HTML MODULE 01
INTRODUCTION
OVERVIEW OF HTML AND CSS

Summer 2021 – Web Development using ASP .Net Core MVC

CONTACT INFO

Instructor	Dr. Razvan Alex. Mezei
Email address	rmezei@stmartin.edu
Office	Panowicz Hall, Room 111
Office Phone	360-688-2748

OFFICE HOURS

- **Drop-In** (no appointment needed!):
  - zoom
- **By Appointment:**
  - Email me in advance to set up an appointment

COURSE OBJECTIVES

- Describe how to create and run a web application using an IDE (such as Visual Studio).
- Create and style HTML5 pages.
- Add interactivity to HTML5 pages using JavaScript.
- Create HTML5 forms and validate user input.
- Create JavaScript code.
- Configure the pipeline of ASP.NET Core web applications by using middleware and leverage dependency injection across MVC applications.
- Add controllers to an MVC application to manage user interaction, update models, and select and return views.
- Develop a web application that uses the ASP.NET Core routing engine to present friendly URLs and a logical navigation hierarchy to users.
- Create views, which display and edit data and interact with models and controllers, in an MVC application.
- Create MVC models and write code that implements business logic within model methods, properties, and events.
- Connect an ASP.NET Core application to a database by using Entity Framework Core.
- Implement a consistent look and feel across an entire MVC web application.

LEARNING OBJECTIVES

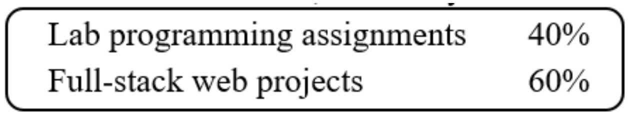
- **Lesson 1. Overview of HTML and CSS**
- **Lesson 2. Creating and Styling HTML Pages**
- **Lesson 3. Introduction to JavaScript**
- **Lesson 4. Creating Forms to Collect and Validate User Input**
- **Lesson 5. Creating Objects and Methods by Using JavaScript**
- **Lesson 6. Creating Interactive Pages by Using HTML5 APIs**

- **Lesson 7. Designing ASP.NET Core MVC Web Applications**
- **Lesson 8. Configure Middleware and Services in ASP.NET Core**
- **Lesson 9. Developing Controllers**
- **Lesson 10. Developing Views**
- **Lesson 11. Developing Models**
- **Lesson 12. Using Entity Framework Core in ASP.NET Core**
- **Lesson 13. Using Layouts, CSS and JavaScript in ASP.NET Core MVC – if time**

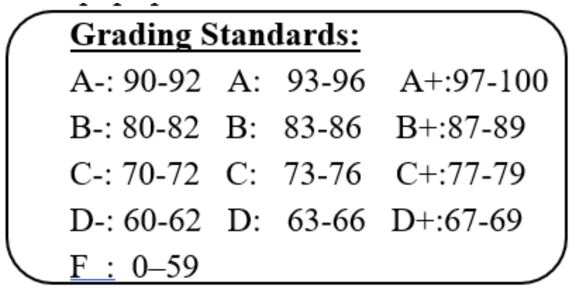
METHODS OF ASSESSMENT

In addition to the ones below, there may be some unannounced pop-quizzes.

- **Unexcused absences may lower your grade!**



Lab programming assignments	40%
Full-stack web projects	60%



Grading Standards:

A-: 90-92	A: 93-96	A+:97-100
B-: 80-82	B: 83-86	B+:87-89
C-: 70-72	C: 73-76	C+:77-79
D-: 60-62	D: 63-66	D+:67-69
F : 0–59		

Programming assignments:

- are due on Sundays!

Full-stack web projects:

- Are due on Mondays!
- You can work alone, OR
- You can pick a colleague to work together on this project. No more than two students per project!

Some homeworks may be due on a holiday – finish them early, deadlines are the due dates, not start dates!

Course Schedule (tentative)

- The dates, schedule, requirements etc. are tentative. The instructor reserves the right to make changes to the course schedule during the duration of the course. Students will be informed, in advance, of any changes and an updated schedule will be distributed and posted to Moodle. Students are responsible for checking Moodle and completing any requirements posted there.

Syllabus Updates and Moodle Listings

- The dates, schedule, requirements etc. are tentative. The instructor reserves the right to make changes to the syllabus during the duration of the course. Students will be informed, in advance, of any changes and an updated syllabus will be distributed and posted to Moodle. Students are responsible for checking Moodle and completing any requirements posted there.

- **Please find a complete copy of the syllabus in moodle**

How to Get Started with Moodle

- Visit: http://moodle.stmartin.edu
- See "Getting Started with Moodle," "Tutorials for Students," and "Frequently Asked Questions" in the pane entitled "NAVIGATION" in the left margin.
- LOGIN with your Saint Martin's username and password when ready. For username and password help, see: https://www.stmartin.edu/directory/integrated-technology-services/technology-help

Standard St. Martin's information

- Helpful Links
  - Link to Saint Martin's Undergraduate Academic Catalog https://www.stmartin.edu/sites/default/files/smu-files/registrar/2020-undergraduate-academic-catalog.pdf
  - Link to Saint Martin's Graduate Academic Catalog https://www.stmartin.edu/sites/default/files/smu-files/registrar/grad-catalog-19-20.pdf
  - Link to Student Handbook currently in use: https://www.stmartin.edu/sites/default/files/smu-files/student-affairs/2020-21-8-24-saint-martins-student-handbook.pdf
  - Link to emergency/weather information: http://www.stmartin.edu/directory/office-public-safety
  - Link to sign up for e2campus emergency alert text messaging: https://www.e2campus.net/my/stmartin/signup.htm
  - Link to Office of Registrar forms: http://www.stmartin.edu/forms?topic=Registrar

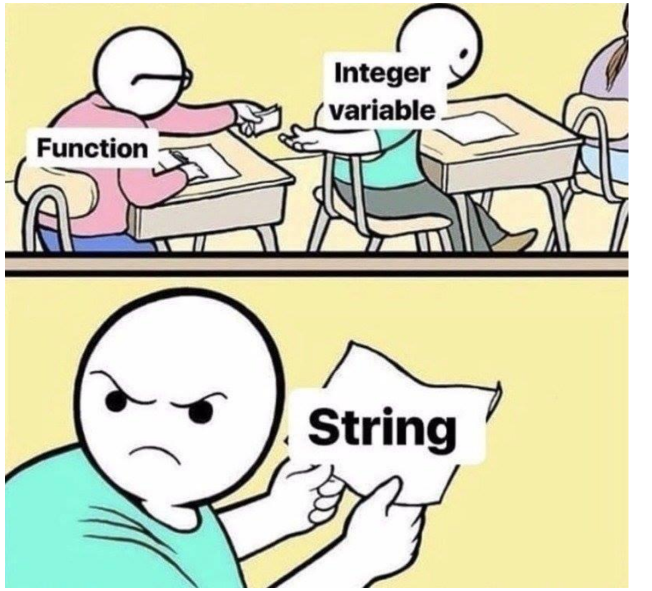
Standard St. Martin’s information

- Saint Martin’s University and the O’Grady Library
  - The O’Grady Library collections and services are available to all Saint Martin’s University students, regardless of physical location. To explore the collections, services and research tools available, visit the O’Grady Library homepage: https://www.stmartin.edu/academics/academic-resources/ogrady-library

- Research assistance
  - **Appointments**: Librarians are available to make either in-person or phone appointments with individuals and/or groups working on research projects. The librarians can do a general tour of the library, orient you to the services and collections available or provide in-depth instruction on using the research tools available for your class. To request a consultation, please use the form: https://www.stmartin.edu/academics/academic-resources/ogrady-library/research/consultations
  - **Chat service**: The library also partners with a nationwide group of librarians to offer a chat service 24 hours a day, 7 days a week. Working late on a paper and not sure where to start? The chat service librarians can access almost all of the library’s resources and can often help you get started on a project or help you figure out if something is available. To start a chat, go to: http://tinyurl.com/dyaew23
  - **Research guides**: The library also has a series of research guides that pull together key databases, reference sources and websites that are particularly useful in a subject area. To explore the research guides, go to: http://stmartin.libguides.com/
  - **How do I…? tutorials**: The O’Grady Library has a collection of “how to” tutorials designed to help you get started on research at the library. To explore the tutorials, go to http://stmartin.libguides.com/tutorials/

- **Administrative Policies/Support**
  - Access and Accommodations
    - Your experience in this class is important to me. If you have already established accommodations with Disability Support Services for Students (DSS), please communicate your approved accommodations to me at your earliest convenience so we can discuss your needs in this course.
    - If you have not yet established services through DSS, but have a temporary health condition or permanent disability that requires accommodations (conditions include but not limited to; mental health, attention-related, learning, vision, hearing, physical or health impacts), you are welcome to contact DSS at 360-438-4580 or dss.testing@stmartin.edu or smu.dss@stmartin.edu DSS offers resources and coordinates reasonable accommodations for students with disabilities and/or temporary health conditions. Reasonable accommodations are established through an interactive process between you, your instructor(s) and DSS. It is the policy and practice of the Saint Martin’s University to create inclusive and accessible learning environments consistent with federal and state law

- Academic Honesty/Professionalism
  - **WHAT IS ACADEMIC INTEGRITY?** Saint Martin’s University is a community of faculty, students and staff engaged in the exchange of ideas in the ongoing pursuit of academic excellence. Essential to our mission is a focused commitment to scholarly values, intellectual integrity and a respect for the ideas, beliefs and work of others. This commitment extends to all aspects of academic performance. All members are expected to abide by ethical standards both in their conduct and their exercise of responsibility to themselves and toward other members of the community. As an expression of our shared belief in the Benedictine tradition, we support the intellectual, social, emotional, physical and spiritual nurturing of students.
  - **WHAT IS ACADEMIC DISHONESTY?** Saint Martin’s University defines Academic Dishonesty as violating the academic integrity of an assignment, test and/or evaluation of any coursework. This dishonest practice occurs when you seek to gain for yourself or another, an academic advantage by deception or other dishonest means. You have a responsibility to understand the requirements that apply to particular assessments and to be aware of acceptable academic practice regarding the use of material prepared by others. Therefore, it is your responsibility to be familiar with the policies surrounding Academic Dishonesty as these may differ from other institutions.

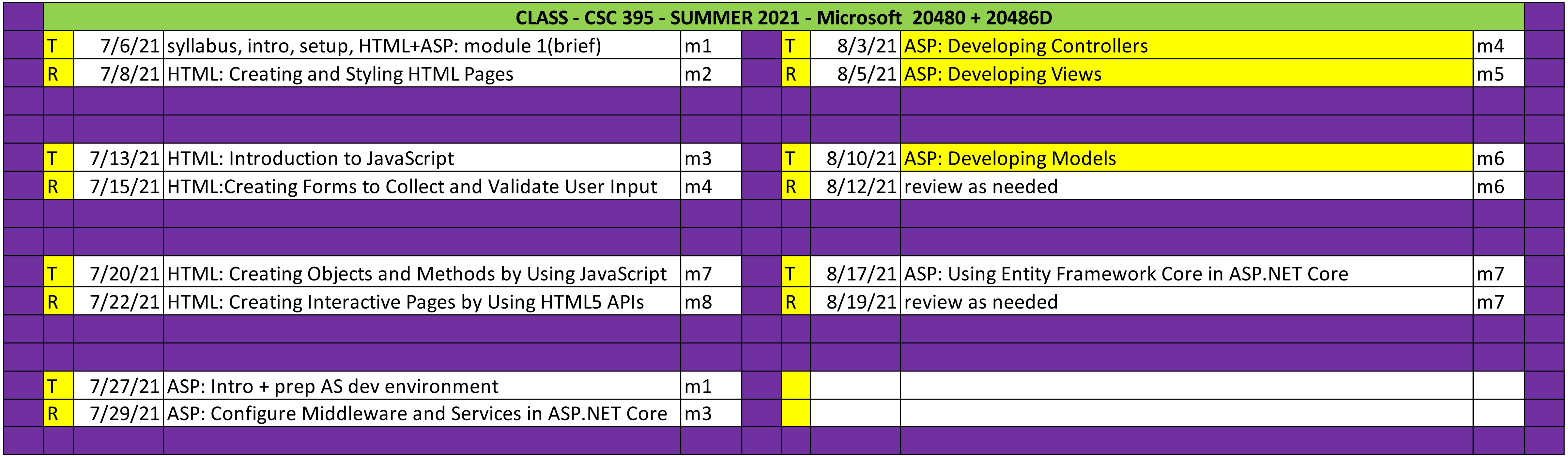


- University Sanctioned Activities
  - If you are absent from class due to university sanctioned activities, such as sports, it is your responsibility to request that the absence be excused, otherwise, the absence will be recorded as unexcused. Absent students are responsible for catching up with the class, and if any assignments are due on the day of the absence, it is your responsibility to turn in the assignments on time (prior to class). Assignments may be submitted as an attachment to email: rmezei@stmartin.edu. Please request the policy handout, “Requirement for receiving Excused Absence” on the first day of the class if you think this policy might apply to you.

- Center for Learning, Writing, and Advising
  - The Center for Student Learning, Writing and Advising offers free academic services for all Saint Martin’s students at all levels of achievement in pursuit of intellectual growth and academic excellence. The Learning Center is home to the STEM Study Center which provides subject area peer tutoring (science, technology, engineering, and math as well as business/ accounting/economics, and world languages). At the Writing Center, students meet with trained peer readers to discuss their academic, personal and professional writing. The Advising Center works with students with academic advising, connecting with campus support resources, transition and self-exploration guidance, personalized academic improvement plans, learning workshops, and support major change. The Advising Center staff also works closely with the University’s Early Alert Program — a referral system that supports student success. Saint Martin’s Disability Support Services is located in the Center for any student with a disability who is interested in using their accommodations. These students can connect with the Disability Support Services Coordinator who will evaluate the documentation, determine appropriate accommodations, and serve as a learning resource and advocate with assisting students in meeting their academic goals. https://www.stmartin.edu/academics/academic-resources/center-student-learning-writing-and-advising

- Counselling and Wellness Center
  - The Counseling and Wellness Center (CWC) is committed to helping you meet the challenges of life you may experience during college. The CWC promotes and enhances the health and development of students through professional mental health services, education and training. Integrating faith, reason and service, we empower you to develop self-awareness, knowledge and skills, necessary to make healthy choices and build relationships in a multicultural world. Integrating faith, reason and service, the CWC empowers students to develop self-awareness, knowledge and the skills necessary to make healthy choices and build relationships in a multicultural world. https://www.stmartin.edu/directory/counseling-and-wellness-center
- Sexual Misconduct/Sexual Harrassment Reporting
  - Saint Martin's University is committed to providing an environment free from sex discrimination, including sexual harassment and sexual violence. There are Title IX/sexual harassment posters around campus that include the contact information for confidential reporting and formal reporting. Confidential reporting is where you can talk about incidents of sexual harassment and gender-based crimes including sexual assault, stalking, and domestic/relationship violence. This confidential resource can help you without having to report your situation to the formal reporting process through the Dean of Students – Ms. Melanie Richardson, Associate VP of Human Resources – Ms. Cynthia Johnson, Public Safety – Mr. Will Stakelin, or the Office of the Provost – unless you request that they make a report. Please be aware that in compliance with Title IX and under the Saint Martin's University policies, educators must report incidents of sexual harassment and gender-based crimes including sexual assault, stalking, and domestic/relationship violence.
  - If you disclose any of these situations in class, on papers, or to me personally, I am required to report it.

TENTATIVE SCHEDULE



CLASS - CSC 395 - SUMMER 2021 - Microsoft 20480 + 20486D								
T	7/6/21	syllabus, intro, setup, HTML+ASP: module 1(brief)	m1		T	8/3/21	ASP: Developing Controllers	m4
R	7/8/21	HTML: Creating and Styling HTML Pages	m2		R	8/5/21	ASP: Developing Views	m5
T	7/13/21	HTML: Introduction to JavaScript	m3		T	8/10/21	ASP: Developing Models	m6
R	7/15/21	HTML:Creating Forms to Collect and Validate User Input	m4		R	8/12/21	review as needed	m6
T	7/20/21	HTML: Creating Objects and Methods by Using JavaScript	m7		T	8/17/21	ASP: Using Entity Framework Core in ASP.NET Core	m7
R	7/22/21	HTML: Creating Interactive Pages by Using HTML5 APIs	m8		R	8/19/21	review as needed	m7
T	7/27/21	ASP: Intro + prep AS dev environment	m1					
R	7/29/21	ASP: Configure Middleware and Services in ASP.NET Core	m3					

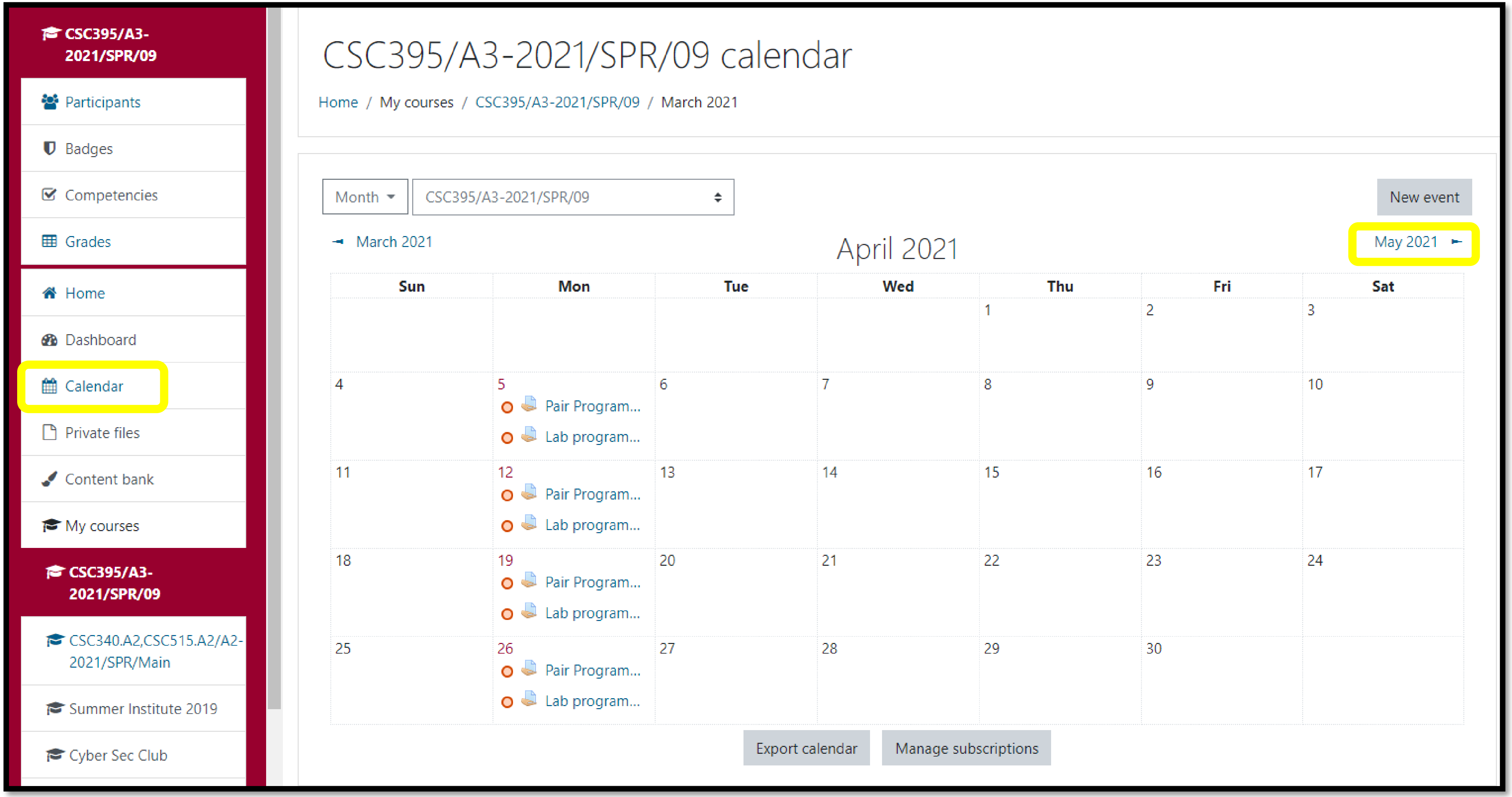
MOODLE NAVIGATION FOR THIS COURSE

- http://moodle.stmartin.edu/

Saint Martin's
UNIVERSITY

moodle

CALENDAR



CSC395/A3-2021/SPR/09

Participants

Badges

Competencies

Grades

Home

Dashboard

Calendar

Private files

Content bank

My courses

CSC395/A3-2021/SPR/09

CSC340.A2,CSC515.A2/A2-2021/SPR/Main

Summer Institute 2019

Cyber Sec Club

CSC395/A3-2021/SPR/09 calendar

Home / My courses / CSC395/A3-2021/SPR/09 / March 2021

Month ▾ CSC395/A3-2021/SPR/09 ⬍ New event

← March 2021 April 2021 May 2021 →

Sun	Mon	Tue	Wed	Thu	Fri	Sat
				1	2	3
4	5 ○ Pair Program... ○ Lab program...	6	7	8	9	10
11	12 ○ Pair Program... ○ Lab program...	13	14	15	16	17
18	19 ○ Pair Program... ○ Lab program...	20	21	22	23	24
25	26 ○ Pair Program... ○ Lab program...	27	28	29	30	

Export calendar Manage subscriptions

LET'S BEGIN

- In here we'll learn about **Web Applications**

- In particular we'll focus on:
  - [Part of] The Microsoft 20-480 curriculum for: **Programming in HTML5 with JavaScript and CSS3**
  - [Part of] The Microsoft 20-486 curriculum for: **ASP .NET Core MVC framework**

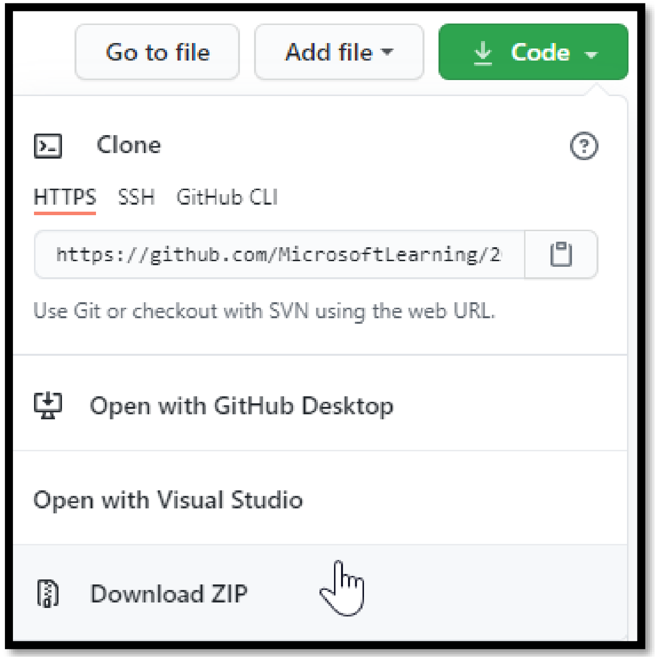
RELATED MICROSOFT COURSES

- 20480C: Programming in HTML5 with JavaScript and CSS
- 20483C: Programming in C#
- 20486D: Developing ASP.NET Core MVC Web Applications

WORK ASSIGNMENTS

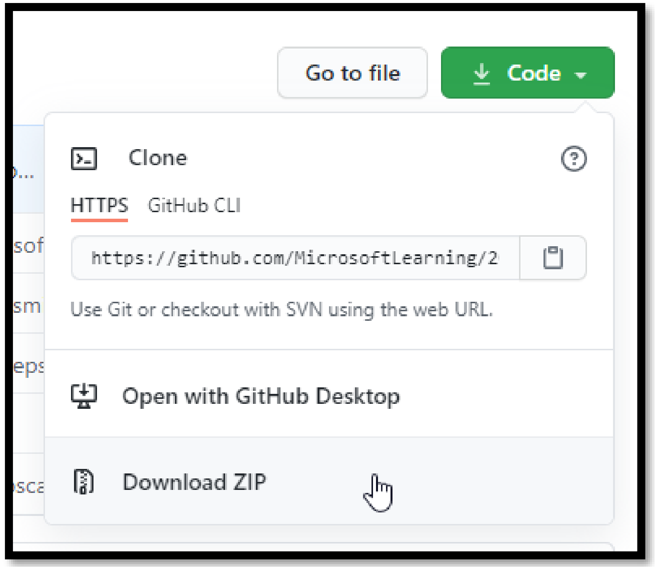
- ## Weekly Homework Assignments
  - please consider working in teams of two, so you can help each other
  - https://github.com/MicrosoftLearning/20480-Programming-in-HTML5-with-JavaScript-and-CSS3
    ^^^ source for homework projects (**first 3 weeks**)
  - https://github.com/MicrosoftLearning/20486D-DevelopingASPNETMVCWebApplications
    ^^^ source for homework projects (**next 4 weeks**)

- ## Full-stack web projects
  - Please create your very own Web application/Project.
  - As we go along, each week we'll learn new interesting topics. Apply these to your **Full-stack web projects.**



https://github.com/MicrosoftLearning/2

Use Git or checkout with SVN using the web URL.



https://github.com/MicrosoftLearning/2

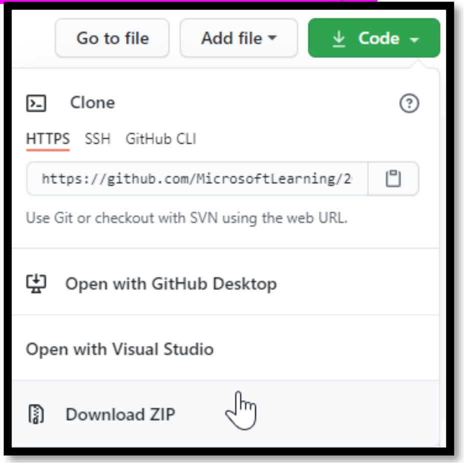
Use Git or checkout with SVN using the web URL.

TEXTBOOK(S) INFORMATION

- Here are the main (free!) resources I will use:
- HTML
  - https://www.w3schools.com/ ←amazing resource for HTML, CSS, JavaScript, jQuery, and Bootstrap
    - https://www.w3schools.com/html/default.asp ← HTML Tutorial
    - https://www.w3schools.com/css/default.asp ← CSS Tutorial
    - https://www.w3schools.com/js/default.asp ← JavaScript Tutorial
  - https://developer.mozilla.org/en-US/docs/Learn ← Another great resources for Web Development
  - https://jsfiddle.net/ ← great for testing HTML+CSS+JavaScript
  - https://eloquentjavascript.net/ ← free JavaScript ebook
    - https://eloquentjavascript.net/Eloquent_JavaScript.pdf ← the PDF version
- ASP .NET Core MVC
  - https://docs.microsoft.com/en-us/aspnet/core/mvc/overview?view=aspnetcore-5.0 ←for “textbook”
  - https://docs.microsoft.com/en-us/learn/certifications/courses/20486-d ←as guide

PREPARING THE DEV ENVIRONMENT & HOMEWORK ENVIRONMENT (FOR HTML)

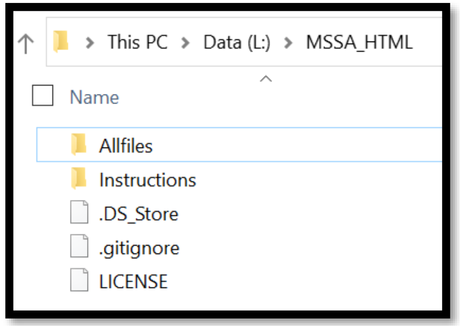
- **Step 1**: download the instructions and code from:
  - https://github.com/MicrosoftLearning/20480-Programming-in-HTML5-with-JavaScript-and-CSS3
  - Note: this is a little large file!



https://github.com/MicrosoftLearning/2

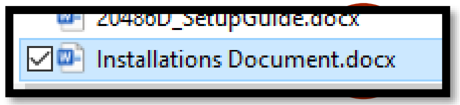
Use Git or checkout with SVN using the web URL.

- **Step2**: Unzip the file into a directory with a short path



Allfiles
Instructions
.DS_Store
.gitignore
LICENSE

- **Step3**: The **Instructions** subdirectory contains installation steps
  - Follow the steps show in **Installations Document for 20486D**
    - This installation covers what you'll need for both parts of this course.
  - Useful links:
    - https://www.visualstudio.com/downloads/ Visual Studio Community 2017/2019
    - https://go.microsoft.com/fwlink/?linkid=867670 SQL server management
    - https://nodejs.org/en Node.js



20486D_SetupGuide.docx
☑ Installations Document.docx

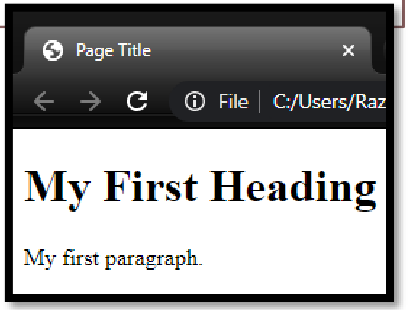
MODULE 1
OVERVIEW OF HTML AND CSS

WHAT IS HTML?

```
<!DOCTYPE html>
<html>
        <head>
                <title>Page Title</title>
        </head>
        <body>

                <h1>My First Heading</h1>
                <p>My first paragraph.</p>

        </body>
</html>
```

- **HTML**, which stands for **H**yper **T**ext **M**arkup **L**anguage, describes the structure of a Web page.
- It consists of a series of elements that tell a web browser how to display the content.

- The <!DOCTYPE html> declaration specifies that this document is an **HTML5** document
- The <html> **element** is the root element of a page
- The <head> **element** contains meta information about a page
- The <title> **element** specifies a title for the page (shown in the page's tab)
- The <body> **element** defines the body, a container for all the visible contents, such as headings, paragraphs, images, etc.
  - The <h1> **element** defines a large heading
  - The <p> **element** defines a paragraph



My First Heading

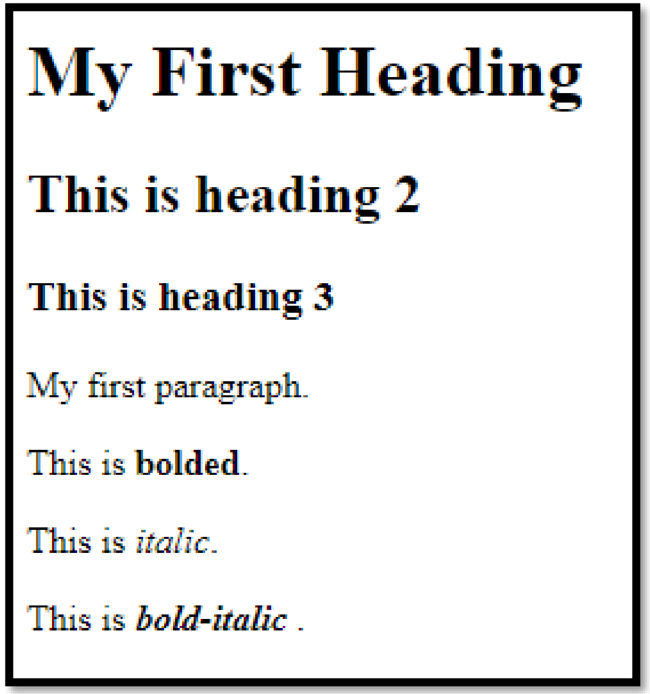
My first paragraph.

- An HTML element is content enclosed by a start tag (for example: <p>) and an end tag (for example: </p>).
  - Elements can be nested.
  - Some elements (such as <br>) have no content. Such elements are called empty elements and they have no end tag.
  - **HTML is Not Case Sensitive**

- Source: https://www.w3schools.com/html/html_intro.asp

LET'S CREATE A FIRST HTML PAGE (SKIP)

```
<!DOCTYPE html>

<html>
  <head>
        <title>Page Title</title>
  </head>
  <body>

     <h1>My First Heading</h1>
     <h2>This is heading 2</h2>
     <h3>This is heading 3</h3>
     <p>My first paragraph.</p>
     <p>This is <strong> bolded</strong>. </p>
     <p>This is <em>italic</em>.</p>
     <p>This is <strong> <em>bold-italic</em> </strong>.</p>

  </body>

</html>
```



My First Heading

This is heading 2

This is heading 3

My first paragraph.

This is **bolded**.

This is *italic*.

This is ***bold-italic*** .

WHAT ARE HTML ATTRIBUTES?

- HTML **elements** can have attributes - used to provide additional information about elements
  - **Attributes** are always specified in the start tag and usually come in name/value pairs such as: name="value"

- For example:
  - <a href="https://www.w3schools.com">Visit the page W3Schools</a>
  - the above line is an **anchor** element
  - the <a> is the start tag
  - the </a> is the start tag
  - Visit the page W3Schools is the content of this element
  - href is an attribute
  - "https://www.w3schools.com" is the value of the attribute

- Other examples:
  - <img src="img_typo.jpg" width="500" height="600" alt="Girl with a jacket">
  - <p style="color:red;">This is a red paragraph.</p>
  - <p title="I'm a tooltip">This is a paragraph.</p>
  - In the above examples, identify the **element**, **tags**, **attributes**, **content**

- Source: https://www.w3schools.com/html/html_attributes.asp ← check this out for more attributes!

HTML LINKS

- Hyperlinks (HTML Links) allow users to move from one page to another (or even within a page).

- For example:
  - <**a** href="https://www.google.com">Search on Google<**/a**>
  - <a> tag defines a hyperlink
  - href attribute indicates the link's destination.

- User the **target** attribute to specify where to open the linked document. Possible values for this attribute:
  - _self – this is the **default**, it opens the document in the same window as it was clicked
  - _blank - opens the document in a new window or tab
  - _parent - opens the document in the parent frame
  - _top - opens the document in the full body of the window
  - <a href="https://www.google.com" target="_blank">Search on Google</a>

- Use mailto: inside the href attribute to create a link that opens the user's email program:
  - <a href="mailto:someone@example.com">Send email</a>

- One can also use an **image** as a **link**
  - <a href="default.asp"> <img src="smiley.gif" alt="HTML tutorial" style="width:42px;height:42px;"> </a>

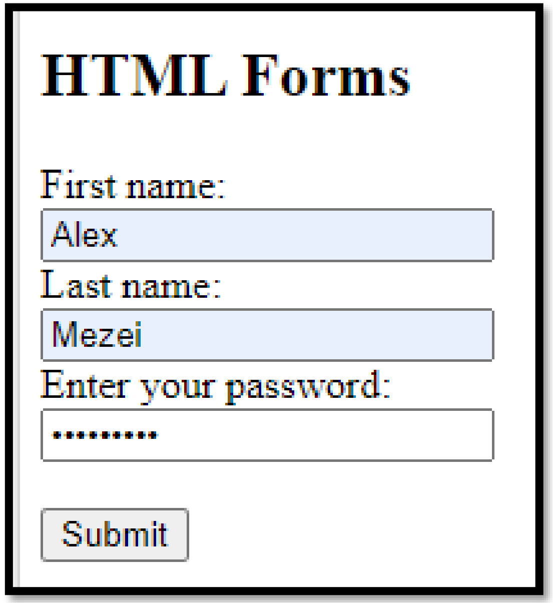
- Source: https://www.w3schools.com/html/html_links.asp
  - On your own please read about Buttons … <button onclick="do something"> HTML Button </button>

HTML FORMS

- Use the **<form> element** to create an HTML form for user input.
  - The <form> element is a container for multiple input elements, such as: text fields, checkboxes, submit buttons, etc.
  - Note: the form itself is not visible!
  - Note: the default width of an input field is 20 characters

- Example:

```
<form action="/login_page.php">

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

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

    <label for="pass">Enter your password:</label><br>
    <input type="password" id="pass" name="pass"><br><br>

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

</form>
```



HTML Forms

First name:

Alex

Last name:

Mezei

Enter your password:

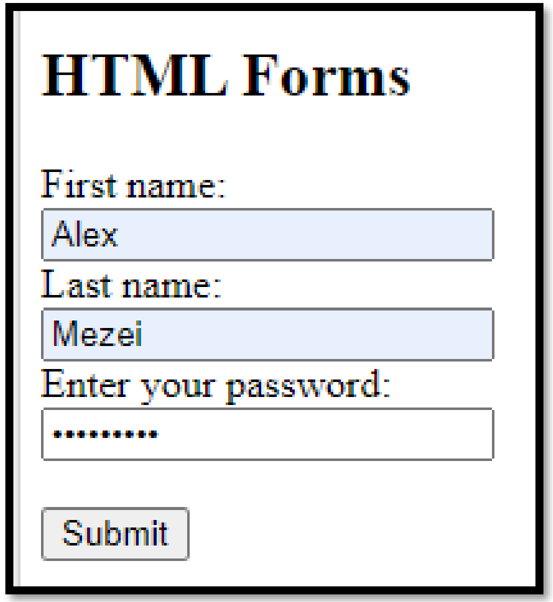
•••••••••

- Note: if the **name** attribute is omitted, the value of that input field will not be sent at all.
- Note: the for attribute of the <label> tag must equal to the id attribute of the <input> element to bind them together.

- Source: https://www.w3schools.com/html/html_forms.asp ← see also radio buttons, and checkboxes!

THE ACTION AND METHOD ATTRIBUTES

- Example:
  - <form action="/login_page.php">
  -   <label for="fname">First name:</label><br>
  -   <input type="text" id="fname" name="fname" ><br>
  -   <label for="lname">Last name:</label><br>
  -   <input type="text" id="lname" name="lname" ><br>
  -   <label for="pass">Enter your password:</label><br>
  -   <input type="password" id="pass" name="pass"><br><br>
  -   <input type="submit" value="Submit">
  - </form>



HTML Forms

First name:
Alex

Last name:
Mezei

Enter your password:
•••••••••

- The action attribute defines the URL of the action to be called when the form is submitted.
  - GET: appends name/value pairs to the **URL**, therefore not secure, useful if the user wants to bookmark searches
  - POST: appends form data inside **body** of the request, not in the URL, has no size limitation

- The method attribute specifies the HTTP method used when the form data is submitted.
  - <form action="/login_page.php" method="get"> (test it!)
  - <form action="/login_page.php" method="post"> (test it!)

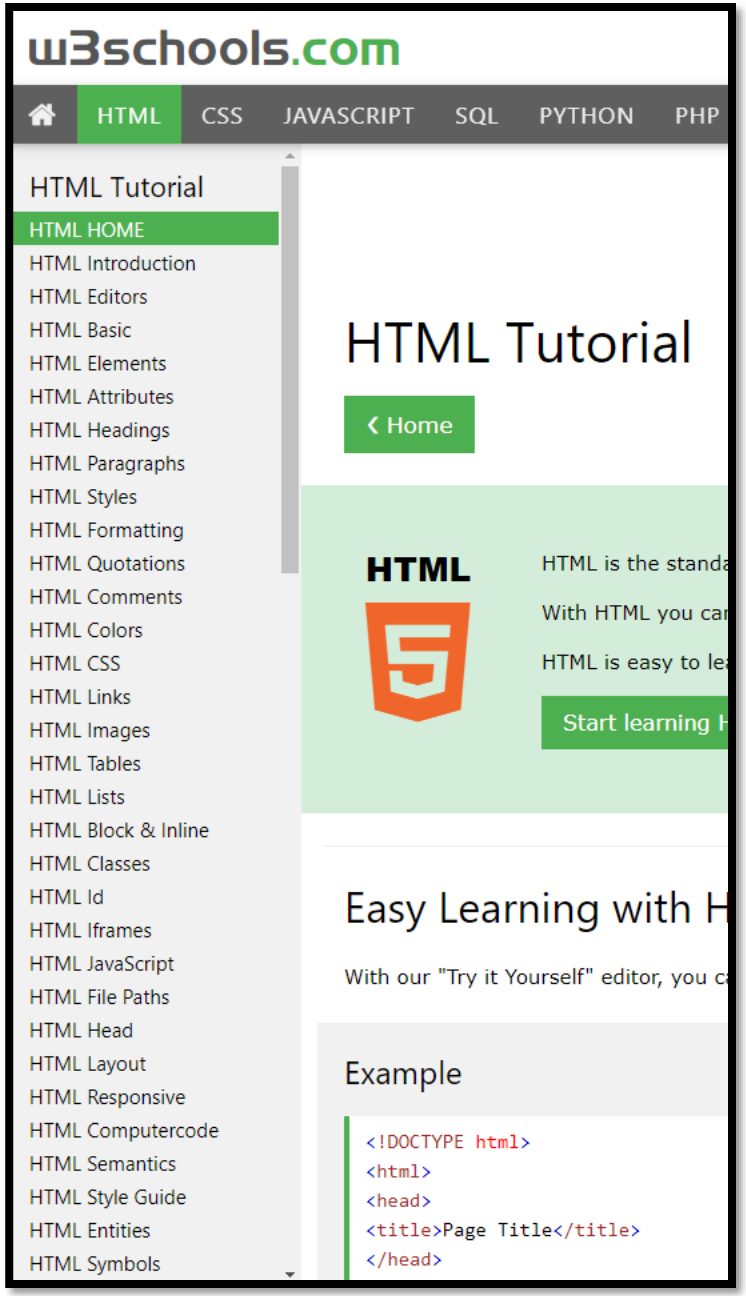
- Source: https://www.w3schools.com/html/html_forms_attributes.asp ← check out autocomplete

HTML

- In this course, you will be expected to
  - either know these elements or
  - be able to research them as needed.

- Use this site as a great guide:
  - https://www.w3schools.com/html/default.asp

- For example:
  - <ul>
  - <li>Computer Science</li>
  - <li>Engineering</li>
  - <li>Mathematics</li>
  - </ul>



HTML Tutorial

HTML is the standa

With HTML you car

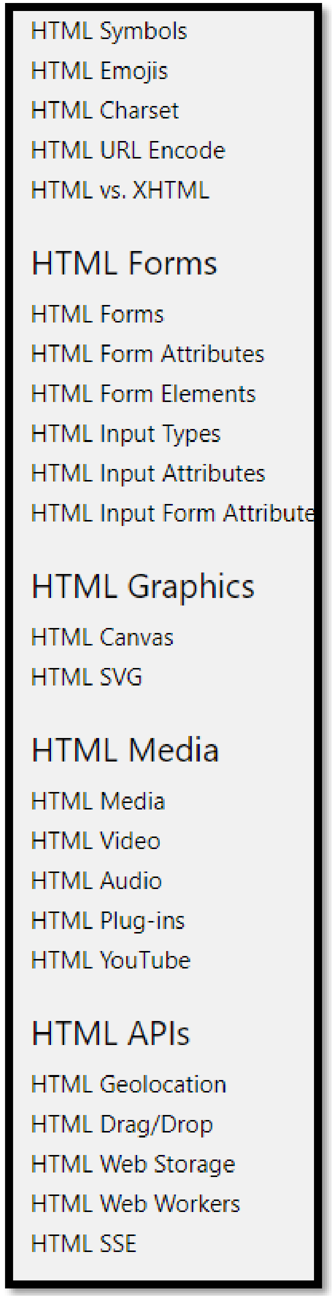
HTML is easy to le

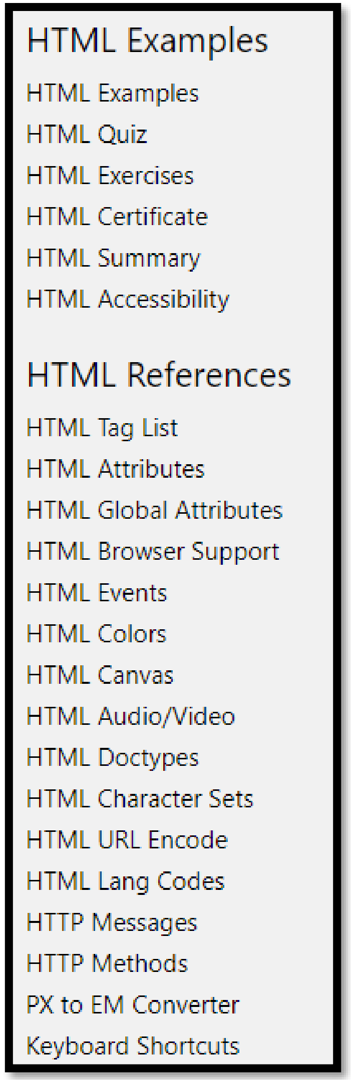
Easy Learning with H

With our "Try it Yourself" editor, you c

Example

```
<!DOCTYPE html>
<html>
<head>
<title>Page Title</title>
</head>
```





IN-CLASS DEMO

Demonstration: Creating a Simple Contact Form

- Source/Steps
- https://github.com/MicrosoftLearning/20480-Programming-in-HTML5-with-JavaScript-and-CSS3/blob/master/Instructions/20480C_MOD01_DEMO.md

HTML JAVASCRIPT

- JavaScript helps make HTML pages more dynamic and interactive.

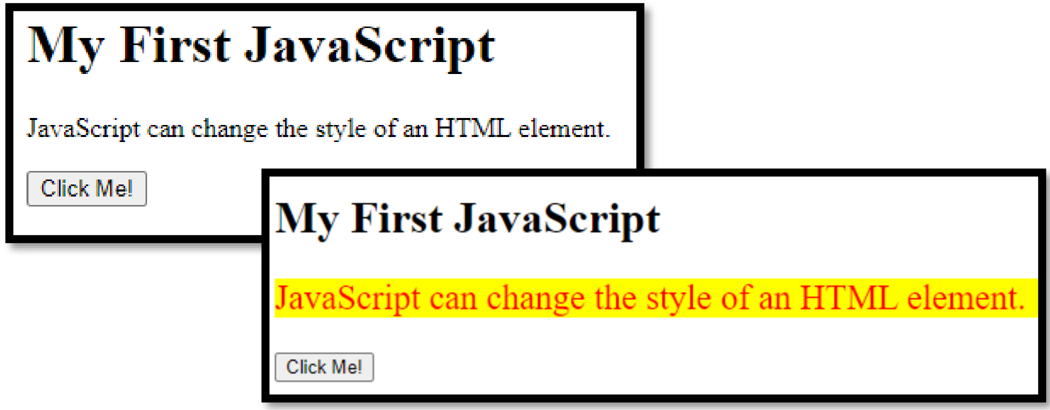
- Use <script> tag to define a client-side script.

- The method getElementById() "finds" an HTML element (here, it searched for element with id="demo").
  - In our example we change its style (font size, color, …).

- Use <noscript> tag to define an alternate content when scripts are not supported or disabled
  - See here an example.

- Sources:
  - https://www.w3schools.com/html/html_scripts.asp
  - https://www.w3schools.com/js/default.asp
  - https://www.w3schools.com/js/js_intro.asp

```
<h1>My First JavaScript</h1>

<p id="demo">JavaScript can change the style of an HTML element.</p>

<script>
function myFunction() {
  document.getElementById("demo").style.fontSize = "25px";
  document.getElementById("demo").style.color = "red";
  document.getElementById("demo").style.backgroundColor = "yellow";
}
</script>

<button type="button" onclick="myFunction()">Click Me!</button>
```



My First JavaScript

JavaScript can change the style of an HTML element.

My First JavaScript

JavaScript can change the style of an HTML element.

CASCADING STYLE SHEETS (CSS)

- CSS used to style HTML pages (it describes how HTML elements should be displayed).
  - HTML was NOT intended to contain tags for formatting a web page
  - The following is possible, but not encouraged: <p style="color:red;"> some text </p>

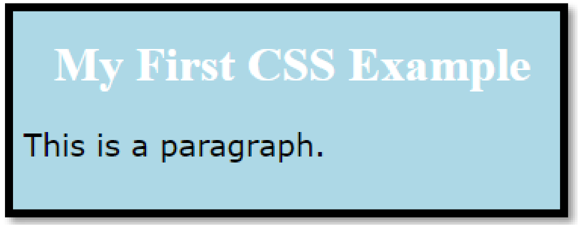
- Typically, we use external .css files.
  - By changing one such file, one can change the look of an entire website.
  - Use the <link> element to reference an external style sheet:
    <link rel="stylesheet" type="text/css" href="myCSSfile.css">
  - We can also add a <style> element into a page's head element to include page-specific styles ➔ ➔ ➔

- CSS Syntax:
  - **selector** {property1:value1;   property2:value2; }
  - use a CSS selector to select HTML element(s) you want to style.

- Example:
  - p { color: red;   text-align: center; }
  - In this example, all **<p> elements** will be **center-aligned**, with a **red text color**:

- Sources:
  - https://www.w3schools.com/css/default.asp
  - https://www.w3schools.com/css/css_intro.asp

```
<style>
body {
  background-color: lightblue;
}

h1 {
  color: white;
  text-align: center;
}

p {
  font-family: verdana;
  font-size: 20px;
}
</style>
</head>


<body>

<h1>My First CSS Example</h1>
<p>This is a paragraph.</p>

</body>
```



My First CSS Example

This is a paragraph.

CSS SELECTORS (1)

- use CSS selectors to select which HTML element(s) you want to style.
  - --
- CSS element selector: example: p { color: red; text-align: center; } if time see also
  - selects HTML elements based on the element name.
  - in this example, it selects all **<p> elements** (all paragraphs). For example: <p>Hello World!</p>
  - --
- CSS id selector: example: #par1 { color: red; text-align: center; } if time see also
  - uses the id attribute of an HTML element to select a specific (only one!) element.
  - since the **id** of an element is unique within a page, the id selector is used to select one unique element!
  - in this example, we assume some element has the id set to par1, for example: <p id="par1">Hello World!</p>
  - --
- CSS class selector: example: .par2 { color: red; text-align: center; } if time see also
  - uses the class attribute of an HTML element to select specific (potential several!) element(s).
  - since the **id** of an element is unique within a page, the id selector is used to select one unique element!
  - in this example, we assume some elements have the class set to par2. Example: <p class="par2">Hello World!</p>
  - --
- Source: https://www.w3schools.com/css/css_selectors.asp

CSS SELECTORS (2)

- One can specify which specific HTML elements of a given class should be affected.
  - example: .par2 { color: red; text-align: center; } ← selects all elements that have class="par2"
  - example: h1.par2 { color: red; text-align: center; } ← only h1 elements that have class="par2"
- --
- Note: one element can be assigned to more than one class.
  - example: <p class="center large">Hello World!</p> ← this element has two classes "center" and "large"
- --
- CSS universal selector: example: * { color: red; text-align: center; } if time see also
  - selects all elements on the page.
- --
- CSS grouping selector: example: h1, h2, p { color: red; text-align: center; } if time see also
  - selects all specified elements on the page.
  - in the example above, all h1, all h2, and all p elements are given the same style definitions.
- --
- BEWARE of punctuation! example: section h2 { color: red; text-align: center; }
  - in the example above, it selects all h2 elements nested inside a section element.
- --
- Source: https://www.w3schools.com/css/css_selectors.asp

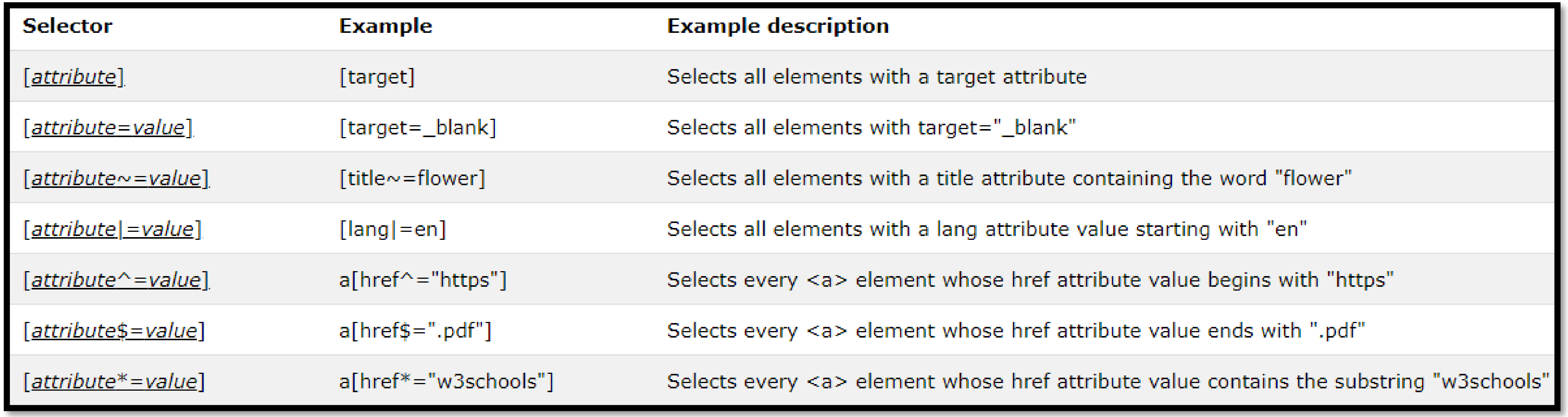
CSS COMBINATORS - SKIP

- A combinator can be used to explain the relationship between the selectors
  - descendant selector (space) ← children, children of children, … see example here
  - child selector (>) ← only direct children … see example here
  - adjacent sibling selector (+) ← one immediately follows the other, same parent see example here
  - general sibling selector (~) ← all elements that are siblings of a given element see example here
- --
- CSS descendant selectors: example: section h2 { color: red; text-align: center; }
  - in the example above, it selects all h2 elements nested inside a section element.
  - in the example above, we are searching for h2 elements that are descendants/inside of a section element)
- --
- Source: https://www.w3schools.com/css/css_combinators.asp

CSS ATTRIBUTE SELECTORS – SKIP

- Used to select HTML elements that have specific attributes or attribute values.

- CSS [attribute] selector: example: a[target] { color: red; text-align: center; } if time see also
  - selects all specified elements on the page that have the specified attribute.
  - in the example above, all a elements that use an attribute called target.
- ---
- CSS [attribute="value"] selector: example: a[target="_blank"] { color: red; text-align: center; } if time see also
  - selects all specified elements on the page that have the specified attribute and value
  - in the example above, all a elements that use a target="_blank" attribute
- ---
- Please check out the other ones too …



Selector	Example	Example description
[attribute]	[target]	Selects all elements with a target attribute
[attribute=value]	[target=_blank]	Selects all elements with target="_blank"
[attribute~=value]	[title~=flower]	Selects all elements with a title attribute containing the word "flower"
[attribute|=value]	[lang|=en]	Selects all elements with a lang attribute value starting with "en"
[attribute^=value]	a[href^="https"]	Selects every <a> element whose href attribute value begins with "https"
[attribute$=value]	a[href$=".pdf"]	Selects every <a> element whose href attribute value ends with ".pdf"
[attribute*=value]	a[href*="w3schools"]	Selects every <a> element whose href attribute value contains the substring "w3schools"

- Source: https://www.w3schools.com/css/css_attribute_selectors.asp

SKIPPED

- --
- CSS Pseudo-**elements**: https://www.w3schools.com/css/css_pseudo_elements.asp
  - select and style a part of an element
  - for example, they can be used to:
    - Style the first letter, or line, of an element
    - Insert content before, or after, the content of an element
  - selector::pseudo-element { property: value; }
  - Example: p::first-line { color: #ff0000; font-variant: small-caps; }
- --
- CSS Pseudo-**classes**: https://www.w3schools.com/css/css_pseudo_classes.asp
  - select elements based on a certain state
  - for example, it can be used to:
    - Style an element when a user mouses over it
    - Style visited and unvisited links differently
    - Style an element when it gets focus
  - selector:pseudo-class { property: value; }
  - Example: a.highlight:hover { color: #ff0000; }
- --

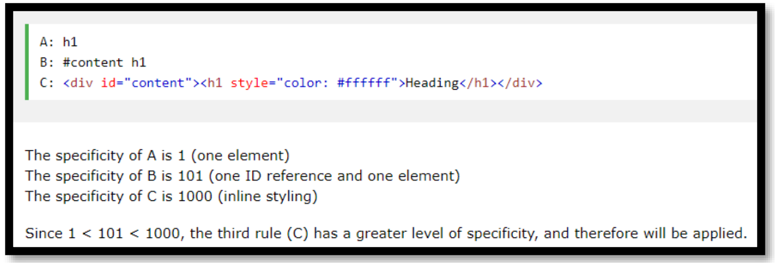
CASCADE AND INHERITANCE – SKIP

- Some CSS property values set on parent elements are inherited by their child elements, and some aren't.
  - When applying same HTML element can be the target of multiple CSS selectors.
  - These selectors may contain conflicting values. Which one applies?

- Main factors:

- Importance: use !important to over override ALL styling rules for a given property on an element
  - Also see this example

- Specificity:
  - The highest specificity wins (i.e. most specific gets applied last)
  - The universal selector (*) has low specificity, the ID selectors are highly specific!
  - To calculate specificity:
    - add 100 for each ID
    - add 10 for each attribute, class or pseudo-class
    - add 1 for each element name or pseudo-element



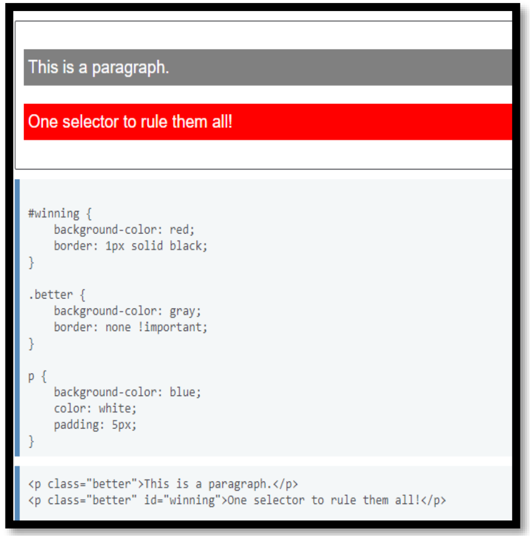
```
A: h1
B: #content h1
C: <div id="content"><h1 style="color: #ffffff">Heading</h1></div>
```

The specificity of A is 1 (one element)
The specificity of B is 101 (one ID reference and one element)
The specificity of C is 1000 (inline styling)

Since 1 < 101 < 1000, the third rule (C) has a greater level of specificity, and therefore will be applied.

- Order:
  - For multiple rules with same specificity, they are applied in the order they appear
  - So, the last one applied wins.

- Sources:
  - https://developer.mozilla.org/en-US/docs/Learn/CSS/Building_blocks/Cascade_and_inheritance
  - https://www.w3schools.com/css/css_important.asp
  - https://www.w3schools.com/css/css_specificity.asp

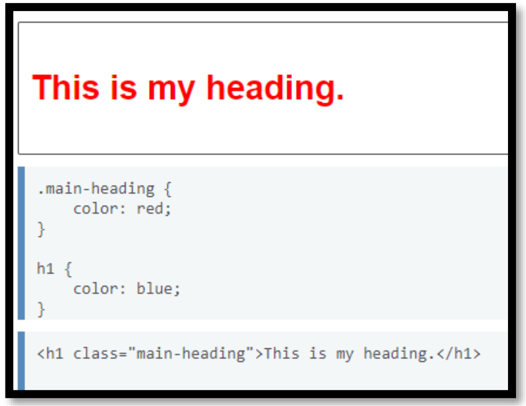


This is a paragraph.

One selector to rule them all!

```
#winning {
    background-color: red;
    border: 1px solid black;
}

.better {
    background-color: gray;
    border: none !important;
}

p {
    background-color: blue;
    color: white;
    padding: 5px;
}
```

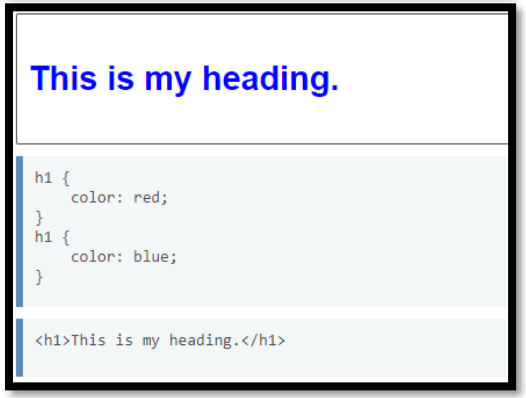
```
<p class="better">This is a paragraph.</p>
<p class="better" id="winning">One selector to rule them all!</p>
```



This is my heading.

```
.main-heading {
    color: red;
}

h1 {
    color: blue;
}
```

```
<h1 class="main-heading">This is my heading.</h1>
```



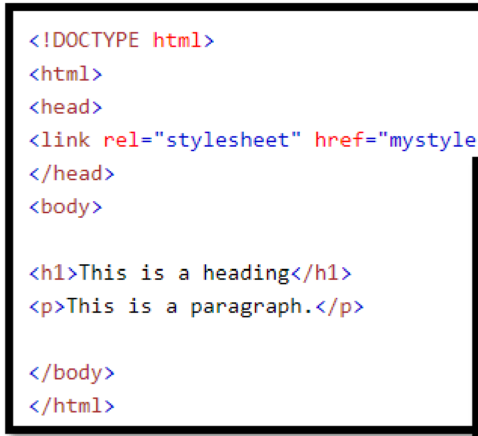
This is my heading.

```
h1 {
    color: red;
}
h1 {
    color: blue;
}
```

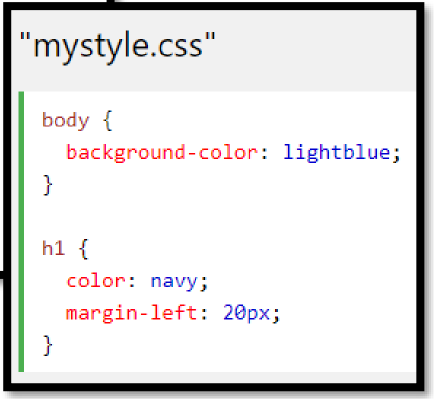
```
<h1>This is my heading.</h1>
```

HOW TO ADD CSS

- External CSS
  - Include a reference to an external style sheet file in the <link> element, inside the head section.
  - You can use one such file for the entire website!
  - Preferred for consistency.



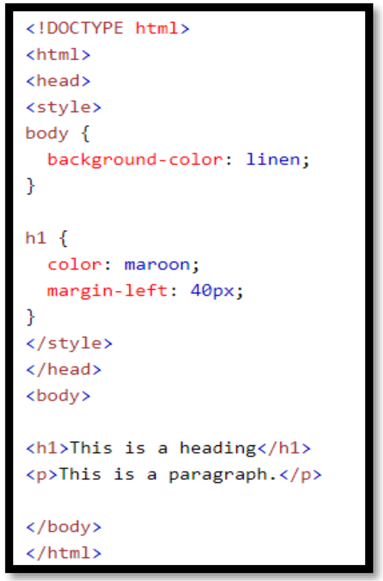
```
<!DOCTYPE html>
<html>
<head>
<link rel="stylesheet" href="mystyle.css">
</head>
<body>

<h1>This is a heading</h1>
<p>This is a paragraph.</p>

</body>
</html>
```



"mystyle.css"

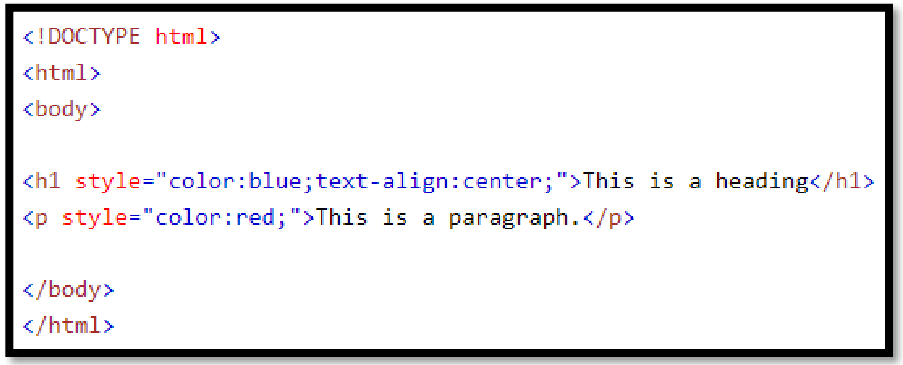
```
body {
  background-color: lightblue;
}

h1 {
  color: navy;
  margin-left: 20px;
}
```

- Internal CSS
  - Define your CSS style inside a <style> element, inside the head section.
  - Used when each page has a unique style



```
<!DOCTYPE html>
<html>
<head>
<style>
body {
  background-color: linen;
}

h1 {
  color: maroon;
  margin-left: 40px;
}
</style>
</head>
<body>

<h1>This is a heading</h1>
<p>This is a paragraph.</p>

</body>
</html>
```

- Inline CSS
  - add a style attribute directly to the relevant element



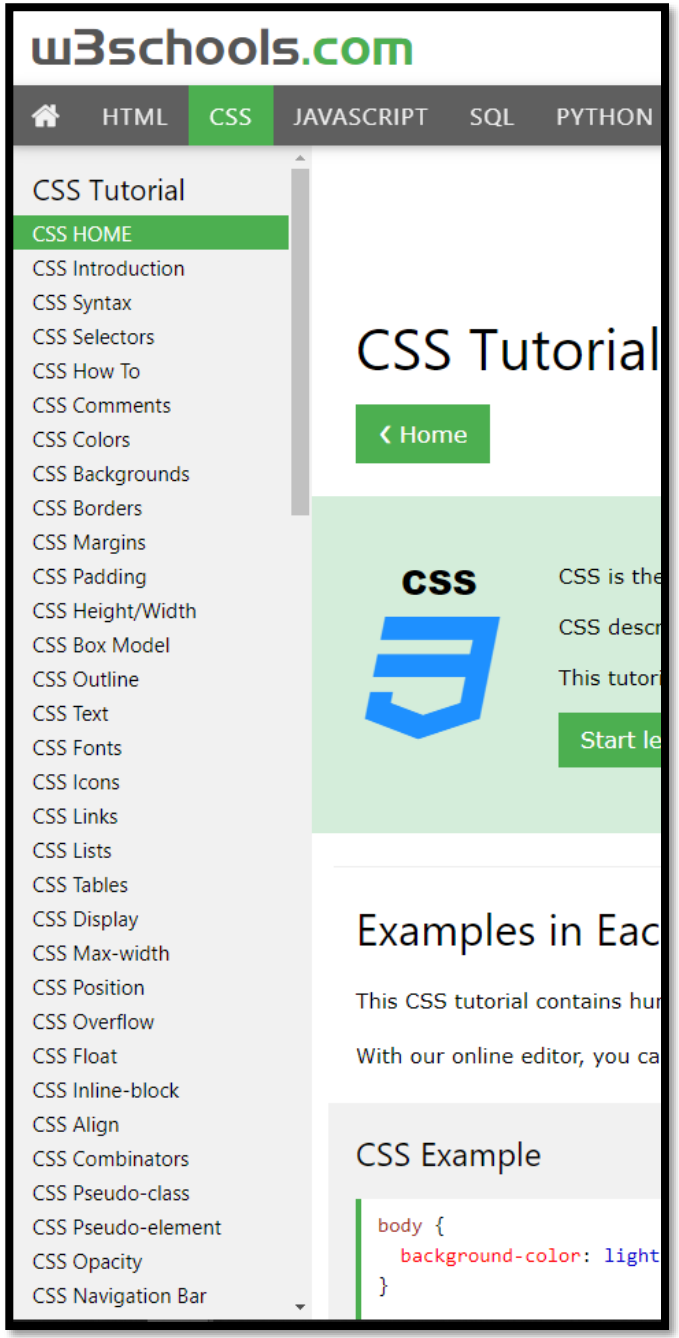
```
<!DOCTYPE html>
<html>
<body>

<h1 style="color:blue;text-align:center;">This is a heading</h1>
<p style="color:red;">This is a paragraph.</p>

</body>
</html>
```

- Source: https://www.w3schools.com/css/css_howto.asp

CSS

- In this course, you will be expected to
  - either know these topics or
  - be able to research them as needed.

- Use this site as a great guide:
  - https://www.w3schools.com/css/default.asp



w3schools.com

CSS Tutorial

CSS HOME
CSS Introduction
CSS Syntax
CSS Selectors
CSS How To
CSS Comments
CSS Colors
CSS Backgrounds
CSS Borders
CSS Margins
CSS Padding
CSS Height/Width
CSS Box Model
CSS Outline
CSS Text
CSS Fonts
CSS Icons
CSS Links
CSS Lists
CSS Tables
CSS Display
CSS Max-width
CSS Position
CSS Overflow
CSS Float
CSS Inline-block
CSS Align
CSS Combinators
CSS Pseudo-class
CSS Pseudo-element
CSS Opacity
CSS Navigation Bar

CSS Tutorial

CSS is the

CSS descr

This tutori

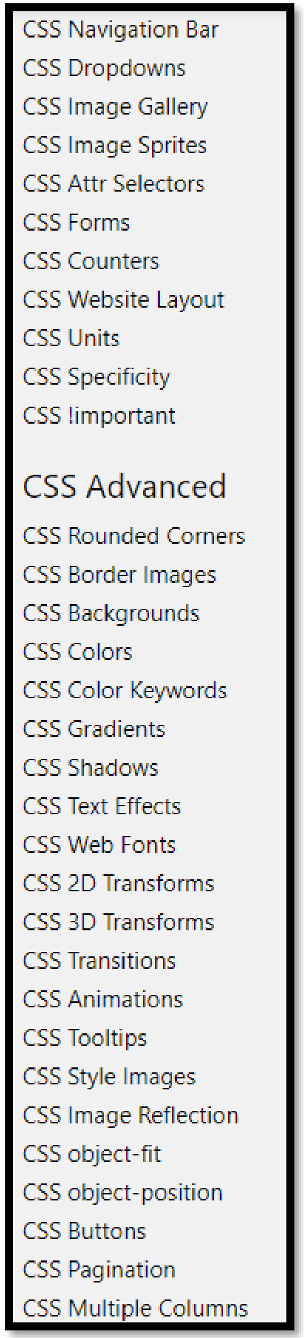
Examples in Eac

This CSS tutorial contains hur

With our online editor, you ca

CSS Example

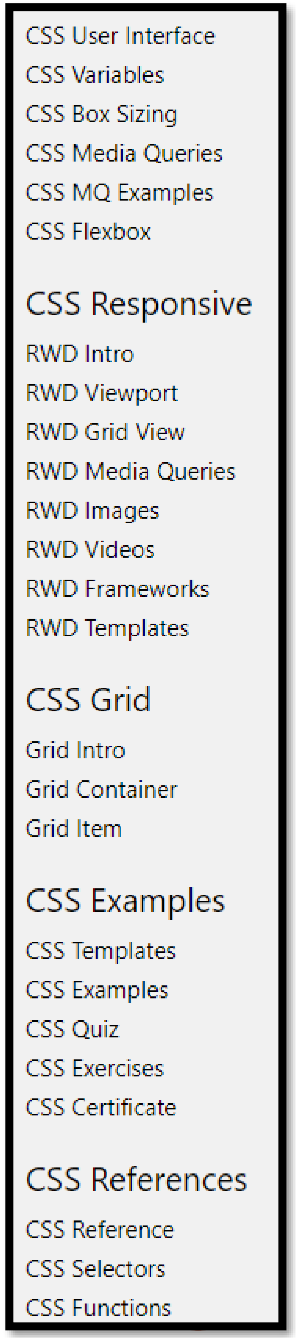
```
body {
  background-color: light
}
```



CSS Navigation Bar
CSS Dropdowns
CSS Image Gallery
CSS Image Sprites
CSS Attr Selectors
CSS Forms
CSS Counters
CSS Website Layout
CSS Units
CSS Specificity
CSS !important

CSS Advanced

CSS Rounded Corners
CSS Border Images
CSS Backgrounds
CSS Colors
CSS Color Keywords
CSS Gradients
CSS Shadows
CSS Text Effects
CSS Web Fonts
CSS 2D Transforms
CSS 3D Transforms
CSS Transitions
CSS Animations
CSS Tooltips
CSS Style Images
CSS Image Reflection
CSS object-fit
CSS object-position
CSS Buttons
CSS Pagination
CSS Multiple Columns



CSS User Interface
CSS Variables
CSS Box Sizing
CSS Media Queries
CSS MQ Examples
CSS Flexbox

CSS Responsive

RWD Intro
RWD Viewport
RWD Grid View
RWD Media Queries
RWD Images
RWD Videos
RWD Frameworks
RWD Templates

CSS Grid

Grid Intro
Grid Container
Grid Item

CSS Examples

CSS Templates
CSS Examples
CSS Quiz
CSS Exercises
CSS Certificate

CSS References

CSS Reference
CSS Selectors
CSS Functions

LAB/HOMEWORK: EXPLORING THE CONTOSO CONFERENCE APPLICATION

- **Module 01**
  - Exercise 1: Exploring the Contoso Conference Application
  - Exercise 2: Examining and Modifying the Contoso Conference Application

- You will find the **high-level** steps on the following page:
  https://github.com/MicrosoftLearning/20480-Programming-in-HTML5-with-JavaScript-and-CSS3/blob/master/Instructions/20480C_MOD01_LAB_MANUAL.md

- You will find the **detailed** steps on the following page:
  https://github.com/MicrosoftLearning/20480-Programming-in-HTML5-with-JavaScript-and-CSS3/blob/master/Instructions/20480C_MOD01_LAK.md

- For your homework submit one zipped folder with your complete solution.