https://www.example.com
 - ☐ http://www.example.com.homepage.html
 - ☐ www://example.com
-
- ☐ Which of the following HTTP Status Codes is for a missing page?
 - ☐ 200
 - ☐ 400
 - ☐ 401
 - ☐ 403.20
 - ☐ 404
 - ☐ 500
 - ☐ 504

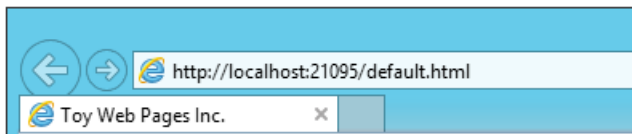
Hands on Lab

Exercise 1 – Create a web page.

This is your chance to experiment with your lab environment and HTML editors. In this lab you will create a new site and add a page.

- ☐ On your lab PC or virtual machine, launch Visual Studio.
- ☐ In Visual Studio click the File menu, New and Web Site.
- ☐ Even though we will not be writing any C# code, select “Visual C#” from the Templates folder.
- ☐ Click ASP.NET Empty Web Site.
- ☐ At the bottom of the dialog box click the “**Web Location**” dropdown and note that Visual Studio can edit files on your local file system (C:), directly to a web server using HTTP (a server with Front Page Extensions) or directly to a web server using FTP (File Transfer Protocol – ftp://www.example.com).
Leave File System selected.
- ☐ At the bottom of the dialog box delete the default path and enter:
C:\HTMLCSS\Lab1\website
(Ask your instructor if you should be using another location.)
Tip: If you type a path with non-existent folders, Visual Studio will create them for you!
- ☐ Click OK to create your new web site project.

- ☐ You now have an (almost) empty web site. As Visual Studio's templates are designed for ASP.NET projects it has added two files to the project: web.config and web.debug.config.
- ☐ Right-click Web.config, click Delete and OK.
- ☐ We need a new web page... In the Visual Studio menus click Website and Add New Item.
- ☐ Click HTML Page and at the bottom of the dialog box change the name to "Home.html".
- ☐ Click Add.
You now have a bare minimum web page!
- ☐ Just below the <body> tag type:
This is my first web page!
- ☐ While saving your files before doing anything else is always a good idea, Visual Studio will automatically save the file when you perform the next step. If you are the cautious type... go ahead and click the save button in the ribbon or click File and then Save home.html.
There are also two save buttons in the ribbon: . The first will save the current file and the second will save all open files.
- ☐ In the menu bar, click File and then click View in Browser (Internet Explorer). Or, right-click the file's name in the Solution Explorer pane and click View in Browser (Internet Explorer). Or, right-click any blank space around your HTML code and click View in Browser (Internet Explorer).
- ☐ You have a web page!
You could now upload this file to a public web server and share it with the world!
- ☐ Notice the URL in the address bar. It is not a path to a local file (C:\...) and it does not look like a real web server.

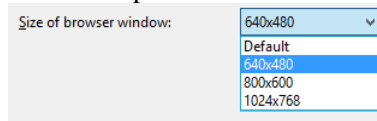


When you launch a page from Visual Studio it launches a local web server called IIS Express. It adds a random port number for your application.

- ☐ In the browser's menu (if the menu is not displayed, tap the ALT key) click View and the Source. You will see the exact same HTML you saw in Visual Studio.
- ☐ Close the window displaying the HTML source.
- ☐ Close Internet Explorer.
- ☐ If you have multiple browsers installed on your PC or Virtual Machine... Open another browser (Firefox, Chrome, etc.) and in the address bar type the path to your page. Something like:
C:\HTMLCSS\Lab1\WebSite\home.html. You should test new pages in as many browsers as possible!
- ☐ Close the browser.
- ☐ In the menu bar, click File and then click Browse with. Or, right-click the file's name in the Solution Explorer pane and click Browse with.

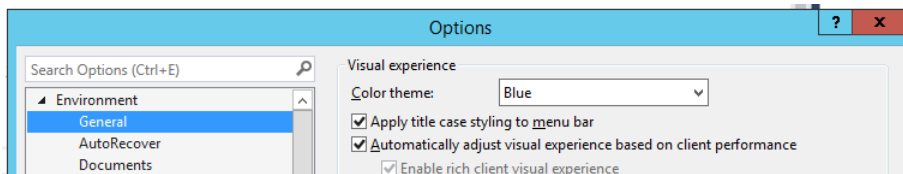
If you have multiple browsers installed, you can pick them here or pick one to be default for Visual Studio.

You can also pick a size for the browser window.

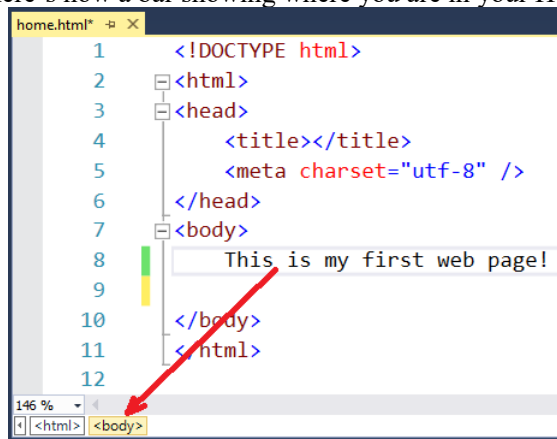


Exercise 2 – Explore Visual Studio Options

- ☐ Return to Visual Studio. Click the Tools menu and then Options...
- ☐ In the Environment branch click General.



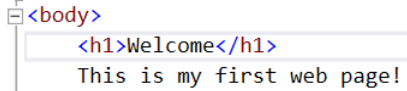
- ☐ Click the Color theme dropdown and click Dark.
- ☐ Click OK.
- ☐ Don't like it? Go back to Tools and Options and try Light.
- ☐ Don't like it? Switch back to Blue.
- ☐ While there are 1000's of options to experiment with, here is one you might find useful: Go to Tools and Options. Expand Text Editor and expand HTML. Click Advanced and change Enable Tag Navigator from false to true. Click OK.
- ☐ In your HTML editor click in the line of text you typed earlier. Notice at the bottom of the editor that there's now a bar showing where you are in your HTML structure.



- ☐ Click on the <html> box in the navigator and notice how the text above is now highlighted.
- ☐ Click on the <body> box in the navigator and notice how the text above is now highlighted. The navigator can help you to understand the structure of a page, and to quickly find the beginning and ending tags of a block of code.

Exercise 3 – Experimenting with tags

- ☐ Click in front of the word “This” in the text you typed earlier. Press Enter to open a new line.
- ☐ In that new line type `<h` (just the less than symbol and the letter “h”).
Note the dropdown with all of the options that include “<h”. This is a list of tags that are legal inside of the current parent tag (`<body>`).
- ☐ Finish typing the `<h1>` tag and notice that end tag (`</h1>`) was automatically typed for you.
- ☐ Between these two tags type “Welcome”.



```
<body>
  <h1>Welcome</h1>
  This is my first web page!
```

- ☐ View the page in a browser.



Welcome

This is my first web page!

- ☐ Close the browser and close Visual Studio.

While you could have just used Notepad to create your page, editors like Visual Studio make it easier by providing:

- Intellisense and auto-complete type features.
- Color coding of the code to identify tags, text and errors.
- Tools to create images, icons, CSS and XML files.

After completing the lab review the proposed solution on the next page for tips.

Q&A Time!

Did you find anything new or interesting, or that you would like to more about?

Anything not work as expected?

Module 1 Lab Solution

Questions

☐ Which is the latest version of HTML?

- ☐ HTML 4.01
- ☐ HTML 5
- ☐ **HTML 5.1**
- ☐ XHTML

A better question might have been “Which is latest, reasonably supported, version of HTML?” Which would have been “HTML 5”.

☐ Which of the following browsers support HTML5 and CSS3?

- ☐ **Microsoft Edge**
- ☐ Internet Explorer 6 and 7
- ☐ Internet Explorer 8 and 9
- ☐ **Internet Explorer 10 and 11**
- ☐ **Google Chrome**
- ☐ **Mozilla Firefox**
- ☐ **Apple Safari**

For a better answer, visit <http://caniuse.com> and <http://html5test.com>.

☐ HTML stands for:

- ☐ Hypertext Meta Language
- ☐ Harvard Text Markup Language
- ☐ **Hypertext Markup Language**
- ☐ Hot Typesetting Markup Language

☐ CSS Stands for:

- ☐ Cyclic Style Sheets
- ☐ **Cascading Style Sheets**
- ☐ Common Style System
- ☐ Cascading Scripting Styles

☐ Shims and Shivs:

- ☐ Add new features to web browsers.
- ☐ Add new features to HTML and CSS.
- ☐ Supply missing functionality to older browsers without impacting newer browsers.
- ☐ **Make a CSS or HTML feature in all recent browsers behave exactly the same way.**

☐ Playfills:

- ☐ Add new features to web browsers.
- ☐ Add new features to HTML and CSS.
- ☐ **Supply missing functionality to older browsers without impacting newer browsers.**
- ☐ Make a CSS or HTML feature in all recent browsers behave exactly the same way.

☐ Which of the following URLs are not valid?

- ☐ `http://www.example.com`
- ☐ `http://example.com`
- ☐ `http://example.com/home.html`
- ☐ `http://www.example.com:81`
- ☐ **`http://www.example.com.81`** (see note 1 below)
- ☐ **`http://www.example.com/81`** (see note 2 below)
- ☐ **`httpx://www.example.com`**
- ☐ `https://www.example.com`
- ☐ **`http://www.example.com.homepage.html`** (dot before page name)
- ☐ `www://example.com`

Note 1: In theory, this could be valid if there was a top-level domain named “81”.

Note 2: In theory, this could be valid if there was a folder named “81”.

☐ Which of the following HTTP Status Codes is for a missing page?

- ☐ 200
- ☐ 400
- ☐ 401
- ☐ 403.20
- ☐ **404**
- ☐ 500
- ☐ 504

Module 2 Lab – Core HTML Elements

Questions

☐ Which of these elements must have a closing tag?

- ☐ `<body>`
- ☐ ``
- ☐ `<div>`
- ☐ ``
- ☐ `
`
- ☐ ``
- ☐ `<i>`

☐ Which of the following are block elements?

- ☐ ``
- ☐ `<div>`
- ☐ ``
- ☐ `
`
- ☐ ``
- ☐ `<i>`

☐ Which of the following is correct?

- ☐ `
</br>`
- ☐ `
`
- ☐ `
`
- ☐ `
`

☐ Which of the following is CSS code:

- ☐ `<h1>Welcome!</h1>`
- ☐ `style="color:red;"`
- ☐ `<style> h1 { font-size:24pt; } </style>`
- ☐ ``

☐ Which of the following is case sensitive?

- ☐ The server name part of a URL `http://www.Example.com`
- ☐ The part of the URL after the server name: `http://www.Example.com/Sales/Report.Html`
- ☐ CSS ID and Class selectors.
- ☐ CSS properties and values.
- ☐ XML
- ☐ HTML
- ☐ JavaScript

- ☐ Which of the following is a correct HTML comment?
 - ☐ // This is a comment
 - ☐ /* This is a comment */
 - ☐ # This is a comment #
 - ☐ <!-- This is a comment -->

- ☐ Which is the correct DOCTYPE for HTML 5?
 - ☐ <!DOCTYPE html>
 - ☐ <!DOCTYPE html 5.0>
 - ☐ <!DOCTYPE html PUBLIC "-//W3C//DTD HTML 4.0//EN">
 - ☐ <!DOCTYPE html PUBLIC "-//W3C//DTD HTML 5.0//EN">
 - ☐ <!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN" "http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd">

- ☐ Which of the following is true for the <title></title> tag?
 - ☐ Only allowed as the first line inside of <body>.
 - ☐ Only allowed inside of the <head> block.
 - ☐ Should be used at the top of each section of a page.
 - ☐ Is displayed in the browser's tab or title bar.
 - ☐ Is the default text for bookmarks and favorites.

- ☐ Are descriptive metadata tags such as <meta name="keywords"> the best way to improve search engine placement? I.e. improve search engine optimization (SEO).
 - ☐ Yes
 - ☐ No

- ☐ Which browsers and devices should you test your site with?
 - ☐ Only the top three brands (Chrome, Firefox, Safari) less than two years old.
 - ☐ Only the top two types of phones: Android and Apple.
 - ☐ Every browser ever made.
 - ☐ Those used by the users you want to do business with.

Hands on Lab

Exercise 1 – Creating and Validating a Page

- ☐ Create a page using Notepad.
 - Open Notepad.
 - Type the following HTML and text.

<!DOCTYPE html>

```

<html>
<head>

  <title>Nancy's Flowers</title>

</head>
<body>

<h1>Welcome to Nancy's Flowers</h1>

<h2>This month's special: Clematis</h2>



<h2>Last month's special</h2>



</body>
</html>

```

- Open another Notepad and click File and Open.
- Navigate to C:\HTMLCSS\Labs and open SampleText.txt
- Copy the text under “Sample 1”.
- Paste text in your new web page directly under the first tag and again under the second tag.
- Click File, SaveAs.
- Change the Encoding to UTF-8.
- Enter the filename “notepadtest.html”
- Save the file to your Lab2 folder. (Most often C:\HTMLCSS\Labs\Labs2)
- Open a browser type the path into the address bar.
Most often C:\HTMLCSS\Labs\Labs2\notepadtest.html
- Verify that the page works. If not, return to Notepad and fix the problem.

☐ Validate your HTML.

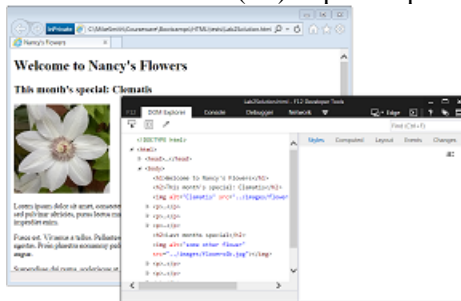
- Copy the web page’s text from Notepad.
- Open a browser and navigate to: <http://validator.w3.org>
- Click the Validate by Direct Input tab.
- Paste in the code you copied from Notepad.
- Click Check.
- Scroll down and review the validation report.
- Fix any errors!
 - Warnings tend towards best practices. You will probably see a warning about the language attribute. Click the Declaring the overall language of a page link to see the recommendations.

- If you want a perfect score on the validation, then add “lang=“en”” (or the language you use in your page!) to the <html> tag.
`<html lang=“en”>`

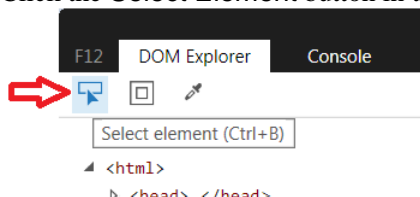
Exercise 2 – Browser tools

Each browser offers a set of web developer tools. While each has their own names and shortcut keys for these features, most use the F12 key to open them. For this example we will use Internet Explorer, but you should also open any other browsers you have installed and check out their developer tools.

- ❑ Open Internet Explorer.
- ❑ Open the web page you created in Exercise 1.
- ❑ Press F12 or select **F12 Developer Tools** from the **Tools** menu.
- ❑ The tools panel will probably open and display in the bottom half of the browser. You can click the Unpin icon (📌) to float the debug pane so you can drag it away from the browser, or onto a second monitor. You can click the Pin icon (📌) to put the pane back into the browser.



- ❑ Click the DOM Explorer tab and note the display of the HTML. Your tags are now displayed as a collection of elements. Each can be expanded or collapsed.
 - Click the arrow next to <head> to reveal the title tag.
- ❑ You can explore your page by just clicking an object, such as an image or heading.
 - Click the Select Element button in the DOM Explorer tab.



- Click the flower image in the page and note that the related HTML is now highlighted in the DOM Explorer.
 - Click the Select Element button again and click the first heading in the page (“Welcome ...”) and note that the <h1> tag is now selected in the DOM Explorer.
- ❑ The Console tab is used to display errors, warnings and information about recent activity. You can filter these by clicking these three buttons:



- ❑ Click the Debugger tab. This is used for testing and debugging JavaScript.
- ❑ Click the Network tab. This is used for monitoring requests from the page to the server and back.

- Click the green arrow to start logging activity.



- Click the refresh button in the browser.



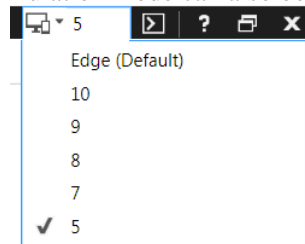
- You should see three entries. One for the HTML file and two for the images. Here you can see how long each request took, and if there were any errors. You will find more useful information when accessing a web server.

URL	Protocol	Met...	Result	Type	Received	Taken	Initiator	Time
file:///C:/HTMLCSS/Labs/Lab2/Lab2Solution.html	file		(File)		0 B	< 1 ms	refresh	
file:///C:/HTMLCSS/Labs/images/flowers4b.jpg	file		(File)		0 B	< 1 ms		
file:///C:/HTMLCSS/Labs/images/flowers1-20.jpg	file		(File)		0 B	< 1 ms		

- Click the Emulation tab. You may need to click the dropdown to find it.
- From the Document mode dropdown click each version and watch to see if the display of the page changes.

Notes:

- Internet Explorer only emulates older IE versions.
- “Edge” is not the new browser supplied with Windows 10. It is the current version of the browser. (Most likely IE 11.)
- Emulation mode can also be selected from a dropdown in the F12 tool bar.



- Close the F12 tools by clicking the X at the far right of its toolbar.
- Open each of the other browsers you have installed on your PC and take a look at their developer tools.

Exercise 3 – Add a Copyright Message

- Return to your Notepad file and scroll to the bottom of the page.
- Just above the `</body>` tag add a copyright message. As this message will need to use the copyright symbol, we will need to use and HTML entity (`©`):
Copyright `©` 2017
- Return to your browser and refresh the page to confirm that the message is there.
- This would look better centered, so we need to put it in a block-level container, a `<div>`, and then style that container. Add the `<div>` around your copyright message.
`<div>Copyright © 2017</div>`
- Now add a style attribute to the `<div>` to center the text.
`<div style="text-align:center;" >Copyright © 2017</div>`
- Return to your browser and refresh the page to confirm that the message is now centered.

- ☐ The customer has looked at the page and has asked to have the copyright message displayed in bold. There are at least three ways to do this:
 - Use the bold tags:
`<div style="text-align:center;" >Copyright © 2017</div>`
 - Use the strong tags:
`<div style="text-align:center;" >Copyright © 2017</div>`
 - Use CSS:
`<div style="text-align:center; font-weight:bold;" >Copyright © 2017</div>`
- ☐ Add all three of the above to your page and confirm that that all produce the exact same display. Keep the third one as your final solution.
- ☐ Which of the three is the best solution? _____

Exercise 4 – Create a Style Sheet

- ☐ Return to your Notepad file and scroll to the bottom of the page. The copyright message there has the CSS inline in the `<div>` tag. If we want to update this in the future we would need to find every page, and every use of that CSS, and manually change it. As a start, let's move it into a style block.
- ☐ Select the existing style text, `"text-align:center; font-weight:bold;"`, and cut it from the page.
- ☐ Remove the style attribute and the quotes leaving the following:
`<div >Copyright © 2017</div>`
- ☐ Give the DIV a class attribute so we can reference it from the style sheet.
`<div class="copyright" >Copyright © 2017</div>`
- ☐ If you refresh the page in the browser now it will still have the copyright text, but it will be left aligned and not bold.
- ☐ Return to your Notepad file and scroll to the top of the page. Just after the `</title>` end tag add the `<style>` block.
`<style>`
`</style>`
- ☐ Inside of the style block add a class selector named `"copyright"`. Note the period that goes in front of class selectors.
`<style>`
`.copyright { }`
`</style>`
- ☐ Paste the CSS you copied earlier between the curly brackets. You can format this across multiple lines if you like.
`<style>`
`.copyright { text-align:center; font-weight:bold; }`
`</style>`
 or
`<style>`
`.copyright {`
`text-align:center;`


```

    font-weight:bold;
}
</style>

```

- ☐ Refresh the page in the browser and confirm that the copyright message is bold and centered.
- ☐ Style the <h1> tags to be green. Inside of the style block added earlier add a tag selector for “h1”.


```

<style>
    .copyright { text-align:center; font-weight:bold; }
    h1 {
    }
</style>

```
- ☐ Add the CSS set the color to red.


```

<style>
    .copyright { text-align:center; font-weight:bold; }
    h1 { color:red; }
</style>

```
- ☐ Note that you could have set the color several ways. We will explore color more in a later module.


```

color:#f00;
color:#ff0000;
color:rgb(255,0,0);

```
- ☐ Refresh the page in the browser and confirm that the headings are now red.
- ☐ For consistency across multiple pages in a site, the style sheet should be stored externally.
- ☐ Copy the two lines from inside of the style block.
- ☐ Open a new Notepad. Paste the two lines of the CSS. You do not use <style> tags in a linked file. The text should look like this:


```

    .copyright { text-align:center; font-weight:bold; }
    h1 { color:red; }

```
- ☐ Save the file as Lab2.css.
- ☐ Return to the Notepad that contains your page.
- ☐ Delete the style block. (<style> ... </style>)
- ☐ Where the style block was add a <link> tag to link to the external style sheet.

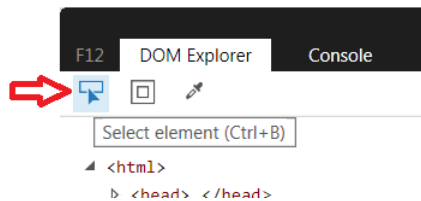

```

<link rel="stylesheet" type="text/css" href="Lab2.css">

```
- ☐ Refresh the page in the browser and confirm that the headings are still green and the footer is bold and centered.

- ☐ Press F12 or select **F12 Developer Tools** from the **Tools** menu.
- ☐ Scroll the page down so you can see the footer.
- ☐ Click the DOM Explorer tab and note the display of the HTML. Your tags are now displayed as a collection of elements. Each can be expanded or collapsed.

- ❑ Click the **Select Element** button in the DOM Explorer tab.



- ❑ Click the copyright line in the page and note that the related HTML is now highlighted in the DOM Explorer AND the being applied to the copyright line is displayed along with a note as to where it came from (Lab2.css).



- ❑ You can “play” with the CSS here. Click in the blank space below the “Inline style” text to type and display CSS real-time. The change will not be saved anywhere. If you want to keep the change you will need to edit your page or the CSS file.

Try some of samples below. You will need to type these in two parts. Type the property (“color”), press TAB, type the value “green” then press Enter.

```
color:green;
font-family:arial;
transform: rotate(15deg);
text-shadow: 4px 4px 4px gray;
```

- ❑ Close the F12 pane.

Q&A Time!

Find anything new or interesting?

Anything not work as expected?

Module 2 Lab Solution

Questions

☐ Which of these elements must have a closing tag?

- ☐ `<body>`
- ☐ ``
- ☐ `<div>`
- ☐ ``
- ☐ `
`
- ☐ ``
- ☐ `<i>`

☐ Which of the following are block elements?

- ☐ ``
- ☐ `<div>`
- ☐ ``
- ☐ `
`
- ☐ ``
- ☐ `<i>`

☐ Which of the following is correct?

- ☐ `
</br>`
- ☐ `
`
- ☐ `
`
- ☐ `
`

Depends on the version of HTML. The forth is the correct version for HTML 5. Both the second and third are for versions before that. Only the first will be flagged by the validators as an error. All will be handled by every browser.

☐ Which of the following is CSS code:

- ☐ `<h1>Welcome!</h1>`
- ☐ `style="color:red;"`
- ☐ `<style> h1 { font-size:24pt; } </style>`
- ☐ ``

☐ Which of the following is case sensitive?

- ☐ The server name part of a URL `http://www.Example.com`
- ☐ The part of the URL after the server name: `http://www.Example.com/Sales/Report.Html (*)`
- ☐ **CSS ID and Class selectors.**
- ☐ CSS properties and values.
- ☐ **XML**
- ☐ HTML

☐ **JavaScript**

* Depends on the server. Some servers and applications are case sensitive after the server name.

☐ Which of the following is a correct HTML comment?

- ☐ // This is a comment
- ☐ /* This is a comment */
- ☐ # This is a comment #
- ☐ **<!-- This is a comment -->**

The first and second are comment blocks for JavaScript. The second is also a comment block for CSS.

☐ Which is the correct DOCTYPE for HTML 5?

- ☐ **<!DOCTYPE html>**
- ☐ <!DOCTYPE html 5.0>
- ☐ <!DOCTYPE html PUBLIC "-//W3C//DTD HTML 4.0//EN">
- ☐ <!DOCTYPE html PUBLIC "-//W3C//DTD HTML 5.0//EN">
- ☐ <!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN" "http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd">

☐ Which of the following is true for the <title></title> tag?

- ☐ Only allowed as the first line inside of <body>.
- ☐ **Only allowed inside of the <head> block.**
- ☐ Should be used at the top of each section of a page.
- ☐ **Is displayed in the browser's tab or title bar.**
- ☐ **Is the default text for bookmarks and favorites.**

☐ Are descriptive metadata tags such as <meta name="keywords"> the best way to improve search engine placement? I.e. improve search engine optimization (SEO).

- ☐ Yes
- ☐ **No**

☐ Which browsers and devices should you test your site with?

- ☐ Only the top three brands (Chrome, Firefox, Safari) less than two years old.
- ☐ Only the top two types of phones: Android and Apple.
- ☐ Every browser ever made.
- ☐ **Those used by the users you want to do business with.**

Module 3 Lab – Cascading Style Sheets

Questions

- ☐ CSS Selectors can be used to select:
 - ☐ HTML tags such as H1.
 - ☐ HTML tag content. `<h1>this is content</h1>`
 - ☐ Tags with a class attribute `<h1 class="boldRed">this is content</h1>`
 - ☐ Tags with an ID attribute. `<h1 id="productName">this is content</h1>`
 - ☐ Tags with a name attribute. `<h1 name="productNumber">this is content </h1>`

- ☐ When working with CSS for page layout you should use:
 - ☐ `<div>` tags.
 - ☐ HTML context tags like `<section>`, `<header>`, `<footer>`, `<article>`, etc.
 - ☐ `<table>` tags.
 - ☐ Tags such as `<p>`, ``, ``... i.e. none of the above.

- ☐ Which of the following is well supported in most modern browsers:
 - ☐ CSS1
 - ☐ CSS2
 - ☐ CSS3
 - ☐ CSS4

- ☐ Which of following prefixes might be used to improve CSS support in some browsers?
 - ☐ -flex-
 - ☐ -ms-
 - ☐ -moz-
 - ☐ -apple-
 - ☐ -ms-
 - ☐ -webkit-

- ☐ CSS lets you use several different units of size. Which of the following is true?
 - ☐ px (pixel) is always one dot or pixel on a screen.
 - ☐ pt (point), being from the typesetting world, is a more accurate choice for fonts.
 - ☐ in (inch) is not a consistent unit due to varying screen sizes.
 - ☐ 1in = 96px = 2.54cm = 72pt always.

- ☐ When designing pages for multiple devices you should:
 - ☐ Use absolute units (ex: 100px) to ensure tight control of the display the page.
 - ☐ Use relative units (ex: 150%) to let elements scale base on the size of other page objects.

- ☐ After enclosing one styled block inside of another styled block the inner most text changed size unexpectedly.

```
<div style="font-size:120%">
  <div style="font-size:2em">
    <span style="font-size:.8em">Hello!</span>
  </div>
</div>
```

Which of the following is true?

- ☐ With the current design, all of the units are relative to next immediate parent. When the outer <div> was added it made everything it contains 120% larger.
- ☐ To fix the font to a never changing size change the style to an absolute unit such as px, pt or cm.
- ☐ “em” should have been used instead of “%” on the outer <div>.
- ☐ To eliminate the scaling of the by the addition of the <div>, but so still let the text scale with default or root font size, use “rem” instead of “em”.
- ☐ All of the sizes should have been in absolutes like px or pt.

- ☐ Which of the following will style all of the heading tags:

- ☐ h1, h2, h3, h4, h5, h6 { color:red; }
- ☐ h1 h2 h3 h4 h5 h6 { color:red; }
- ☐ h1>h2>h3>h4>h5>h6 { color:red; }
- ☐ h1+h2+h3+h4+h5+h6 { color:red; }

- ☐ Valid CSS Media Types include:

- ☐ screen
- ☐ tablet
- ☐ speech
- ☐ phone
- ☐ print

- ☐ You have a <div> with a width of 100px. You add a 5px border, 2px margin and 10px padding. How much is being used by the div?

- ☐ 100px
- ☐ 102px
- ☐ 105px
- ☐ 110px
- ☐ 114px
- ☐ 134px

- ☐ For the same <div> listed above, how much space is available inside of the <div> for content?

- ☐ 66px
- ☐ 80px

- ☐ 96px
 - ☐ 100px
 - ☐ 134px
-
- ☐ Shapes like triangles can be created with:
 - ☐ HTML Entities
 - ☐ CSS and a <div>
 - ☐ Images and the tag
 - ☐ HTML 5 canvas object and JavaScript
 - ☐ HTML 5 SVG and XML

Hands on Lab

Exercise 1 – Creating and Validating a Page

- ☐ On your lab PC or virtual machine, launch Visual Studio.
- ☐ In Visual Studio click the File menu, New and Web Site.
- ☐ Even though we will not be writing any C# code, select “Visual C#” from the Templates folder.
- ☐ Click ASP.NET Empty Web Site.
- ☐ At the bottom of the dialog box click the “Web Location” dropdown and note that Visual Studio can edit files on your local file system (C:), directly to a web server using HTTP (a server with Front Page Extensions) or directly to a web server using FTP (File Transfer Protocol – ftp://www.example.com).
Leave File System selected.
- ☐ At the bottom of the dialog box delete the default path and enter:
C:\HTMLCSS\Lab3\website
(Ask your instructor if you should be using another location.)
Tip: If you type a path with non-existent folders, Visual Studio will create them for you!
- ☐ Click OK to create your new web site project.
- ☐ You now have an (almost) empty web site. As Visual Studio’s templates are designed for ASP.NET projects it has added two files to the project: web.config and web.debug.config.
- ☐ Right-click Web.config, click Delete and OK.
- ☐ We need a new web page... In the Visual Studio menus click Website and Add New Item.
- ☐ Browse to C:\HTMLCSS\Labs3, select “Lab3starter.html” and click Add.
- ☐ In the Solution Explorer pane, click “Lab3starter.html”.
- ☐ Take a look at the page. In the menu bar, click File and then click View in Browser (Internet Explorer). Or, right-click the file’s name in the Solution Explorer pane and click View in Browser (Internet Explorer). Or, right-click any blank space around your HTML code and click View in Browser (Internet Explorer).

- ❑ In the Solution Explorer, right-click lab3starter.html and click Rename. Change the name to “default.html”.

This page needs a little work!

- The title needs be bigger, bolder and centered.
- The headings need a change in fonts, color and size.
- The navigation area needs to look like navigation.
- The copyright and links at the bottom of the page need work.

We are going to mix a little “best practice” CSS with a little not so best practice to explore how CSS cascades.

- ❑ Let’s add a style sheet to the project.
 - We need a new file... In the Visual Studio menus click **Website** and **Add New Item**.
 - In the dialog box, click **Style Sheet** and at the bottom change the name to “TWPlstyles.css”.
- ❑ Change the default font size for the page to 18px and set the color to blue.

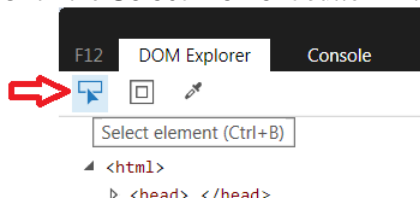
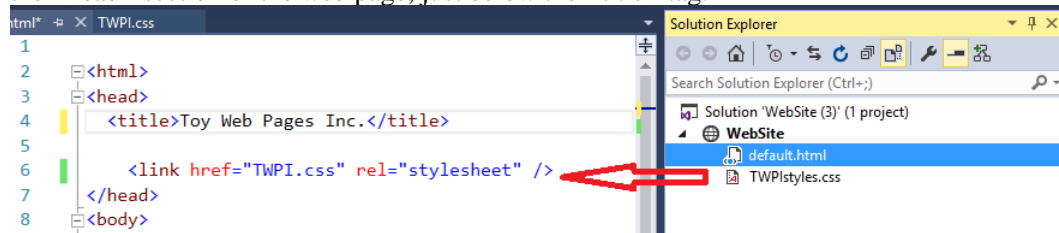
```
body {
    font-size:18px;
    color:blue;
}
```
- ❑ Let’s change the font for the headings to Arial.

```
body {
    font-size:18px;
    color:blue;
}
h1, h2 {
    font-family:arial;
}
```

Notice how the Visual Studio helps you write the CSS with the dropdowns of valid choices and color coding.

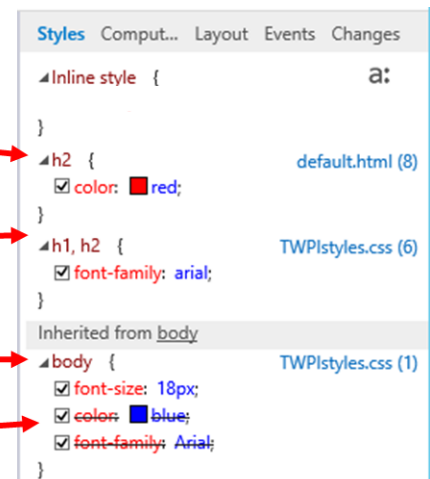
- ❑ Save the CSS file.
- ❑ Click on the tab at the top of the editor window for **default.html** or in the Solution Explorer pane click **default.html**.
- ❑ We now need to link the external CSS file to our page. You can type the link:
`<link href="TWPlstyles.css" rel="stylesheet" />`
or you can drag the file and let Visual Studio create the link for you.

- Place your mouse over the CSS file name in the Solution Explorer. Don't click, just drag the file into the <head> section of the web page, just below the <title> tag.



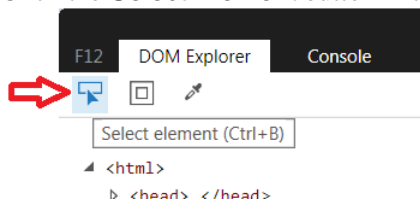
- Click on "Building Blocks!" text. (<h2> tagged text.)
- Note the Styles panel on the left. Read it from the bottom up.

- 3 The text was then styled with the "h2" style added directly to the page.
- 2 The text was then styled with the "h2" style, also found in the TWPIstyles.css file
- 1 The text was first styled by the "body" tag style in the TWPIstyles.css file.
But the font-family and the color was overwritten or cascaded by something else.



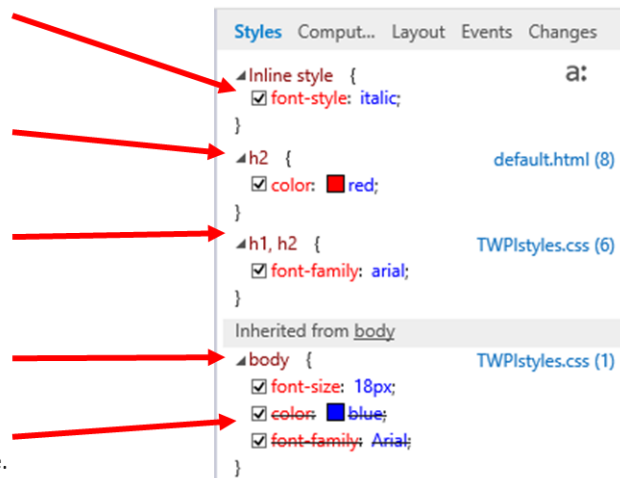
- ❑ The “New Car Kits!” need to be italic. While we could add `<i>` or `` tags around the text, it’s usually better to use CSS. Let’s make this change by using inline CSS.
 - Return to Visual Studio and our web page file.
 - Find the “`<h2>New Car Kits!</h2>`” section.
 - Add an inline style to the `<h2>` tag to make the text italic.


```
<h2 style="font-style:italic;">New Car Kits!</h2>
```
- ❑ Save the page, switch back to the browser and refresh the page. The “New Car Kits!” text should now be italic.
- ❑ If the F12 tools are not currently displayed, press F12 or select **F12 Developer Tools** from the **Tools** menu.
 - Click the Select Element button in the DOM Explorer tab.



- Click on “New Car Kits!” text. (`<h2>` tagged text.)
- Note the Styles panel on the left. Read it from the bottom up.

- 4 The text was then styled with an inline style.
- 3 The text was then styled with the “h2” style added directly to the page.
- 2 The text was then styled with the “h1” style, also found in the TWPIstyles.css file
- 1 The text was first styled by the “body” tag style in the TWPIstyles.css file.
But the font-family and the color was overwritten or cascaded by something else.



- This is “cascading” at work. The CSS is applied in order as it is found:
 - The linked TWPIstyles.css was applied first, with styles applied top down.
 - The `<style>` block is then applied.
 - Any inline CSS is then applied.
- The F12 tools can really help you debug CSS issues.

Exercise 2 – CSS Box Model – Customize the Header

- ❑ Return to Visual Studio and take a look at the `<header>` tag and its contents. We need to make the font larger and wrap in a big box. Maybe something like this:

Toy Web Pages Inc. 😊

Too ugly? Then you be the designer! Get creative!

- ☐ Display the TWPIstyles.css file again. At the bottom of the existing CSS code add a definition for the <header> tag.

```
header { }
```

- ☐ Make the font 48px and centered.

```
header {  
  font-size:48px;  
  text-align:center;  
}
```

- ☐ Save your changes and return to the browser and refresh the page and confirm the changes.

- ☐ Add a border.

```
header {  
  font-size:48px;  
  text-align:center;  
  border:1px solid;  
}
```

or:

```
header {  
  font-size:48px;  
  text-align:center;  
  border-width:1px;  
  border-style:solid;  
}
```

- ☐ Save your changes and return to the browser and refresh the page and confirm the changes.

- ☐ Set a background color of light blue.

```
header {  
  font-size:48px;  
  text-align:center;  
  border:1px solid;  
  background-color:lightblue;  
}
```

- ☐ Save your changes and return to the browser and refresh the page and confirm the changes.

- ☐ Add some space around the text.

```
header {  
  font-size:48px;  
  text-align:center;  
  border:1px solid;  
  background-color:lightblue;  
  padding:40px;  
}
```

- ☐ Save your changes and return to the browser and refresh the page and confirm the changes. Don't like it? Then you make it pretty!

Exercise 3 – CSS Box Model – Adding Navigation Tabs

In this exercise we will make the links at the top of the page look like tabs.



- ☐ Return to Visual Studio and take a look at the `<nav>` tag and its contents. Anchor tags (`<a>`) are inline elements and are usually formatted as text. The CSS box model can be used to greatly change appearance from text to something more graphical.
By default the `<a>` is styled as blue underlined text. We need to first turn off the underline.
- ☐ Display the `TWPIstyles.css` file again.
- ☐ Add CSS to reformat only the `<a>` tags that are inside of a `<nav>` tag and remove the underline.

```
nav a {  
    text-decoration:none;  
}
```
- ☐ Add CSS to add a border around the `<a>` tags.

```
nav a {  
    text-decoration:none;  
    border:1px solid;  
}
```
- ☐ Save your changes and return to the browser and refresh the page and confirm the borders.
- ☐ You should now have boxes around the `<a>` tags.
- ☐ Add rounded corners to the tabs.

```
nav a {  
    text-decoration:none;  
    border:1px solid;  
    border-radius:5px 5px 0 0;  
}
```
- ☐ Save your changes and return to the browser and refresh the page and confirm the borders.
- ☐ That looks better, but there are gaps between the tabs. You can fix this by removing the spaces between the `<a>` tags or by adjusting the margins. Margins are usually added as a positive number, but you can also use negative margins.

```
nav a {  
    text-decoration:none;  
    border:1px solid;  
    border-radius:5px 5px 0 0;  
    margin-right:-5px;  
}
```
- ☐ Save your changes and return to the browser and refresh the page and confirm the borders.

- ❑ The tabs might look better with a little more space around the text. Padding is used to add space between the border and the object's contents.

```
nav a {  
  text-decoration:none;  
  border:1px solid;  
  border-radius:5px 5px 0 0;  
  margin-right:-5px;  
  padding:5px;  
}
```

- ❑ Save your changes and return to the browser and refresh the page and confirm the borders.
- ❑ Two more things are needed to complete the tab design, some space above the tabs and a line across the bottom. For this we will modify the <nav> container. Add a new style definition for the <nav> tag.

```
nav { }
```

- ❑ Add a bottom border and margin to the top.

```
nav {  
  border-bottom:1px solid;  
  margin-top:10px;  
}
```

- ❑ Save your changes and return to the browser and refresh the page and confirm the borders.
- ❑ Close, but the border is overlapping the tabs. We need to push it down a bit with a little padding.

```
nav {  
  border-bottom:1px solid;  
  margin-top:10px;  
  padding-bottom:5px;  
}
```

Exercise 4 – Working with a Pseudo Class

- ❑ Let's add a little animation to the tabs. We will start by adding a background during mouse over. The <a> supports four pseudo-classes: a:link, a:hover, a:visited and a:active. When the mouse moved over the tag the "hover" state is activated. Note that a pseudo-class is prefixed with a ":" instead of a "." used for class.

```
nav a:hover {  
}
```

- ❑ Add the effect. This example sets a background color, but you could change the font, the width, the color of the text, the color of the border, and just about anything else you can think of.

```
nav a:hover {  
  background-color:lightgray;  
}
```

- ❑ Save your changes and return to the browser and refresh the page and confirm the borders.

- ☐ Your page should now look like this:



Exercise 5 – Validate Your Code

- ☐ Validate your HTML.
- Copy the web page's text from Visual Studio.
 - Open a browser and navigate to: <http://validator.w3.org>
 - Click the Validate by Direct Input tab.
 - Paste in the code you copied from Notepad.
 - Click Check.
 - Scroll down and review the validation report.
 - Fix any errors!
- ☐ Validate your CSS.
- Copy the CSS code from Visual Studio.
 - Open a browser and navigate to: <http://jigsaw.w3.org/css-validator>
 - Click the By Direct Input tab.
 - Paste in the code you copied from Notepad.
 - Click Check.
 - Scroll down and review the validation report.
 - Fix any errors!

Module 3 Lab Solution

Questions

☐ CSS Selectors can be used to select:

- ☐ **HTML tags such as H1.**
- ☐ HTML tag content. `<h1>this is content</h1>`
- ☐ **Tags with a class attribute** `<h1 class="boldRed">this is content</h1>`
- ☐ **Tags with an ID attribute.** `<h1 id="productName">this is content</h1>`
- ☐ Tags with a name attribute. `<h1 name="productNumber">this is content </h1>`

CSS cannot select based on content, only structure.

The “name” attribute is used in HTML form elements to name submitted data values.

☐ When working with CSS for page layout you should use:

- ☐ **<div> tags.**
- ☐ **HTML context tags like <section>, <header>, <footer>, <article>, etc.**
- ☐ <table> tags.
- ☐ Tags such as <p>, , ... i.e. none of the above.

DIVs and DIV-like tags provide the most functionality for page layout.

☐ Which of the following is well supported in most modern browsers:

- ☐ **CSS1**
- ☐ **CSS2**
- ☐ **CSS3 ***
- ☐ CSS4

CSS3 is “almost ready” in modern browsers. There are still unsupported and inconsistently supported CSS3 features. Use sites like canisue.com and test with every browser and device you need to support.

☐ Which of following prefixes might be used to improve CSS support in some browsers?

- ☐ -flex-
- ☐ -ms-
- ☐ -moz-
- ☐ -apple-
- ☐ -ms-
- ☐ -webkit-

The prefixes indicate browser brands. Apple iOS and Safari use webkit.

- ☐ CSS lets you use several different units of size. Which of the following is true?
- ☐ px (pixel) is always one dot or pixel on a screen.
 - ☐ pt (point), being from the typesetting world, is a more accurate choice for fonts.
 - ☐ **in (inch) is not a consistent unit due to varying screen sizes.**
 - ☐ **1in = 96px = 2.54cm = 72pt always.**

You can use any of the absolute units interchangeably. Inch and centimeter are usually accurate with printers as long as the page is being displayed at 100%.

- ☐ When designing pages for multiple devices you should:
- ☐ Use absolute units (ex: 100px) to ensure tight control of the display the page.
 - ☐ **Use relative units (ex: 150%) to let elements scale base on the site of other page objects.**

But, this depends on your application. You may find that for some uses you will need to take tight control of the display and even create CSS variations for each supported device.

- ☐ After enclosing one styled block inside of another styled block the inner most text changed size unexpectedly.

```
<div style="font-size:120%">
  <div style="font-size:2em">
    <span style="font-size:.8em">Hello!</span>
  </div>
</div>
```

Which of the following is true?

- ☐ **With the current design, all of the units are relative to next immediate parent. When the outer <div> was added it made everything it contains 120% larger.**
 - ☐ **To fix the font to a never changing size change the style to an absolute unit such as px, pt or cm.**
 - ☐ “em” should have been used instead of “%” on the outer <div>.
 - ☐ **To eliminate the scaling of the by the addition of the <div>, but so still let the text scale with default or root font size, use “rem” instead of “em”.**
 - ☐ All of the sizes should have been in absolutes like px or pt.
- ☐ Which of the following will style all of the heading tags:
- ☐ **h1, h2, h3, h4, h5, h6 { color:red; }**
 - ☐ h1 h2 h3 h4 h5 h6 { color:red; }
 - ☐ h1>h2>h3>h4>h5>h6 { color:red; }
 - ☐ h1+h2+h3+h4+h5+h6 { color:red; }

- ☐ Valid CSS Media Types include:

- ☐ **screen**
- ☐ tablet
- ☐ **speech**
- ☐ phone

☐ **print**

☐ You have a <div> with a width of 100px. You add a 5px border, 2px margin and 10px padding. How much is being used by the div?

- ☐ 100px
- ☐ 102px
- ☐ 105px
- ☐ 110px
- ☐ 114px
- ☐ **134px**

$$100 + 5*2 + 2*5 + 10*2 = 134$$

☐ For the same <div> listed above, how much space is available inside of the <div> for content?

- ☐ 66px
- ☐ 80px
- ☐ 96px
- ☐ **100px**
- ☐ 134px

☐ Shapes like triangles can be created with:

- ☐ **HTML Entities**
- ☐ **CSS and a <div>**
- ☐ **Images and the tag**
- ☐ **HTML 5 canvas object and JavaScript**
- ☐ **HTML 5 SVG and XML**

All of the above! (But we have not covered Canvas and SVG yet.)

Module 4 Lab – Fonts and Text

Questions

- ☐ Which of the following is a Sans-serif font?

 - ☐ Arial The quick brown fox.
 - ☐ Times Roman The quick brown fox.
 - ☐ Courier New The quick brown fox.
 - ☐ WingDings *~m ™ ☐◆✕Ⓜ& ∂ ☐☐◆■ ↗ ☐☒.
 - ☐ Tahoma The quick brown fox.

☐ For best font support you should:

 - ☐ Only use the browser default fonts: sans-serif, serif, monospace, script
 - ☐ Use fonts common the devices you wish to support.
 - ☐ Use fonts common the devices you wish to support, but provide a list of fallback fonts ending with the browser default font nearest to your preferred font.

☐ Which of the following will produce italic text:

 - ☐ CSS: font-style:italic
 - ☐ CSS: font-face:italic
 - ☐ Tag:
 - ☐ Tag: <i>
 - ☐ Tag: <italic>

☐ Which of the following will produce 12px text in every browser:

 - ☐ font-size:12px
 - ☐ font-size:x-small
 - ☐ font-size:smaller
 - ☐ font-size:medium

☐ Which of the following should be used when you want to stress text in both a browser and a screen reader:

 - ☐ Tag:
 - ☐ Tag:
 - ☐ CSS: font-syle:bold

☐ Which of the following will center text:

 - ☐ CSS: font-align:center
 - ☐ CSS: text-align:center
 - ☐ CSS: float:center

- ☐ Tag: `<center>some text</center>`
- ☐ You need to supply a bulleted list of product names. You should use:
 - ☐ Tag: ``
 - ☐ Tag: ``
 - ☐ Tag: `<table>`
 - ☐ CSS: `tag:before { content:"• "; }`

Hands on Lab

Exercise 1 – Selecting Fonts

You have been asked to redesign a site by just editing the CSS. It's best if you don't have to make any HTML changes.

- ☐ On your lab PC or virtual machine launch Visual Studio. If Visual Studio is already open with another project, then click File, Close Solution. (If prompted to save a ".sln" file, click No.)
- ☐ In Visual Studio click the File menu, Open and Web Site. In the Open Web Site dialog type or navigate to `C:\HTMLCSS\Lab4` and click Open.

The web site is similar to the one you created in Lab 3.

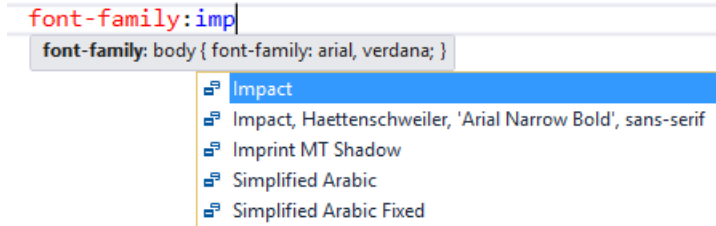
- ☐ In the Solution Explorer pane click `default.html` and review the code.
- ☐ In the Solution Explorer pane click `TWPIstyles.css` and review the code.
- ☐ Redisplay `default.html` and then view the page in a browser. (File, View in Browser)
Our corporate design standards encourage the following:
 - Company Logo: font: Impact
 - Major headings (h1, h2, etc.) font: Comic Sans MS
 - Special promotion headings: font: same as Major heading with a shadow and no other effects.
 - All other text: font: Cambria
- ☐ Step 1... you probably want to talk to the design people about the Comic Sans MS font. While it's on Windows clients, it may not be that common on other devices. Fallback will probably be the browser's generic cursive font. And... many people really hate Comic Sans! But, you can't reach the design folks right now, so we will just roll on with Comic Sans.
- ☐ In the Solution Explorer pane click `TWPIstyles.css`, or click the tab in the editor for the style sheet.
- ☐ Let's start with the company name. If we could make changes to the HTML, it might be best to wrap the company name in a span, give the span an ID and then style that ID.

```
<span ID="companyName">Toy Web Pages Inc.</span>
```

 and

```
#ID { css code here }
```
- ☐ But as we have been asked not to change the HTML code we will style the `<header>` tag. In the existing style sheet there is already a "header" block. Add the font selection code in that block.

- Add a blank line and type “font-family” and the colon (:). Start typing “Impact” and note that Visual Studio not only has it in the list, but has a recommended list of fallbacks.



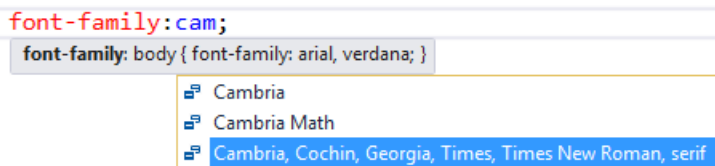
- Until you get some time to do some web searching or browser testing to see what works in your list of “must support” devices, take the recommendation for now.

```
font-family: Impact, Haettenschweiler, 'Arial Narrow Bold', sans-serif
```

- ❑ Save your changes and return to the browser and refresh the page and confirm the font change.
- ❑ Go back to Visual Studio.
- ❑ Add the change to make all <h?> tags to use Comic Sans MS and add a fallback to “cursive”. Note that the font name includes spaces, so enclose it in single or double quotes.
 - As there is already block for h1 and h2, update it:


```
h1, h2, h3, h4, h5, h6 {
  font-family: "Comic Sans MS", cursive;
}
```

- ❑ All other text should be in “Cambria”. As the existing style sheet already has a “body” block, just edit the existing font-family property. Delete “arial”. As you start typing “Cambria” note that Visual Studio has a recommendation for fallbacks. Go ahead and use the recommendation, and later test with various browsers and devices.



- ❑ You’ve also been asked to add shadows to “special promotion” headings. Taking a look at the HTML and the text it looks like this is the “New Car Kits!” heading. We have a couple of problems here:
 - That heading has not been given an ID or a class name (“class=“promotion””).
 - That heading has an inline style that probably should be part of CSS definition stored in the external style sheet. In any case they said “no other effects”, so we have to get rid of the italics.
- ❑ While the best practice would be delete the inline style and add a class name, we were asked to not change the HTML. So how could you select this one, and only one, <h2> tag? (don’t read ahead! ☺)
- ❑ The only thing that makes it unique is the existing inline style. We can use a combination of a tag selector and an attribute selector. As CSS is bit finicky about selectors based on CSS, we will need to use a wildcard match here. Something like this:


```
h2[style*="italic"]
```
- ❑ So let’s get rid of the italics. Try this:


```
h2[style*="italic"] {
  font-style: normal;
}
```

- ❑ Save your changes and return to the browser and refresh the page and see if the italics is gone. It's not! Remember the order for cascading of CSS: the last change applied overrides all of the other changes. I.e. the inline CSS is the last applied (or the closest to the text). You can use the F12 tools to verify this. Note that the top most (last applied) CSS is the inline style.

font-style is first defined here, but overridden here.



- ❑ We can force the override of an inline style by adding “!important” to our style definition.

```
h2[style*="italic"] {
  font-style:normal; !important
}
```

- ❑ Save your changes and return to the browser and refresh the page and confirm the font change.

Now the style sheet wins over the inline style.



- ❑ Now let's add the shadow CSS to the same block as the “font-style:normal”. . Look it up in your handout, or Google/Bing it!
- ❑ Save your changes and return to the browser and refresh the page and confirm the font change.

Exercise 2 – Validate Your Code

- ❑ Validate your HTML.
 - Copy the web page's text from Visual Studio.
 - Open a browser and navigate to: <http://validator.w3.org>
 - Click the Validate by Direct Input tab.
 - Paste in the code you copied from Notepad.
 - Click Check.
 - Scroll down and review the validation report.
 - Fix any errors!

☐ Validate your CSS.

- Copy the CSS code from Visual Studio.
- Open a browser and navigate to: <http://jigsaw.w3.org/css-validator>
- Click the By Direct Input tab.
- Paste in the code you copied from Notepad.
- Click Check.
- Scroll down and review the validation report.
- Fix any errors!

Module 4 Lab Solution

Questions

☐ Which of the following is a Sans-serif font?

- ☐ **Arial** **The quick brown fox.**
- ☐ Times Roman The quick brown fox.
- ☐ Courier New The quick brown fox.
- ☐ WingDings *~m ™ ◻♦×№& ∂◻◻♦■ ↗◻⊠.
- ☐ **Tahoma** **The quick brown fox.**

Times Roman has serifs, Courier New is a monospace font and WingDings is obviously not text!

☐ For best font support you should:

- ☐ Only use the browser default fonts: sans-serif, serif,, monospace, script
- ☐ Use fonts common the devices you wish to support.
- ☐ **Use fonts common the devices you wish to support, but provide a list of fallback fonts ending with the browser default font nearest to your preferred font.**

☐ Which of the following will produce italic text:

- ☐ **CSS: font-style:italic**
- ☐ CSS: font-face:italic
- ☐ **Tag: **
- ☐ **Tag: <i>**
- ☐ Tag: <italic>

☐ Which of the following will produce 12px text in every browser:

- ☐ **font-size:12px**
- ☐ font-size:x-small
- ☐ font-size:smaller
- ☐ font-size:medium

The named sizes, xx-small ... xx-large, do not have sizes defined in the W3 specification. Each browser may have their definition.

☐ Which of the following should be used when you want to stress text in both a browser and screen reader:

- ☐ Tag:
- ☐ **Tag: **
- ☐ CSS: font-syle:bold

The tag has sematic value. The and CSS only define the display of the text. Screen readers may also add stress to , but is preferred.

☐ Which of the following will center text:

- ☐ CSS: `font-align:center`
- ☐ **CSS: `text-align:center`**
- ☐ CSS: `float:center`
- ☐ Tag: `<center>some text</center>`

The `<center>` tag will center text, but it is obsolete and should not be used.

☐ You need to supply a bulleted list of product names. You should use:

- ☐ Tag: ``
- ☐ **Tag: ``**
- ☐ Tag: `<table>`
- ☐ **CSS: `tag:before { content:"• "; }`**

Bullets imply an unordered list (``). Using CSS we can add a bullet or any other character in front of any tag. Search engines, screen readers and other automation will understand that items in a `` are related while the CSS is just cosmetic and not semantic value. (Note: the “•” may work, but you should use an HTML entity like

Module 5 Lab – Color and Backgrounds

Questions

☐ Which of the following are valid color names?

- ☐ red
- ☐ candyappled
- ☐ #f00;
- ☐ #ffffff;
- ☐ #red;
- ☐ rgb(0,0,255)

☐ Which of the following can have color applied?

- ☐ text
- ☐ borders
- ☐ padding
- ☐ margins
- ☐ <div> backgrounds
- ☐ page backgrounds

Hands on Lab

Exercise 1 – Working with Color

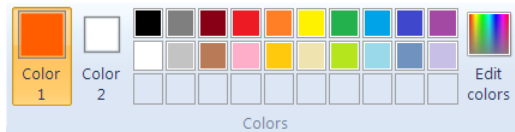
You have been asked give freshen the site by applying some new colors. There are all kinds of strategies for picking color combinations, a color wheel for example, or maybe searching the web for “color theme” sites. One technique is to ask the client to pick a picture that they like and then pick colors from that. If there is a product photo that is used frequently and as part of a banner on the site, see what colors are there.

As an example, our toy site sells a model airplane with a distinctive set of colors. Here’s part of it:

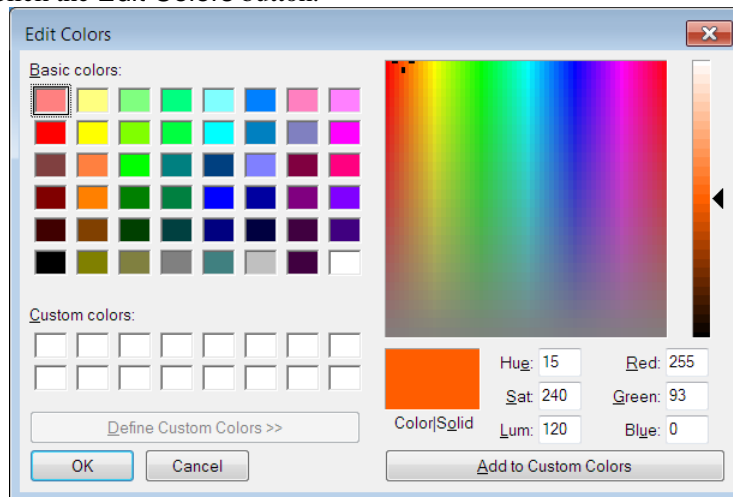


The primary colors here are white, orange (or an orange-red), black and silver. While you may be focusing on the airplane, also consider the green of the grass, or if it was visible, the blue sky. We will pick a set of these colors and give it a try.

- ☐ On your lab PC or virtual machine launch Windows Paint. Open the file C:\HTMLCSS\Labs\Lab5\images\airplane.png.
- ☐ In the toolbar click the color picker tool (eyedropper).
- ☐ Move the mouse to a sample of the orange that you like and click. The color will appear in the Color 1 box in the toolbar.



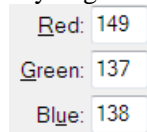
- ☐ Click the Edit Colors button.



- ☐ Note the numbers for Red, Green and Blue. You can use these as-is with the CSS `rgb()` function or you can open the Windows calculator, click View, Programmer, and then convert these three numbers to hexadecimal so you can use them with the “#000000” notation. For the example above we get:
`rgb(255,93,0)` or `#FF5D00` (or `#ff5d00`)

- ☐ Repeat the above steps to get the silver color.

If you got:



then use:

`rgb(149,137,138)` or `#95898A` (or `#95898a`)

That's so close to a gray that you might just use: (Remember, test, test, test!)

`rgb(144,144,144)` or `#909090` (or `#95898a`)

- ☐ Black is, well, black, so just use `rgb(0,0,0)` or `#000000`.
- ☐ Just in case you need one more color later on, grab a grass green too. (But the grass does not look so good in this picture!) Here's one that might work: `rgb(106,159,105)` `#6A9F6B`

- ☐ Now you decide where to use these colors...
 - White is always safe for the page background.
 - The title area might be silver or white text on the orange background. White will be better for contrast. Maybe add a silver shadow to the text and a thick (5px or so) black border around the box.
 - The menu tabs might be white text on a black background.
 - The headings could be in orange.
 - Use a linear gradient (white to orange) behind the <aside> warning text.
 - That leaves black for the rest of the text.
- ☐ On your lab PC or virtual machine launch Visual Studio. If Visual Studio is already open with another project, then click **File, Close Solution**. (If prompted to save a “.sln” file, click **No**.)
- ☐ In Visual Studio click the **File** menu, **Open** and **Web Site**. In the **Open Web Site** dialog type or navigate to **C:\HTMLCSS\Lab5** and click **Open**.

The web site is similar to the one you created in Lab 4.

In the style sheet find and edit the “header” section. You will need to set these properties:

```
color
background-color
text-shadow
```

- ☐ Find and edit the body section and change:

```
color
```
- ☐ Save your changes, return to the browser, refresh the page and see how things look.
- ☐ Find and edit the h1,h2,h3... section and change:

```
color
```

You will need to add “!important” to the color to override the <style> block for h2 in the page.
- ☐ Save your changes, return to the browser, refresh the page and see how things look.
- ☐ Find and edit the “nav a” section (the tabs) and change:

```
color
background-color
```
- ☐ Save your changes, return to the browser, refresh the page and see how things look.
- ☐ And to test how the colors work with the original picture, let’s add the picture to the site.
 - Just above the <h1> tag add the following tag. It uses “float” to position it and wrap text around it. We will see more about “float” in a later module.

```

```
- ☐ Save your changes, return to the browser, refresh the page and see how things look.
- ☐ To reinforce the black pin striping on the plane, add a black bottom border to the <h2> tags.

```
h2 {
  border-bottom-color:black;
  border-bottom-style:solid;
  border-width:2px;
}
```
- ☐ Save your changes, return to the browser, refresh the page and see how things look.

The final page should look something like this:



Exercise 2 - Gradients

In CSS 3, gradients come in two flavors: linear and radial. Gradients are defined using these patterns:

- Default (top to bottom)
property: linear-gradient(*startColor*, *color*, *color*, *endColor*)
background: linear-gradient(red, green)
- With a direction:
property: linear-gradient(*directionToStartFrom*, *startColor*, *endColor*)
direction= to left, to right, to top, to bottom
background: linear-gradient(to top, red, green)
- With a direction (diagonal):
property: linear-gradient(*directionToStartFrom*, *startColor*, *endColor*)
direction= to bottom left, to bottom right, to top right, etc.
background: linear-gradient(to top right, red, green)
- With any angle:
property: linear-gradient(*angle*, *startColor*, *endColor*)
background: linear-gradient(30deg, red, green)
- Radial gradient:
property: radial-gradient(*startColor*, *color*, *color*, *endColor*)
background: radial-gradient(red, green)
- Radial gradient:
property: radial-gradient(*startColor*, *color*, *color*, *endColor*)
shape= ellipse (default), circle
background: radial-gradient(circle, red, green)

- ☐ To highlight the “Small parts and child safety!” warning add a linear gradient, white to orange, across the <aside> area. Your first attempt should be something like this:
- ```
aside {
```

```
background: linear-gradient(white, rgb(255,93,0));
}
```

Test that and you might not like the orange starting under the <h1> text. You can adjust the space used for each color with either percentages or absolute units. Try this one as see if it looks a little better.

```
aside {
background: linear-gradient(white 50px, rgb(255,93,0));
}
```

**Small parts and child safety!**

Lorem ipsum dolor sit amet, consectetur  
adipiscing elit. Maecenas porttitor congue massa.  
Fusce posuere, magna

- ☐ Experiment with some of the other gradient options.
- ☐ Try adding a background gradient to other objects on the page such as the headers.  
h1, h2, h3, h4, h5, h6 {

```
...
background:linear-gradient(white, black);
}
```

**New Car Kits!**

If time allows, pick one of the other pictures in the lab's images folder (mountains, trees, winter, etc.) and create another style sheet for the same web page.

# Module 5 Lab Solution

## Questions

☐ Which of the following are valid color names:

- ☐ **red**
- ☐ candyapplered
- ☐ **#f00**
- ☐ **#ff0000**
- ☐ #red
- ☐ **rgb(0,0,255)**

Color names are just text, so no pound signs.

Using three hexadecimal digits (“#f00”) lets you mix up to 256 colors while six hexadecimal digits (#ff0000) can be used to create 16,777,216 colors. Both of those examples define red.

“rgb()” can be used to create 16,777,216 colors by using three decimal numbers (0..255).

“candyapplered” is not one of the HTML color names. But... you can mix it yourself! #FF0800

☐ Which of the following can have color applied?

- ☐ **text**
- ☐ **borders**
- ☐ **padding**
- ☐ margins
- ☐ **<div> backgrounds**
- ☐ **page backgrounds**

Almost every displayed object can have colors. One exception is an object’s margins.

## Hands on Lab

The final CSS: (TWPIstyles.css)

```
body {
 font-size:18px;
 color:black;
 font-family:Cambria, Cochin, Georgia, Times, Times New Roman, serif;
}

h1, h2, h3, h4, h5, h6 {
 font-family:"Comic Sans MS", cursive;
 color:rgb(255,93,0) !important;
 /* background:linear-gradient(white, black); */
}
```

```
h2 {
 border-bottom-color:black;
 border-bottom-style:solid;
 border-width:2px;
}

h2[style*="italic"] {
 font-style:normal !important;
}

header {
 color:white;
 text-shadow: 4px 4px 4px rgb(149,137,138);
 font-size:48px;
 text-align:center;
 border:5px solid;
 border-color:black;
 background-color:rgb(255,93,0);
 padding:40px;
 font-family:Impact, Haettenschweiler, 'Arial Narrow Bold', sans-serif
}

nav a:hover {
 background-color:lightgray;
}

nav a {
 text-decoration:none;
 border:1px solid;
 border-radius:5px 5px 0 0;
 margin-right:-5px;
 padding:5px;
 color:white;
 background-color:black;
}

nav {
 border-bottom:1px solid;
 margin-top:10px;
 padding-bottom:5px;
}

aside {
 background: linear-gradient(white 50px, rgb(255,93,0));
}
```

# Module 6 Lab – Anchors and Hyperlinks

## Questions

☐ Which of the following tags is used to link to another page?

- ☐ <link>
- ☐ <anchor>
- ☐ <a>
- ☐ <hyperlink>
- ☐ <jump>

☐ Which of the following are valid anchor tag pseudo-classes?

- ☐ a:link
- ☐ a:accessed
- ☐ a:visited
- ☐ a:hover
- ☐ a:clicked
- ☐ a:active

## Hands on Lab

### Exercise 1 – Add links to other pages.

The current home page design only includes dummy anchor tags in the navigation tabs. These need to have real links to other pages.

- ☐ On your lab PC or virtual machine launch Visual Studio. If Visual Studio is already open with another project, then click **File, Close Solution**. (If prompted to save a “.sln” file, click **No**.)
- ☐ In Visual Studio click the **File** menu, **Open** and **Web Site**. In the **Open Web Site** dialog type or navigate to **C:\HTMLCSS\Lab6** and click **Open**.

The web site is similar to the one you created in Lab 5, but has a few extra pages.

- ☐ We need to fix up these links on the page:
  - In the tabs: Home, Toys, Sports, More Information.
  - In the footer: About, Contact Us, Privacy
- ☐ Anchor tags follow this pattern:
  - <a href="pagename.type">displayed content</a>
  - <a href="pagename.type" target="\_blank">displayed content</a> (open in new tab)
- ☐ In Visual Studio select the default.html page and create links as follows:
  - Nav area
    - Home → default.html
    - Toys → Toys.html



- Sports → Sports.html
  - More Information → InfoRequest.html
- Footer area:
  - About → AboutUs.html
  - Contact Us → Contact.html
  - Privacy → Privacy.html

- ☐ Save your changes, display the default.html page in the browser and test the links.
- ☐ Now you have only six more pages to update! Copy the list of links from the <nav> area and then paste them into the following pages:
  - AboutUs.html
  - Contact.html
  - InfoRequest.html
  - Privacy.html
  - Sports.html
  - Toys.html
- ☐ Save your changes, display the default.html page in the browser and test the links.
- ☐ Copy the list of links from the <footer> area and then paste them into the following pages:
  - AboutUs.html
  - Contact.html
  - InfoRequest.html
  - Privacy.html
  - Sports.html
  - Toys.html
- ☐ Save your changes, display the default.html page in the browser and test the links.

Tip: Most server-side technologies have ways to automatically insert a consistent navigation structure into all of the pages. This might use a “control” or use an “include” feature.

## Exercise 2 – External Links

- ☐ The AboutUs.html page has an external link that needs a URL. It should open in a new browser tab or window.  
“Visit the AMA site.” → <http://www.modelaircraft.org/>
- ☐ Save your changes, display the default.html page in the browser and test the link.

## Exercise 3 – Image Links

- ☐ In the default.html page find the <img> tag for the airplane. Wrap this in an anchor tag and have it link to “Toys.html”. Either add a title attribute to anchor tag or an alt attribute to the <img> tag so there will be a mouse over tip. Set the text to “See more toys...”.
- ☐ Save your changes, display the default.html page in the browser, mouse over the airplane picture and confirm that there is a mouse over tip. Click the airplane and confirm the destination.

# Module 6 Lab Solution

## Questions

☐ Which of the following tags is used to link to another page?

- ☐ <link>
- ☐ <anchor>
- ☐ <a>
- ☐ <hyperlink>
- ☐ <jump>

<link> is used to link to external files, usually a CSS file. The other items are not real tags.

☐ Which of the following are valid anchor tag pseudo-classes?

- ☐ **a:link**
- ☐ a:accessed
- ☐ **a:visited**
- ☐ **a:hover**
- ☐ a:clicked
- ☐ **a:active**

## Hands on Lab

The final CSS: (TWPIstyles.css)

```
body {
 font-size:18px;
 color:black;
 font-family:Cambria, Cochin, Georgia, Times, Times New Roman, serif;
}
```

```
h1, h2, h3, h4, h5, h6 {
 font-family:"Comic Sans MS", cursive;
 color:rgb(255,93,0) !important;
 /* background:linear-gradient(white, black); */
}
```

```
h2 {
 border-bottom-color:black;
 border-bottom-style:solid;
 border-width:2px;
}
```

```
h2[style*="italic"] {
 font-style:normal !important;
```

```
}
```

```
header {
 color:white;
 text-shadow: 4px 4px 4px rgb(149,137,138);
 font-size:48px;
 text-align:center;
 border:5px solid;
 border-color:black;
 background-color:rgb(255,93,0);
 padding:40px;
 font-family:Impact, Haettenschweiler, 'Arial Narrow Bold', sans-serif
}
```

```
nav a:hover {
 background-color:lightgray;
}
```

```
nav a {
 text-decoration:none;
 border:1px solid;
 border-radius:5px 5px 0 0;
 margin-right:-5px;
 padding:5px;
 color:white;
 background-color:black;
}
```

```
nav {
 border-bottom:1px solid;
 margin-top:10px;
 padding-bottom:5px;
}
```

```
aside {
 background: linear-gradient(white 50px, rgb(255,93,0));
}
```

# Module 7 Lab – Page Layout

## Questions

- ☐ Which of these is the recommended approach to layout a page?
  - ☐ Use <frameset> to layout the master page and load individual HTML documents into each frame.
  - ☐ Use <table> to build a well-defined layout with heading, navigation, content and footer rows and cells.
  - ☐ Use <div> and div-like HTML 5 tags to define the contextual design and apply CSS to control the display and layout.
  
- ☐ When displaying data laid out in rows and columns use:
  - ☐ <table> with <tr> and <td> tags and CSS
  - ☐ <div> plus CSS
  - ☐ <ul> or <ol> tags with enclosed <li> tags
  
- ☐ Which of the following tags are used to build tables.
  - ☐ <table>
  - ☐ <thead>
  - ☐ <thead>
  - ☐ <tfoot>
  - ☐ <tfoot>
  - ☐ <tr>
  - ☐ <cell>
  - ☐ <td>
  - ☐ <th>
  
- ☐ You have used float to control text wrapping around <div> and <img> tags. How do you make sure the next block of text starts below all floating object?
  - ☐ <p style="float:none">...
  - ☐ <p style="float:full">...
  - ☐ <p style="clear:both">...
  - ☐ <p style="float:all">...

## Hands on Lab

### Exercise 1 – Float Images

You have been given a page with two images and a large amount of text. The current layout is very bad. It would look better if one of the images was on the left, the other on the right and the text flowing around them.

## Flowers - Clematis



Lorem ipsum dolor sit amet, consectetur adipiscing elit. Maecenas porttitor congue massa. Fusce posuere, magna sed pulvinar ultricies, purus lectus malesuada libero, sit amet commodo magna eros quis urna. Nunc viverra imperdiet enim. Fusce est. Vivamus a tellus. Pellentesque habitant morbi tristique senectus et netus et

- ☐ On your lab PC or virtual machine launch Visual Studio. If Visual Studio is already open with another project, then click File, Close Solution. (If prompted to save a “.sln” file, click No.)
- ☐ In Visual Studio click the File menu, Open and Web Site. In the Open Web Site dialog type or navigate to C:\HTMLCSS\Lab7 and click Open.
- ☐ There's a quick and dirty way to add our CSS, inline, and a better approach that uses a <style> block or a linked style sheet. For the better approaches we need a unique way to identify each of the images. We could give each <img> tag an ID or a unique class name. We will use ID's for this example.
- ☐ Display the ImageFloat.html page and give each <img> an ID. Use any ID as long as it is meaningful to you and anyone else who will need to support this page.
 

```


```
- ☐ In the <head> section add a <style> block just before </head>.
 

```
<style>

</style>
```
- ☐ In the <style> block add a selector for the first ID. ID selectors start with “#”.
 

```
<style>
 #leftFlower { }
</style>
```
- ☐ Add the float property and a value to float it to the left.
 

```
<style>
 #leftFlower {
 float:left;
 }
</style>
```
- ☐ In the <style> block add a selector for the second ID and float it to the right.
- ☐ Save your changes, display the page in the browser and see it floated the images as you expected.

- ☐ You will probably want to add a margin to the right of the first image and a margin to the left of the second image. Experiment with the number of pixels to get the balance right.  
`margin-right:5px;`
- ☐ Save your changes, display the page in the browser and see if it floated the images as you expected. Adjust the margins as needed.

## Exercise 2 – Float DIVs

Browse down the page and note the blocks of text. These are currently in `<div>` tags, and each takes the full width of the page. They have already been assigned the class “floatingDiv”.

We want to display these in boxes from left to right. We could move them into tables, but this is very inflexible and does not adapt to smaller screens. Instead, we can add CSS to display the DIVs as boxes and then float them across the page until we run out of room. As they are just being “floated”, any that don’t fit on the current line will be wrapped to the next line.

- ☐ In the `<style>` block of the ImageFloat.html page add a class selector for “floatingDiv”.  
`.floatingDiv {`  
`}`
- ☐ Make the `<div>`’s look like a box by setting a width. Add a border so you can better see the boxes and a margin to separate them.  
`.floatingDiv {`  
`width:150px;`  
`margin:5px;`  
`border:1px solid;`  
`}`
- ☐ Save your changes and display the page in the browser. Note that a `<div>` by default adds a line break before and after the `<div>`.
- ☐ We need to float the `<div>`’s so they float across the page until there’s no more room.  
`.floatingDiv {`  
`width:150px;`  
`margin:5px;`  
`border:1px solid;`  
`float:left;`  
`}`
- ☐ Save your changes and display the page in the browser. Note that the `<div>`’s now float across the line. Change the width of your browser and watch how the `<div>`’s wrap to the next line as the browser gets smaller.

## Exercise 3 – Exploring the Power of CSS Style Sheets.

There’s no coding in this exercise! Instead we are going to visit a demo site on the internet named “CSS Zen Garden” and play with the styles there.

- ☐ Open a browser and visit <http://www.csszengarden.com/>
- ☐ Scroll down the page and note the general sections of the page:
  - A title area.
  - Links to download the HTML and CSS.
  - Section headings with names like “The Road to Enlightenment”.
  - A menu with options like “Steel” and “A Robot Named Jimmy”
- ☐ Right-click the page and select “View Source”.
- ☐ Note the simplicity of the HTML code!
  - DIV’s
  - HTML 5 DIV-like tags such as <header>, <footer>, <section>, <aside>, etc.
  - Lists of things are defined using unordered lists (<ul>).
  - Most tags have been assigned a class.
  - A few tags have IDs.
  - That’s just about it!
- ☐ Scroll to the top of the code and note that there is a single linked CSS file.  
`<link rel="stylesheet" media="screen" href="/214/214.css?v=8may2013">`  
There’s no inline CSS at all!
- ☐ Return to the browser and click one of the other style sheets such as “Steel”.
  - Slide your mouse over each of the three “things” that are hanging from the top of the page.
  - Notice the animation and 3D effects in the left panel.
- ☐ Right-click and view the source again. Except for the linked CSS filename, all of the HTML is unchanged! All of the changes are from CSS styling, CSS animation and CSS loaded background images.
- ☐ Click “Apothecary” (not the designer’s name). This one may not display 100% as planned in Internet Explorer 11, so also look at in Firefox or Chrome.
- ☐ Again, right-click and view the source. No changes to the HTML, just the CSS.
- ☐ Try each of the designs.
- ☐ If you would like to study any of the designs, create a directory for each design, download the HTML and CSS files and then open them in Visual Studio. Note that the downloads do not include the images.



# Module 7 Lab Solution

## Questions

- ☐ Which of these is the recommended approach to layout a page?
  - ☐ Use <frameset> to layout the master page and load individual HTML documents into each frame.
  - ☐ Use <table> to build a well-defined layout with heading, navigation, content and footer rows and cells.
  - ☐ **Use <div> and div-like HTML 5 tags to define the contextual design and apply CSS to control the display and layout.**

Framesets are obsolete and tables offer no flexibility.

- ☐ When displaying data laid out in rows and columns use:
  - ☐ **<table> with <tr> and <td> tags and CSS**
  - ☐ <div> plus CSS
  - ☐ <ul> or <ol> tags with enclosed <li> tags

For Excel-like displays of data, use tables. They are simple and scale with the data. You could use DIVs and CSS, but that will generally require a lot of work for a simple “table” of data.

- ☐ Which of the following tags are used to build tables.
  - ☐ **<table>**
  - ☐ <thead>
  - ☐ **<thead>**
  - ☐ <tfooter>
  - ☐ **<tfoot>**
  - ☐ <tr>
  - ☐ <cell>
  - ☐ **<td>**
  - ☐ **<th>**
- ☐ You have used float to control text wrapping around <div> and <img> tags. How do you make sure the next block of text starts below all floating object?
  - ☐ <p style="float:none">...
  - ☐ <p style="float:full">...
  - ☐ **<p style="clear:both">...**
  - ☐ <p style="float:all">...

# Module 8 Lab – Images

## Questions

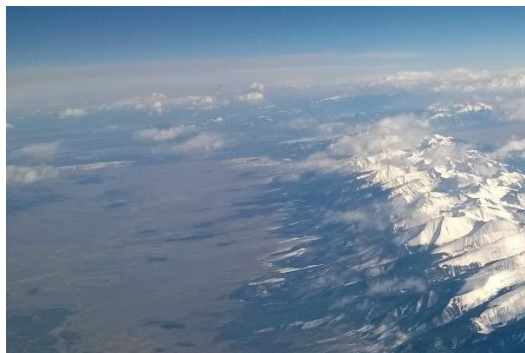
- ☐ The Favicon, if added to the site, is displayed where?
  - ☐ At the top of each page.
  - ☐ In the tabs of many browsers.
  - ☐ In the address bars of many browsers.
  
- ☐ Best practice for images?
  - ☐ Create a master high resolution image and scale as needed by setting the height and width of the `<img>` tag.
  - ☐ Create multiple copies of the image, each pre-scaled for each use.
  
- ☐ Which image type is best for smaller icons with solid areas and only a few colors?
  - ☐ .tiff
  - ☐ .png
  - ☐ .gif
  - ☐ .bmp
  - ☐ .jpg
  
- ☐ Which image type is best for photographs?
  - ☐ .tiff
  - ☐ .png
  - ☐ .gif
  - ☐ .bmp
  - ☐ .jpg
  
- ☐ If you want to set the image using CSS:
  - ☐ Use the `<img>` tag.
  - ☐ Use a `<div>` or other container tag and set the background to the image.
  
- ☐ To treat an image as a clickable map:
  - ☐ Use the `<img>` tag.
  - ☐ Use a `<div>` or other container tag and set the background to the image.
  
- ☐ The best practices for image URLs:
  - ☐ Always use the full path for files on the same site: `http://www.example.com/images/logo.png`

- ☐ Always use the relative path for files on the same site: `/images/logo.png`
  - ☐ Always include the protocol for files on external sites: **http://**`www.somesite.org/image1.png`
  - ☐ Never use the protocol for files on external sites: `www.somesite.org/image1.png`
- 
- ☐ Which tags are needed for an image map?
    - ☐ `<img>`
    - ☐ `<imgmap>`
    - ☐ `<map>`
    - ☐ `<area>`
    - ☐ `<clickablemap>`
- 
- ☐ Background images are added using:
    - ☐ `<background>`
    - ☐ `<backgroundimg>`
    - ☐ `<img>`
    - ☐ `<imgmap>`
    - ☐ Most block tags plus CSS (“background-image:...”)
- 
- ☐ Sprites are:
    - ☐ The same as icons.
    - ☐ Multiple images stored in a single file.
    - ☐ One pixel images.

## Hands on Lab

### Exercise 1 – Add a Background to a Page

Your client for the Toys site is based near the Rockies and would like to have a mountain image used as the page background. They have supplied you with an image named `mountains.jpg`. While it's a pretty picture, it's a bit dark to display text over. You could lighten it using a photo editor. We will leave it as is for now and experiment with changes to the color of the text.



- ☐ On your lab PC or virtual machine launch Visual Studio. If Visual Studio is already open with another project, then click **File, Close Solution**. (If prompted to save a “.sln” file, click **No**.)
- ☐ In Visual Studio click the **File** menu, **Open and Web Site**. In the **Open Web Site** dialog type or navigate to **C:\HTMLCSS\Lab8** and click **Open**.
- ☐ Open the default.html page and view it in a browser. This is the “before”.
- ☐ Return to Visual Studio and open the **TWPIstyles.css** file.
- ☐ In the `<body>` section add the background image.  
`background-image: url('mountains.jpg');`

Before you start typing... try another Visual Studio shortcut!

- Add a blank line just before the closing curly bracket of the “body” section.
  - Move your mouse over the “mountain.jpg” file and then drag the file into that blank line. Magic!
- ☐ Save your changes and display the page in the browser. The image is a little dark. We could switch to white text, or lighten the image.
  - In the “body” section of the style sheet, change the color from black to white.
- ☐ To improve the contrast of the orange heading text you might give all of the headings a 1px or 2px shadow.  
`text-shadow: 1px 1px white;`
- ☐ Save your changes and display the page in the browser. Scroll the page and note that the background moves up and down with the page.
- ☐ Some designers prefer a background that does not scroll with the page. Try a fixed design and see if you like it.  
`background-attachment: fixed;`
- ☐ Save your changes and display the page in the browser. Scroll and note that the background does not move.

## Exercise 2 – Add a Background Image to a DIV

One of the characteristics of the aircraft in the photo is the frequent use of the white stars. One place we could work this into our design is the unused portion of the top `<nav>` area.

- ☐ In Visual Studio double-click the **star.png** image. It’s only 32x35 pixels, so it won’t cover too much of the nav bar... or will it?
- ☐ Select the **TWPIstyles.css** file again. Scroll down to the “nav” selector and add the star as a background image.  

```
nav {
 border-bottom: 1px solid;
 margin-top: 10px;
 padding-bottom: 5px;
 background-image: url(star.png);
}
```

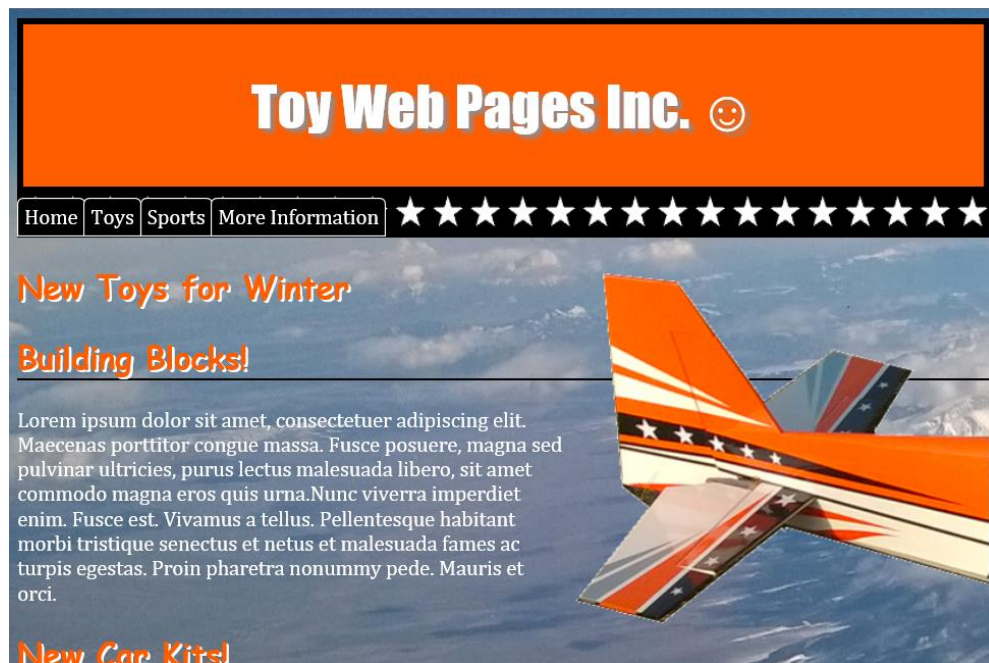
- ❑ Save your changes and display the page in the browser. The stars are there and they repeat to fill the space by default. They actually repeat both horizontally and vertically. As a precaution, in case we change the height of this area in the future, we might want to set the repeat pattern.

```
nav {
 border-bottom:1px solid;
 margin-top:10px;
 padding-bottom:5px;
 background-image:url(star.png);
 background-repeat:repeat-x;
}
```

- ❑ The stars don't quite align ( ☺ ) the way we want them. We need just a little more tweaking. We need to adjust the padding and the height. (Experiment with both of the values if you like.)

```
nav {
 border-bottom:1px solid;
 margin-top:10px;
 /* padding-bottom:5px;*/
 padding-top:10px;
 background-image:url(star.png);
 background-repeat:repeat-x;
 height:28px;
}
```

- ❑ Save your changes and check to see how that looks.
- ❑ As a final tweak, remove the margin-top to remove the gap between the <header> and the <nav> areas.
- ❑ Save your changes and check to see how that looks.



- ❑ You can continue to tweak a project like this for hours! And if you have a few to kill...
- Adjust the margins on the airplane image to get the right edge close to the right edge of the browser.
  - Work on the heading. Experiment with the colors. Maybe add a second shadow to get more of an outline (text-shadow: 2px 2px white, -2px -2px white;) or maybe remove the shadow and add a big blur (text-shadow: 0px 0px 10px white;).
  - Come up with a tricky solution so there's never a part of a star in the background of the <nav> area.
  - Reedit the airplane picture to clean up the artifacts where the colors of the plane and the transparent color meet. (This image is a .png with transparent color selected.)
  - Make the <footer> area look nice. We didn't do anything there.

# Module 8 Lab Solution

## Questions

- ☐ The Favicon, if added to the site, is displayed where?
  - ☐ At the top of each page.
  - ☐ **In the tabs of many browsers.**
  - ☐ **In the address bars of many browsers.**
  
- ☐ Best practice for images?
  - ☐ Create a master high resolution image and scale as needed by setting the height and width of the `<img>` tag.
  - ☐ **Create multiple copies of the image, each pre-scaled for each use.**

Large images can tie up bandwidth and slow page loading. Load the “right sized” image where possible.

- ☐ Which image type is best for smaller icons with solid areas and only a few colors?
  - ☐ .tiff
  - ☐ .png
  - ☐ **.gif**
  - ☐ .bmp
  - ☐ .jpg
  
- ☐ Which image types are best for photographs?
  - ☐ .tiff
  - ☐ **.png**
  - ☐ .gif
  - ☐ .bmp
  - ☐ **.jpg**

.png is both compressed and lossless. .jpg is a close second, but is lossy. .tiff, and .bmp are not widely supported by browsers. .gif only supports 16 or 256 colors.

- ☐ If you want to set the image using CSS:
  - ☐ Use the `<img>` tag.
  - ☐ **Use a `<div>` or other container tag and set the background to the image.**

`<img>` can be changed using JavaScript.

- ☐ To treat an image as a clickable map:

- ☐ **Use the <img> tag.**
- ☐ Use a <div> or other container tag and set the background to the image.
  
- ☐ The best practices for image URLs:
  - ☐ Always use the full path for files on the same site: `http://www.example.com/images/logo.png`
  - ☐ **Always use the relative path for files on the same site: `/images/logo.png`**
  - ☐ **Always include the protocol for files on external sites: `http://www.somesite.org/image1.png`**
  - ☐ Never use the protocol for files on external sites: `www.somesite.org/image1.png`
  
- ☐ Which tags are needed for an image map?
  - ☐ **<img>**
  - ☐ <imgmap>
  - ☐ **<map>**
  - ☐ <area>
  - ☐ <clickablemap>
  
- ☐ Background images are added using:
  - ☐ <background>
  - ☐ <backgroundimg>
  - ☐ <img>
  - ☐ <imgmap>
  - ☐ **Most block tags plus CSS (“background-image:...”)**
  
- ☐ Sprites are:
  - ☐ The same as icons.
  - ☐ **Multiple images stored in a single file.**
  - ☐ One pixel images.

## Hands on Lab

Here’s the final CSS for this lab:

```
body {
 font-size: 18px;
 color: white;
 font-family: Cambria, Cochin, Georgia, Times, Times New Roman, serif;
 background-image: url('mountains.jpg');
 background-attachment: fixed;
}

h1, h2, h3, h4, h5, h6 {
 font-family: "Comic Sans MS", cursive;
 color: rgb(255,93,0) !important;
```



```
/* background:linear-gradient(white, black); */
text-shadow: 2px 2px white;
}

h2 {
 border-bottom-color:black;
 border-bottom-style:solid;
 border-width:2px;
}

h2[style*="italic"] {
 font-style:normal !important;
}

header {
 color:white;
 text-shadow: 4px 4px 4px rgb(149,137,138);
 font-size:48px;
 text-align:center;
 border:5px solid;
 border-color:black;
 background-color:rgb(255,93,0);
 padding:40px;
 font-family:Impact, Haettenschweiler, 'Arial Narrow Bold', sans-serif
}

nav a:hover {
 background-color:lightgray;
}

nav a {
 text-decoration:none;
 border:1px solid;
 border-radius:5px 5px 0 0;
 margin-right:-5px;
 padding:5px;
 color:white;
 background-color:black;
}

nav {
 background-color:black;
 /* margin-top:10px; */
 /* padding-bottom:5px;*/
 padding-top:10px;
 background-image:url(star.png);
 background-repeat:repeat-x;
 height:28px;
```

```
}

aside {
 background: linear-gradient(white 50px, rgb(255,93,0));
}
```

# Module 9 Lab – HTML Forms

## Questions

- ☐ How many `<form>` tags can be in a page?
  - ☐ Only one.
  - ☐ As many as needed.
  
- ☐ To add a multiple lines of text box on a form you would:
  - ☐ Add an `<input>` tag of type “multiline”.
  - ☐ Add an `<input>` tag of type “text” with the “multiple” attribute.
  - ☐ Add a `<textarea>` tag.
  
- ☐ You have a form that will be sending over 5,000 characters of data to a server. Which method should select?
  - ☐ `<form method="GET" ...`
  - ☐ `<form method="PUT" ...`
  - ☐ `<form method="MERGE" ...`
  
- ☐ You have a form that will be sending confidential data to a server. Which method should select?
  - ☐ `<form method="GET" ...`
  - ☐ `<form method="PUT" ...`
  - ☐ `<form method="MERGE" ...`
  
- ☐ You will be sending a part number from a form to a server. Which of the following could you use?
  - ☐ `<input id="partNumber" ...`
  - ☐ `<input name="partNumber" ...`
  - ☐ `<input id="partNumber" name="partNumber" ...`
  - ☐ `<input value="partNumber" ...`
  
- ☐ When using a group of radio buttons, how many buttons can be selected at once?
  - ☐ Zero
  - ☐ One
  - ☐ Two
  - ☐ Many
  
- ☐ When using a group of checkboxes, how many can be selected at once?
  - ☐ Zero

- ☐ One
- ☐ Two
- ☐ Many

## Hands on Lab

### Exercise 1 – Build a Simple Form

This first form will focus on the form elements, and not too much about appearance. You won't see too much sample code here. You will have to figure that out for yourself...

- ☐ On your lab PC or virtual machine launch Visual Studio. If Visual Studio is already open with another project, then click **File, Close Solution**. (If prompted to save a “.sln” file, click **No**.)
- ☐ In Visual Studio click the **File** menu, **Open** and **Web Site**. In the **Open Web Site** dialog type or navigate to **C:\HTMLCSS\Lab9** and click **Open**.
- ☐ Open the **Contact.html** page and review the HTML. It already has the `<form>` tag with the options you will need to send the form data to a test page. You will need to design the rest of the form.
  - This form will need:
    - First name (required) Name on server: firstName
    - Last name (required) Name on server: lastName
    - State (a dropdown list with eight states listed, then form needs to display full state names, but send back just the two letter abbreviations. ) Name on server: state
    - An email address field (required, and must be validated) Name on server: email
    - A phone number field (optional) Name on server: phone
    - A multiline comments box. Name on server: comments
    - A submit and a reset button
  - It should look something like this:



- ☐ Create the form!
- ☐ Test the form.
  - Make sure the form cannot be submitted without supplying the required fields.
  - Make sure that an email address cannot be submitted that does not at least have an “@” and a “.”.
  - After submitting the form to the test page, make sure that the correct names are showing up for each piece of data.

**Optional:**

- ❑ To make the form look nicer, lay it out in a two column table. Something like this:

```
<table>
 <tr>
 <td>First Name:</td>
 <td><input type="text" name="firstName"></td>
 </tr>
 ...
</table>
```

# Module 9 Lab Solution

## Questions

☐ How many `<form>` tags can be in a page?

- ☐ Only one.
- ☐ **As many as needed.**

Some server side technologies may have a limit. For example, ASP.NET Web Forms pages must have one and only one `<form>` that posts back to the same page.

☐ To add a multiple lines of text box on a form you would:

- ☐ Add an `<input>` tag of type “multiline”.
- ☐ Add an `<input>` tag of type “text” with the “multiple” attribute.
- ☐ **Add a `<textarea>` tag.**

☐ You have a form that will be sending over 5,000 characters of data to a server. Which method should select?

- ☐ `<form method="GET" ...`
- ☐ **`<form method="PUT" ...`**
- ☐ `<form method="MERGE" ...`

GET is typically limited to around 2,000 characters of data in the URL query string.

☐ You have a form that will be sending confidential data to a server. Which method should select?

- ☐ `<form method="GET" ...`
- ☐ **`<form method="PUT" ...`**
- ☐ `<form method="MERGE" ...`

GET sends its data as part of the URL for all to see. The URL is stored in the browsers history and can be “replayed”.

☐ You will be sending a part number from a form to a server. Which of the following could you use?

- ☐ `<input id="partNumber" ...`
- ☐ **`<input name="partNumber" ...`**
- ☐ **`<input id="partNumber" name="partNumber" ...`**
- ☐ `<input value="partNumber" ...`

The server receives the form element’s name and its value. ID is for client side use for CSS and JavaScript.

☐ When using a group of radio buttons, how many buttons can be selected at once?

- ☐ **Zero**
- ☐ **One**
- ☐ **Two**
- ☐ **Many**

☐ When using a group of checkboxes, how many can be selected at once?

- ☐ **Zero**
- ☐ **One**
- ☐ **Two**
- ☐ **Many**

## Hands on Lab

Here's the final HTML for the form:

```
<form name="form1" method="GET" action="formresultsdummy.html">
```

```
 First Name: <input type="text" name="firstName" required>

```

```
 Last Name: <input type="text" name="lastName" required >

```

```
 State: <select>
```

```
 <option value="CA">California</option>
```

```
 <option value="FL">Florida</option>
```

```
 <option value="IN">Indiana</option>
```

```
 <option value="KY">Kentucky</option>
```

```
 <option value="MA">Maine</option>
```

```
 <option value="NY">New York</option>
```

```
 <option value="OH">Ohio</option>
```

```
 <option value="VA">Virginia</option>
```

```
</select>

```

```
 Email: <input type="email" name="email" required >

```

```
 Phone number: <input type="tel" name="phone">

```

```
 Comments: <textarea name="comments" rows="5" cols="50"></textarea>

```

```
 <button type="submit">Submit</button>
```

```
 <button type="reset">Reset</button>
```

```
</form>
```

# Module 10 Multimedia

## Questions

- ☐ The <video> tag is part of which versions of HTML?
  - ☐ HTML 3.2
  - ☐ HTML 4.01
  - ☐ XHTML 1.0
  - ☐ HTML 5
  
- ☐ Which of the following are video formats?
  - ☐ WebM
  - ☐ WAV
  - ☐ Ogg/Theora
  - ☐ MP3
  - ☐ MPEG-4/H.264
  
- ☐ Which one of the following has the broadest browser support?
  - ☐ WebM
  - ☐ WAV
  - ☐ Ogg/Theora
  - ☐ MP3
  - ☐ MPEG-4/H.264
  
- ☐ You need for a video to start when the page is loaded, and to continuously play. Which of the following attributes should be added to the <video> tag?
  - ☐ continuous
  - ☐ loop
  - ☐ loopuntilcancel
  - ☐ auto
  - ☐ autoplay
  - ☐ autostart

## Hands on Lab

The team has just gotten back from the country airshow and would like for you to post a brief video of a jet fly by.

### Exercise 1 – Add a Video to the Home Page

We will first need to see if the video is suitable for a web site.



- ☐ Open Windows Explorer and navigate to C:\HTMLCSS\Lab10.
- ☐ Find the Jets.mp4 file. Will this play in most browsers? Or do we need to convert it to a better format?  
\_\_\_\_\_
- ☐ We need the dimensions of the video. Right-click the Jets.mp4 file, click **Properties**, click the **Details** tab and note the height and width of the video.  
Height: \_\_\_\_\_ Width: \_\_\_\_\_
- ☐ On your lab PC or virtual machine launch Visual Studio. If Visual Studio is already open with another project, then click **File, Close Solution**. (If prompted to save a “.sln” file, click **No**.)
- ☐ In Visual Studio click the **File** menu, **Open** and **Web Site**. In the **Open Web Site** dialog type or navigate to C:\HTMLCSS\Lab10 and click **Open**.
- ☐ Open the **Default.html** page and review the HTML. It already has the <form> tag with the options you will need to send the form data to a test page. You will need to design the rest of the form.
- ☐ Between the two sets of <article> tags add a new <article> block similar to this:  
    <article>  
        <h2>Fun at the Airshow!</h2>  
    </article>
- ☐ After the new <h2> tags add the HTML code to display the video.
  - Set the height and width using the values you found earlier.
  - Configure the video tag to display the controls to play the video.
  - Add a message for those users not using an HTML 5 compatible browser.  
    “This video needs an HTML 5 compatible browser.”
- ☐ Save your changes and display the page in the browser. Click the **Play** button and confirm that the video plays.
- ☐ Test the home page in multiple browsers, if available.

## Exercise 2 – Add a Video Thumbnail

The first frame of the video is not very interesting. We would like to display an initial image in the video player that lists the title and date of the video.

- ☐ Open the Windows Paint program.
- ☐ Click **File, Properties**, enter the height and width of the video and click **OK**.
- ☐ Click the **Text** tool and click in the drawing area.
  - Click the font size dropdown and change the font size to **18**.
  - Type “Jet Fly By!”, press **Enter** and type “2/15/2017”
  - Select and drag the text to be roughly centered.
- ☐ Add a few stars to the image!
  - In the **Shapes** area of the ribbon click the five pointed star.
  - Draw a few stars. (Be creative!)
- ☐ Click **File, Properties**, and confirm that the image size is still correct.
- ☐ Save the file to your lab folder as JetsThumb.png. C:\HTMLCSS\Lab10

- ☐ Return to Visual Studio.
- ☐ Right-click the project's name in the Solution Explorer, click Add and Existing Item. Navigate to your new image, click it and click Add.
- ☐ Return to your default.html page and modify the <video> tag to add the "poster" attribute.
- ☐ Save your changes and display the page in the browser. Click the Play button and confirm that the video plays.
- ☐ The result should look something like this:





# Module 10 Lab Solution

## Questions

- ☐ The <video> tag is part of which versions of HTML?
  - ☐ HTML 3.2
  - ☐ HTML 4.01
  - ☐ XHTML 1.0
  - ☐ **HTML 5**
  
- ☐ Which of the following are video formats?
  - ☐ **WebM**
  - ☐ WAV
  - ☐ **Ogg/Theora**
  - ☐ MP3
  - ☐ **MPEG-4/H.264**
  
- ☐ Which one of the following has the broadest browser support?
  - ☐ WebM
  - ☐ WAV
  - ☐ Ogg/Theora
  - ☐ **MP3**
  - ☐ **MPEG-4/H.264**
  
- ☐ You need for a video to start when the page is loaded, and to continuously play. Which of the following attributes should be added to the <video> tag?
  - ☐ continuous
  - ☐ **loop**
  - ☐ loopuntilcancel
  - ☐ auto
  - ☐ **autoplay**
  - ☐ autostart

## Hands on Lab

Code for the video player:

```
<article>
 <h2>Fun at the Airshow!</h2>

 <video width="320" height=178 controls poster="JetsThumb.png">
 <source src="jets.mp4" type="video/mp4">
 Your browser does not support the video tag.
 </video>

</article>
```