MPART 1

I Instructions

In this first part, you will design and write a small website with a **topic of your own choice** consisting of webpages using Hypertext Markup Language (HTML5) code. You are to develop a basic website, which provides information in a clear way and links to other pages.

You are to demonstrate your understanding of the correct use of HTML tags. Use both the code and design view (if you are using MS Expression Web editor). It will be obvious to your marker if you only used the design view, as it shows up in the code structure. Automatically generated code looks different from manually created HTML code. We will look for evidence that you either manually coded the webpages or that you manually corrected the code generated by your software.

Section III of this first part has a **sample case study**. For this first stage, you will need to follow the same steps of the case study **but starting to use your own topic**. YOU MUST NOT USE THIS EXAMPLE TO COMPLETE YOUR ASSIGNMENT. DOING SO WILL ATTRACT NO MARKS.

Your topic is free for you to choose (a hobby? Small business?). However, extra marks will be awarded (see marking criteria) if you decide to pick any of the following:

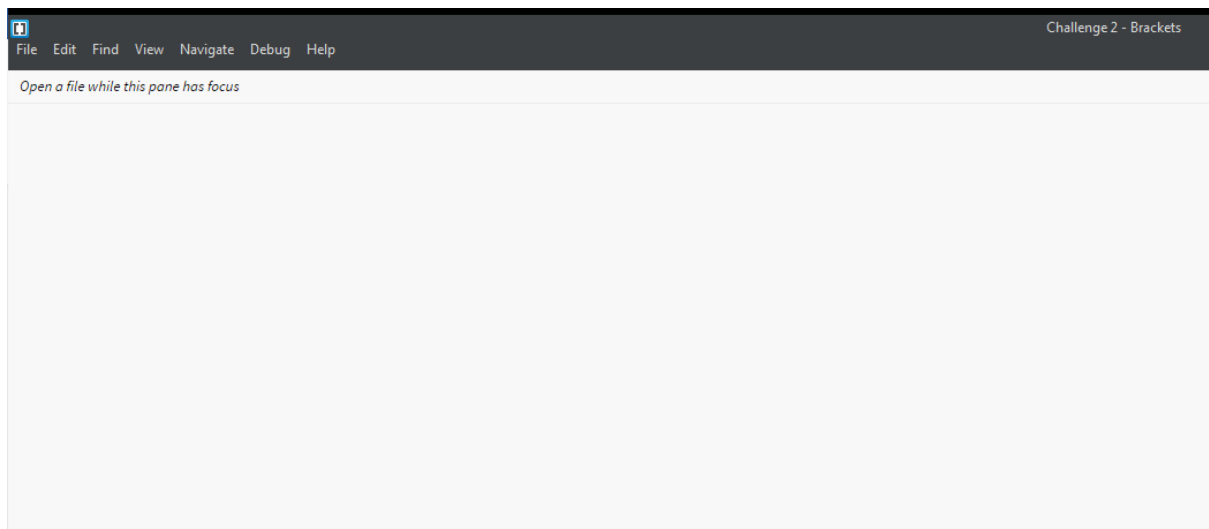- Australia's Landscapes.
- Australia's Wildlife.

These are very broad topics, so please talk with your lecturer/tutor to narrow them down.

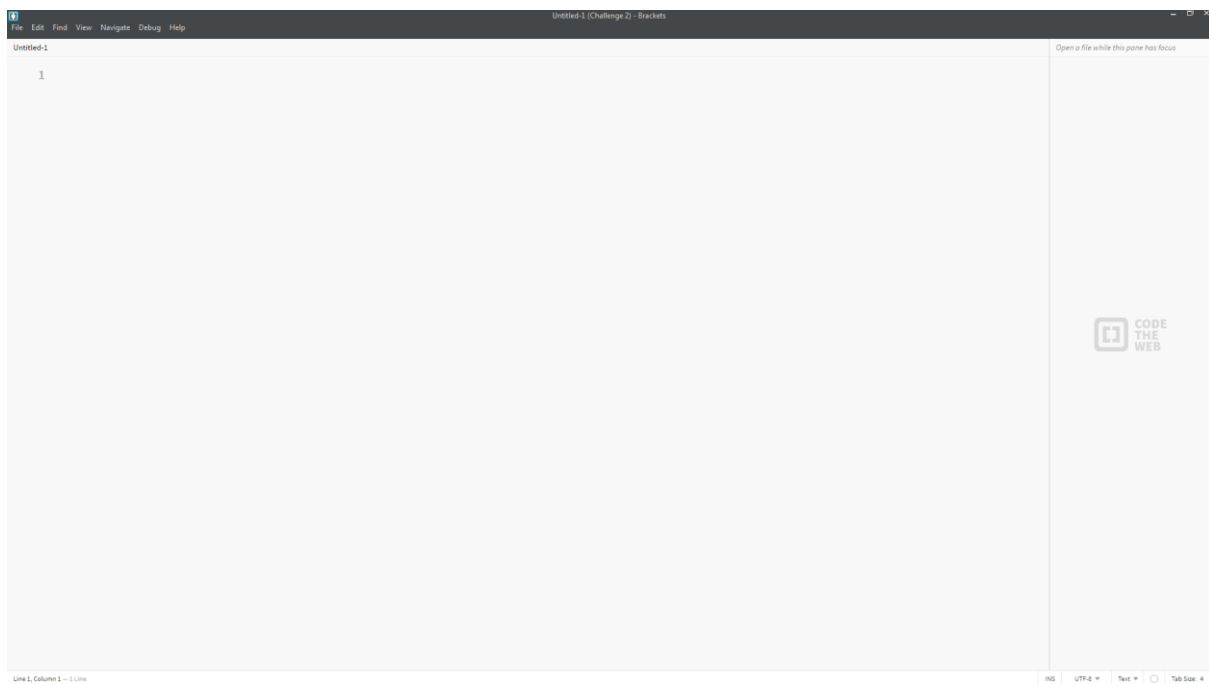### II Step by Step Guide for using Brackets ©

Note: This semester (Sem 2 – 2021) we are shifting towards using the "Brackets" application to design your webpage. I still believe that MEW is a better option to do this, but it is now outdated, and we need to move on. Also note that you are not required to use any of these applications. That is, you can use the software of your taste (Notepad++, Atom, BlueFish, Microsoft Visual Studio) so long as it is YOUR OWN CODE, and IT FOLLOWS THE MARKING GUIDE provided in your Canvas site.
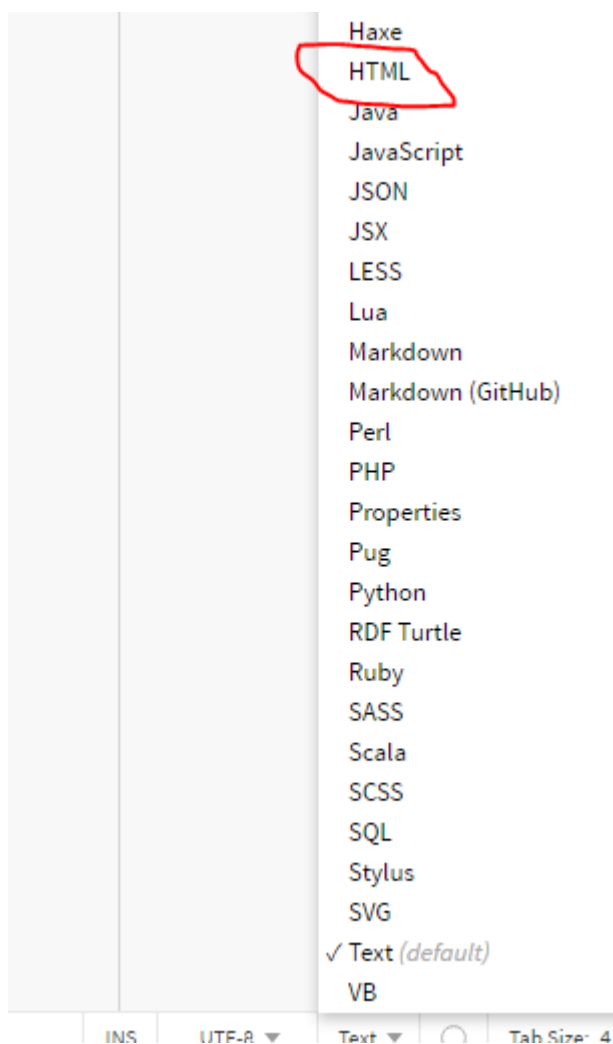
1. Start Brackets

2. Click on File → New

The following screen should prompt:

4. At the bottom right of your screen click in the black arrowhead next to the label "txt":

INS UTF-8 ▼ Text ▼ Tab Size: 4

5. Select "html" in the menu:

Haxe
HTML
Java
JavaScript
JSON
JSX
LESS
Lua
Markdown
Markdown (GitHub)
Perl
PHP
Properties
Pug
Python
RDF Turtle
Ruby
SASS
Scala
SCSS
SQL
Stylus
SVG
✓ Text (default)
VB

INS UTF-8 ▼ Text ▼ Tab Size: 4
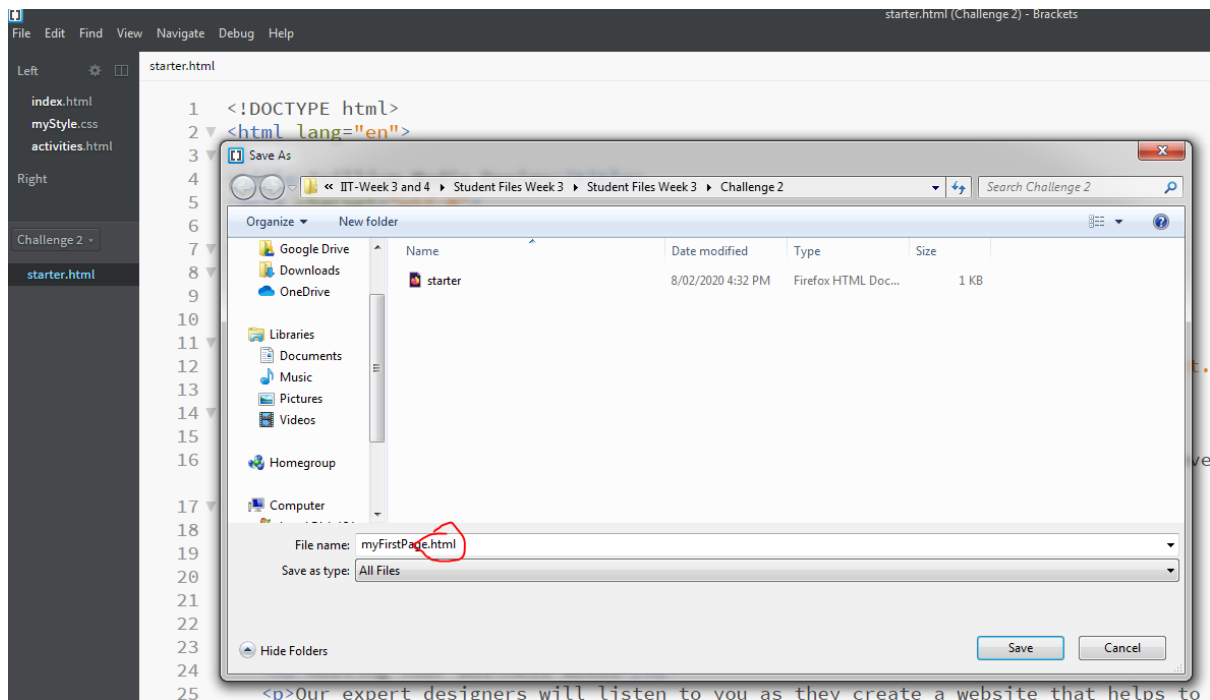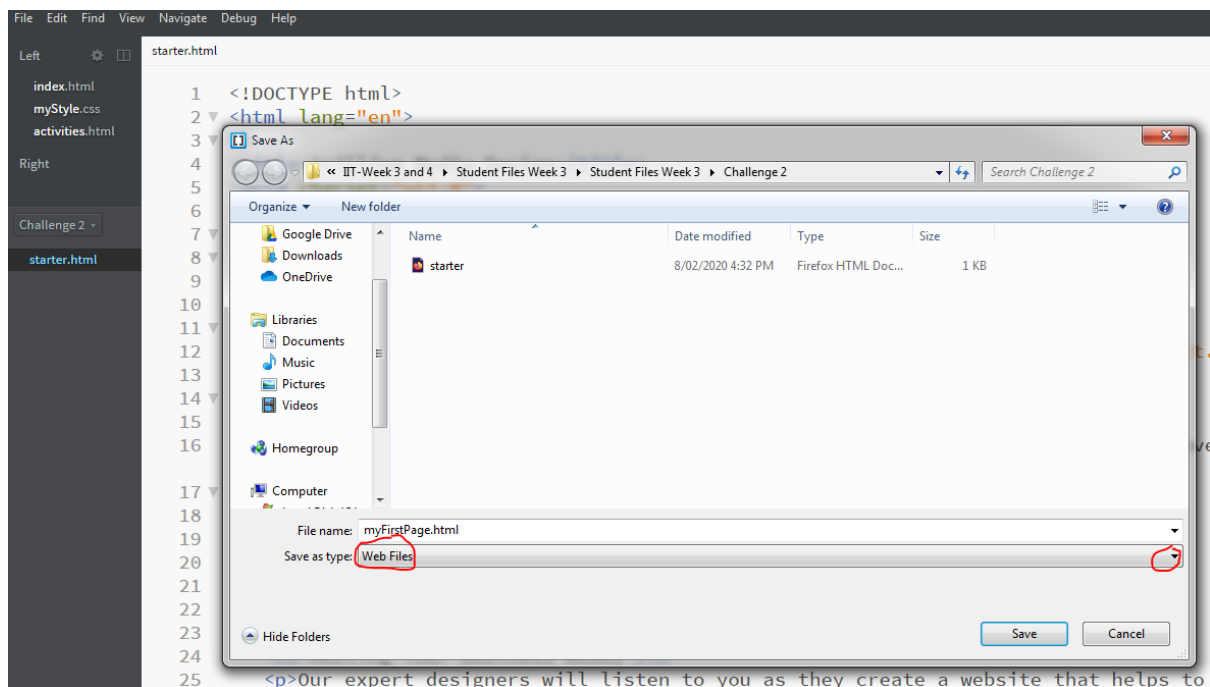
6. You are ready to code!!!

7. When saving your file (using the menu File → Save as...) be sure to add manually the extension of your file (.html).

8. You may need to specify the file type in the drop-down menu as well:

III A sample case study

This is your opportunity to compile everything you have learnt so far. Please, follow your lecturer's guidelines when creating this simple webpage, as you will be undertaking the same steps **to create your own.**
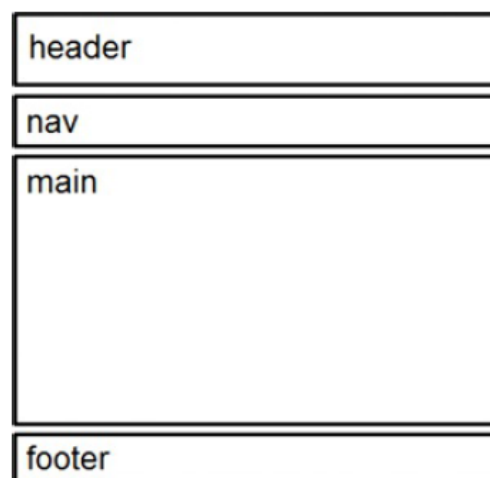
Fish Creek Animal Clinic

Magda Patel is a veterinarian and owner of the Fish Creek Animal Clinic. Her customers are local pet owners who range from children to senior citizens. Magda would like a website to provide information to her current and potential customers. She has requested a home page, a services page, a page for advice from a veterinarian, and a contact page.

A site map for the Fish Creek Animal Clinic website is shown in figure below. The site map describes the architecture of the website, which consists of a "Home" page with three main content pages: "Services," "Ask the Vet," and "Contact."
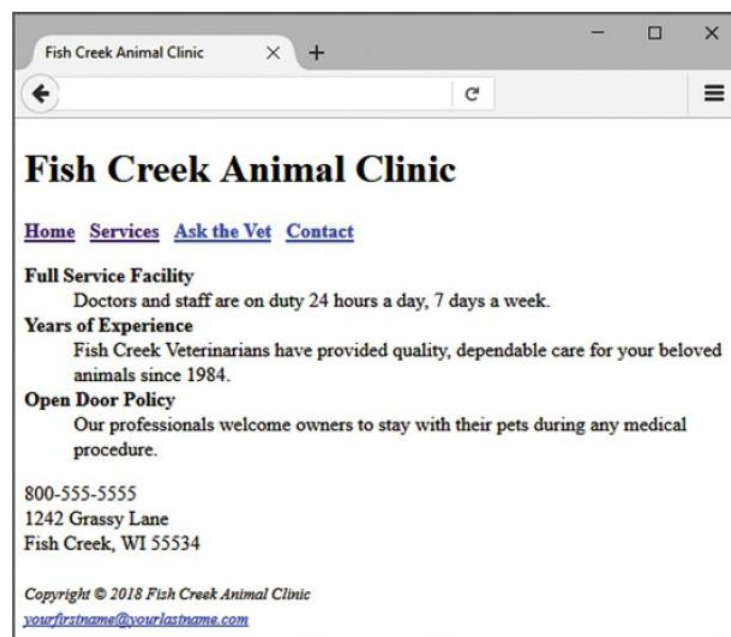

Figure below displays a wireframe sketch of the page layout for the website. It contains a header area, a navigation area, a main content area, and a footer area for copyright information.

Fish Creek wireframe

You have three tasks in this case study:

1. Create a folder for the Fish Creek website.
2. Create the Home page: index.html.
3. Create the Services page: services.html.

- Task 1: The *Website Folder*. Create a folder on your hard drive or portable storage device (USB stick) called "fishcreek" to contain your Fish Creek website files.

- Task 2: The *Home Page*. You will use a text editor application to create the Home page for the Fish Creek Animal Clinic website. The Home page is shown in Figure below.

Fish Creek index.html

Launch Brackets and create a web page with the following specifications:

1. **Web Page Title**. Use a descriptive page title. The company name is a good choice for the home page of a business website. On pages other than the home page, the title typically contains both the company name and a word or a phrase that describes the purpose of the page.

2. **Wireframe Header**. Code the header element with the text, "Fish Creek Animal Clinic" contained within a *heading 1* element.

3. **Wireframe Navigation**. Place the following text within a nav element with bold text (use the *<b>* element)

- Home Services Ask the Vet Contact

- Code anchor tags so that "Home" links to index.html, "Services" links to services.html, "Ask the Vet" links to askvet.html, and "Contact" links to contact.html. Add extra blank spaces between the hyperlinks with the   special character as needed.

4. **Wireframe Main Content**. Code the main page content within a main element.

a) Code the following content in a description list. Configure the text in each dt element to have strong importance and display in bold font weight:

- **Full Service Facility**
  Doctors and staff are on duty 24 hours a day, 7 days a week.
- **Years of Experience**
  Fish Creek Veterinarians have provided quality, dependable care for your beloved animals since 1984.
- **Open Door Policy**
  Our professionals welcome owners to stay with their pets during any medical procedure.

b) Configure the following address and phone number contact information within a div element below the description list. Use line break tags to help you format this area.

- 800-555-5555
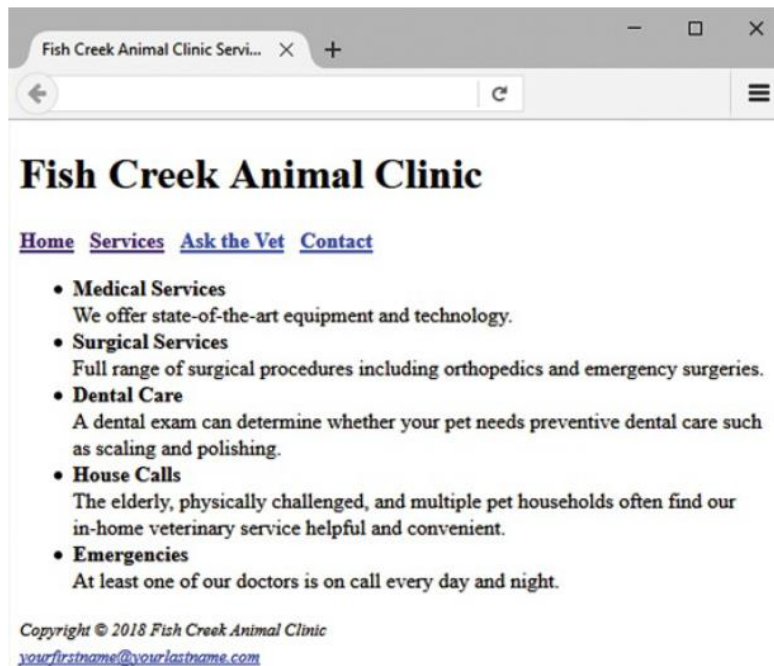- 1242 Grassy Lane
- Fish Creek, WI 55534

5. **Wireframe Footer**. Code the following copyright and e-mail link information within a footer element. Format it with small text size (use the <small> tag) and italics font style (use the <i> tag).

- Copyright © 2018 Fish Creek Animal Clinic
- Place your name in an e-mail link on the line under the copyright.

Your page may seem a little sparse, but don't worry; as you gain experience and learn to use more advanced techniques, your pages will look more professional. White space (blank space) on the page can be added with <br> tags where needed. Your page does not need to look exactly the same as the sample. Your goal at this point should be to practice and get comfortable using HTML.

Save your page in the fishcreek folder, and name it index.html.

- Task 3: **The Services Page**. Create the Services page shown in figure below. A technique that improves productivity is to create new pages based on existing pages so that you can benefit from your previous work. Your new Services page will use the index.html page as a starting point.

Open the index.html page for the Fish Creek website in MWE4. Select File > Save As, and save the file with the new name of services.html in the fishcreek folder. Now you are ready to edit the page.

1. **Web Page Title**. Modify the page title. Change the text contained between the <title> and </title> tags to the following:

- Fish Creek Animal Clinic Services

2. **Wireframe Main Content**.

a) Delete the Home page content description list and contact information.
b) Use an unordered list to add the services content to the page.
c) Configure the name of each services category to use bold font weight and have strong emphasis. (Use the <strong> tag.) Use line break tags to help you configure this area. The service categories and descriptions are as follows:

- **Medical Services**
  We offer state-of-the-art equipment and technology.
- **Surgical Services**
  Full range of surgical procedures including orthopaedics and emergency surgeries.
- **Dental Care**
  A dental exam can determine whether your pet needs preventive dental care such as scaling and polishing.
- **House Calls**
  The elderly, physically challenged, and multiple pet households often find our in-home veterinary service helpful and convenient.

- **Emergencies**

  At least one of our doctors is on call every day and night.

Save your page and test it in a browser. Test the hyperlink from the services.html page to index.html. Test the hyperlink from the index.html page to services.html. If your links do not work, review your work, paying close attention to these details:

- Verify that you have saved the pages with the correct names in the correct folder.
- Verify your spelling of the page names in the anchor elements.

Test again after you make changes.

PART 2

I Instructions

This stage will help you to "dress up" the webpage you created in stage 1 using CSS. Once again, here you'll find a case study, as an example only, to get an idea about what it is expected to complete by the due date. The case study DOES NOT HAVE every element included in the marking criteria. The aim is you to improve the webpage developed in the first stage using your own topic, **but including ALL CSS approaches: embedded, inline, and external**.

You are to demonstrate your understanding of the correct use of CSS styling, configure background and text colour on web pages, create style sheets that configure common colour and text properties, apply inline styles, use embedded style sheets, use external style sheets, configure element and class selectors, utilize the "cascade" in CSS, and validate CSS. In addition, **your website MUST INCLUDE at least THREE PAGES (including the main page). All of them linked by a <nav> menu**.
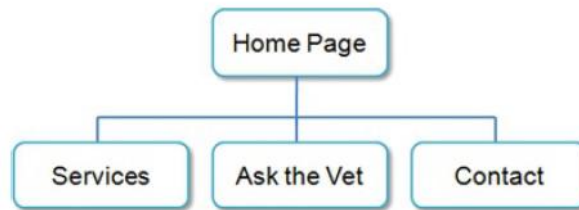
The map of your website as well as references are both important. You must include the map of your website according to the lectures. Similarly, you need to submit a word document with all the references you have come across so far. Validation is still important. See "The W3C CSS Validation Service" to get this task done. The submission of a screenshot of the validation is enough for marking purposes.

Part II of this document has a sample case study that we'll go through during the tutorial. Following the approach of the first stage, you will need to follow the same steps of the case study but using your own topic.
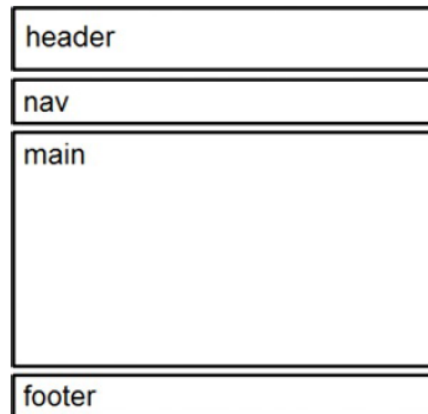
II A sample case study using CSS

This is your opportunity to compile everything you have learnt so far. Please, follow your lecturer's guidelines when creating this simple webpage, as you will be undertaking the same steps **to create your own.**

Following the Fish Creek Animal Clinic Case Study of the previous stage. Figure below shows a site map for the Fish Creek website.
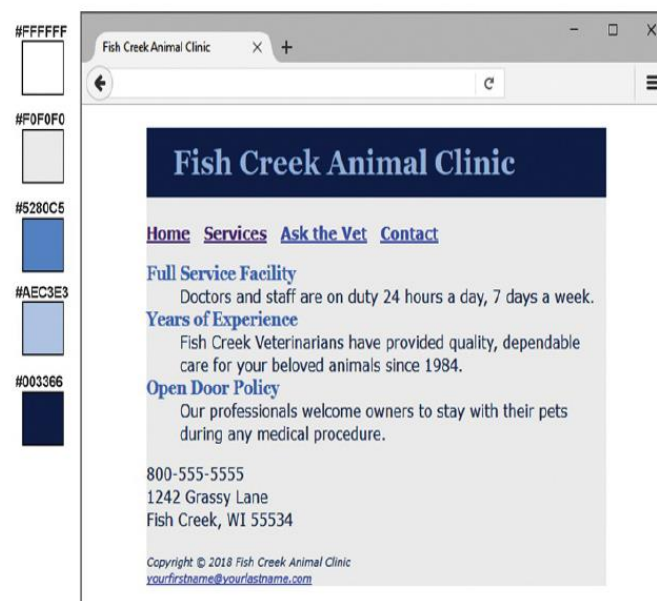
The Home page and Services page were created previously. You will develop a new version that uses an external style sheet to configure text and colour. Figure below depicts the wireframe page layout.

Wireframes are diagrams (layouts) of a page that indicate how the structure of a web page could be configured with the header, nav, main, div, and footer elements.

You have the following tasks:

1. Create a new folder for this Fish Creek case study.
2. Create an external style sheet named fishcreek.css that configures the colour and text for the Fish Creek website.
3. Modify the Home page to utilize an external style sheet to configure colours and fonts. The new Home page and colour swatches are shown in the following figure:

4. Modify the Services page to be consistent with the new Home page.
5. Configure centred page layout.

Hands-On Practice Case Study

Task 1: The Website Folder.

- Create a folder on your hard drive or portable storage device called fishcreekcss.
- Copy all the files from your previous week's fishcreek folder into the fishcreekcss folder.

Task 2: The External Style Sheet.

You will use a text editor to create an external style sheet named fishcreek.css. Code the CSS to configure the following:

- Global styles for the document (use the body element selector) with background colour #FFFFFF; text colour #003366; and Verdana, Arial, or any sans-serif font.
- Styles for the header element selector that configure background colour #003366, text colour #AEC3E3, and a serif font such as Georgia or Times New Roman.
- Styles for the h1 element selector that configure 200% line height and set the text-indent property to 1em.
- Styles for the nav element selector that display text in bold.
- Styles for the dt element selector that configure #5280C5 text colour, size 1.1em font in bold, and a serif font such as Georgia or Times New Roman.
- Styles for a class named category with bold font, text colour #5380C5, and a serif font such as Georgia or Times New Roman.
- Styles for the footer element selector with a small font size (.70em) and italic text.
- Save the file as fishcreek.css in the fishcreekcss folder. Check your syntax with the CSS validator (http://jigsaw.w3.org/css-validator).
- Correct and retest if necessary.

Task 3: The Home Page.

Launch a text editor and open the index.html file. You will modify this file to apply styles from the fishcreek.css external style sheet as follows:

- Add a <link> element to associate the web page with the fishcreek.css external style sheet file.
- Configure the navigation area. Remove the <b> element from the navigation area, because the CSS will configure the bold font style.
- Configure the content area. Remove the <strong> tags from within each dt element, because the CSS will configure the bold font style.
- Configure the page footer area. Remove the <small> and <i> elements—they are no longer needed since CSS is now used to configure the text.

Save the index.html file, and test in a browser.

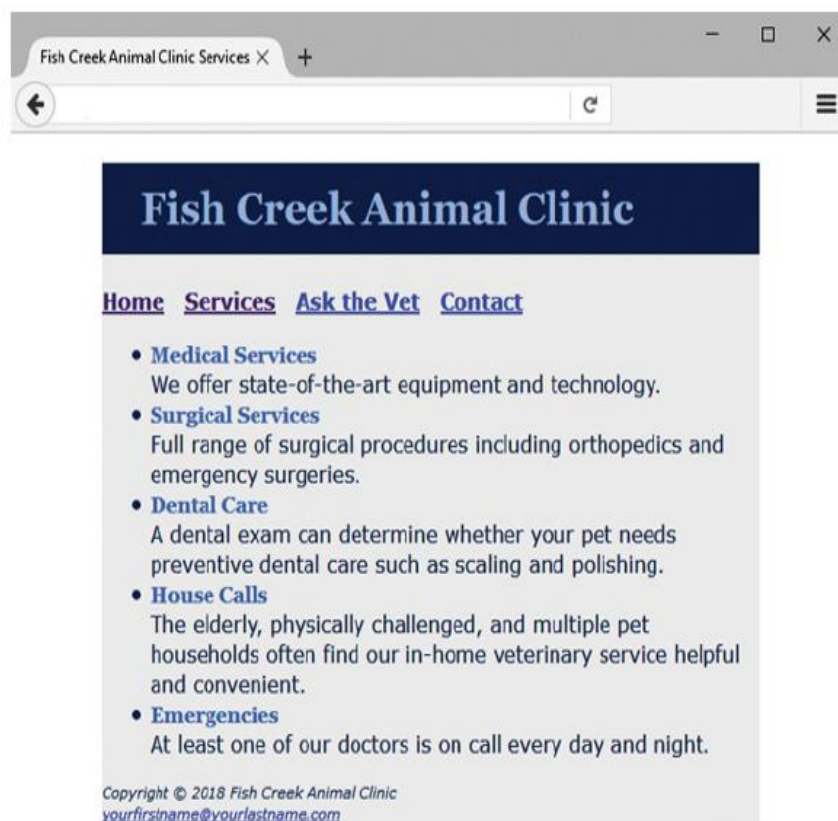**Task 4**: The Services Page.

Launch a text editor and open the services.html file. You will modify this file in a similar manner:

- Add the <link> element, configure the navigation area and page footer areas, configure the category classes (Hint: Use the <span> element to contain the name of each service offered), and remove the strong tags.)
- Save and test your new services.html page.

Task 5: Center Page Layout with CSS.

Modify fishcreek.css, index.html, and services.html to configure page content that is centred with 80% width with a light grey background colour.

- Launch a text editor and open the fishcreek.css file.
- Add a style rule for an id named wrapper with width set to 80%, margin-right set to auto, margin-left set to auto, and background colour set to #F0F0F0.
- Launch a text editor and open the index.html file. Add the HTML code to configure a div element assigned to the id wrapper that "wraps", or contains, the code within the body section. Save and test your index.html page in a browser and you will notice that the page content is now centred within the browser viewport.
- Launch a text editor and open the services.html file. Add the HTML code to configure a div element assigned to the id wrapper that "wraps", or contains, the code within the body section. Save and test your services.html page in a browser and you will notice that the page content is now centred within the browser viewport as shown in Figure below.

Experiment with modifying the fishcreek.css file. Change the page background colour, the font family, and so on. Test your pages in a browser. Isn't amazing how a change in a single file can affect multiple files when external style sheets are used?

PART 3

I Instructions

This final stage, you will complete the design of your small website, with a topic of your own choice, consisting of webpages using Hypertext Markup Language (HTML) code and Cascading Style Sheets (CSS). Your site should provide information in a clear way and links to other pages and websites. You are to include appropriate images, lists and tables on your pages and CSS functionality.

In this stage, you are to show your understanding of good web design principles, such as clear hierarchy, indentation, commenting, easy navigation, and clear presentation of information.

II Requirements

Starting from your previous webpage (at this stage, you should have already developed a main page named index.html with all the basic elements and CSS styling), you need to complete a minimum of three different subpages. These can be on the same hierarchy level or on different levels. Remember that one of the points you will be marked on is good web design and a clear hierarchy! Each page needs to have a **minimum of 50 meaningful words on them**.

Make sure that the user can get to all your pages via hyperlinks as well as return to previous pages via hyperlinks (rather than having to use the back button of the web browser). Also include at least two links (anchor elements) to external websites, e.g. http://www.canberra.edu.au.

All the elements to be assessed for this final task are listed in the marking guide of Section IV.

III Submission

Zip (compress) the entire project folder and submit it via Canvas. Please note that you must first upload your ZIP file and then click on the Submit button. This is now part of the electronic submission via Canvas. **Your submission must include at least three html pages, CSS files, pictures, screenshots with the validation results (CSS and HTML), a word document with the site map, wireframe, and the list of references**.

IV Marking Guide

Elements of HTML design	Marks
Headings	2
Paragraph	2
Line Break	2
Blockquote	2
Ordered List	2
Unordered List	2
Description List	2
Special Characters (at least two)	2
Structural Elements (div, header, footer, main, nav)	4
Anchor Element	2
email Hyperlinks	1
Total 1	23

Styling with CSS	Marks
At least one inline CSS call is used.	3
At least one CSS embedded style is used.	3
At least one external CSS file is implemented	3
Background-colour property is used	2
Font family and colour properties are used.	2
Font size and weight properties are used.	2
Line height property is used.	2
Margin property is used.	2
Text-Align and decoration properties used.	2
At least two classes or IDs are implemented	4
At least one Span element is included.	1
At least one centering instruction is included	1
Total 2	27

Website Completion	Marks
This criterion is linked to a learning outcome Layout and structure	
Is the information on the pages easy to find? Does the layout support the viewing of the pages? Does the student use a suitable colour scheme to produce a highly readable site?	5
This criterion is linked to a learning outcome Navigation	
Is it easy to get from one page to another? Are links (anchor elements) going forwards and backwards? Are all links (both to other pages on the site and external links) working?	5

This criterion is linked to a learning outcome Content	
Has the student created the minimum number of webpages (three pages)? Has the student got a sufficient amount of content (minimum 50 topic-related words) on each page? Has the student used their own words in the content or copied it from somewhere? If copied, less than 10%? Referenced properly in code and on the HTML page?	5
This criterion is linked to a learning outcome Code	
Is the code clearly structured (head, body)? Has the student used empty lines to separate blocks of code? Has the student used different levels of indentation in the code? Did the student use comments in the code?	5
This criterion is linked to a learning outcome Image / graphics files	
Minimum of two images / graphics files on separate pages. What size is the image displayed at? Does the actual image resolution lend itself to be put on a webpage?	5
This criterion is linked to a learning outcome Table	
Minimum of one table with at least three rows and three columns. Has the student chosen a suitable table layout?	5
This criterion is linked to a learning outcome Metadata	
Has the student added metadata to each page? Has the student followed the Dublin Core recommendations?	5
Total 3	35
Site Admin	Marks
HTML and CSS Validation (all pages have been validated and the errors corrected).	5
Site Map and wireframe (s) were included.	5
References were included in a Word document (either APA or Harvard styles used)	5
Extra Marks* (2)	0
Total 4	15
Final Mark	
Grand Total	100

* Included when total is up to 98