# Tutorial

## Activity 1: Writing copy for your first web page

In the lab this week, you will be forking and cloning a starting repo which contains very basic HTML5 web page templates, one which you will modify with some information about yourself.

*These are answers I copied from what I said on My First Webpage (except the first one).*

Some things to think about:

* *How do you use the internet?*

I don’t really get this question, but…I use the internet as a research tool (through mostly Google), entertainment (music, movies, games) and communication medium (chat online with friends, contact far-away families…).

* *What websites do you visit? (NB: "family friendly" !)*

I would say Facebook and Youtube. I think many people these days have FOMO and parhaps I do too, hence why I'm always checking Facebook. For Youtube, it's only a place for me to watch videos and listen to music. You may be wondering why I don't use the free (or paid) version of Spotify, I'm just not used to its interface and it doesn't seem as diverse as Youtube.

Recently I have started to use Reddit more too. It's a forum with a variety of topics up for discussion, ranging from scientific research to showbiz gossips. Sometimes I even search for simple explanations on certain topics in school by looking in the ELI5 section.

* *Have you any programming or web design experience?*

Yes I have. Well, I'd like to think so... I have taken a few introductory courses to other programming languages such as Python and Java. I have also...Googled, uhh I mean researched a bit on HTML and web programming in general before taking this class.

* *Do you see yourself as a programmer or designer, or both?*

I would like to be both if possible. Being able to work on both the back-end and front-end seems amazing. I think coding is a pretty cool job, but making things look beautiful feels good too.

* *What has brought you to this course at RMIT?*

......It is...a mandatory course. But in all honesty, I would've taken it as an elective if possible anyway. I think Web Programming is a practical and interesting subject. The programming courses I took before were a bit...dry, they were mostly "Here's a problem, now write some codes logically to solve it", they lacked the freedom in creating and styling content that Web Programming has.I would like to be both if possible. Being able to work on both the back-end and front-end seems amazing. I think coding is a pretty cool job, but making things look beautiful feels good too.

* *What are you hoping to get out of the course?*

Being able to create a fully functional and beautiful website.

* *What are your interests or hobbies?*

Of course. I have general hobbies such as listening to music, watching movies, reading books and playing games.

I especially have an interest in the crime-solving and thriller genre. Did you know there's a Youtube channel called WIRED which has videos on interviews with real former FBI and CIA agents?

This doesn't mean I dislike happy and light stuff. I do watch romance such as La La Land, Me Before You and even the rom-com series How I Met Your Mother (highly recommended!). Basically the things I watch, read and play are not limited to only things I actually have an interest in.

Work with one or two friends to flesh out your copy.

This activity is to get to know your classmates and yourself a little better. By the end of this exercise you should have some idea of what to put in your personal webpage.

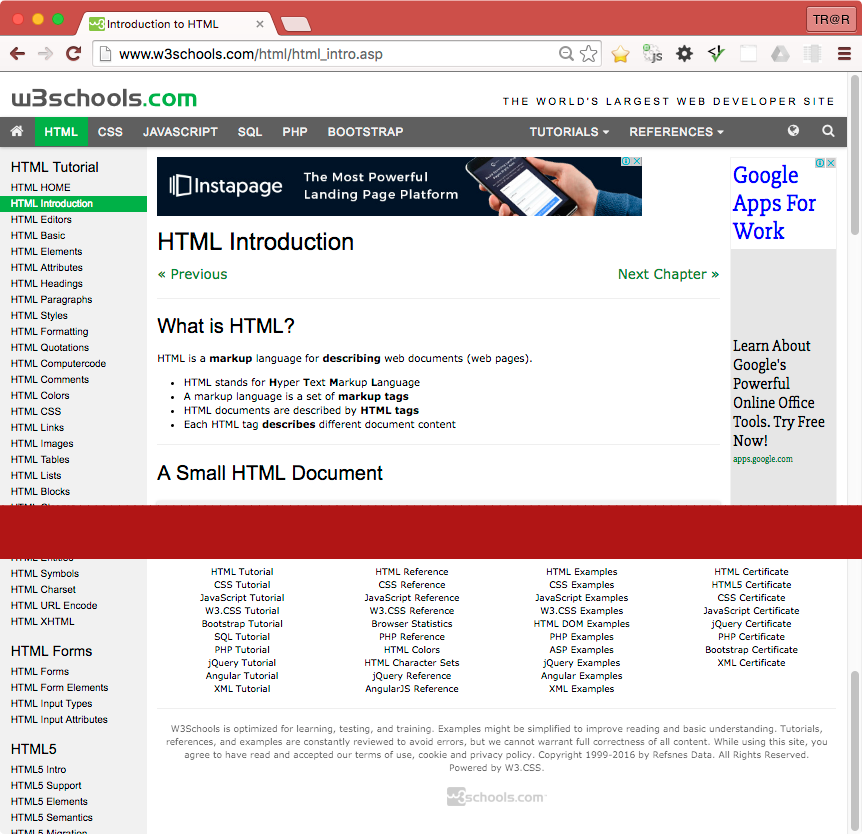
For administrative reasons, there will be a few other fields to fill in (name, student number, a personal photo). This will help the entire web programming team get to know you better and more importantly know how to pitch answers to your questions.

## 

## 

## Activity 2: Examine the Elements of a Webpage

Have a look at the following webpage: <http://www.w3schools.com/html/html_intro.asp>. The top and bottom of the page is shown with a red bar separating the two sections. *If you have a laptop please have a look at the live version*.



* **On the webpage above, identify the following web elements:**
  + **Headings** (eg <h1>...</h1>, <h2>...</h2>, … , <h6>...</h6>)

**A:**

On the grey top navigation bar, each section’s title within the MORE, REFERENCES and EXERCISES drop-down menu is an h3. For example, in the EXERCISES menu: Exercises, Quizzes and Certificates are all h3.

On the side navigation bar, there are 7 h2 elements, easily recognized because they are bigger than other items in the side bar. They are (HTML) Tutorial, Forms, Graphics, Media, APIs, Examples and References.

In the main section of the HTML Introduction, the “HTML Introduction” itself is a heading 1.

“What is HTML?”, “A Simple HTML Document”, “HTML Tags”, “Web Browsers”, “HTML Page Structure”, “The <!DOCTYPE> Declaration”, “HTML History” are heading 2. If you click on “REPORT ERROR” near the bottom of the page, it actually opens up another section and the “Report Error” there is a heading 2.

The “Example”, “Example Explained” under that are heading 3.

On the right, COLOR PICKER, HOW TO, SHARE and CERTIFICATE, plus all 4 column’s titles at the bottom of the page are heading 4.

* + **Paragraphs** (<p></p>),

The line “HTML is the standard markup language for creating Web pages.” is wrapped in <p> tag. So is the “HTML tags are element names surrounded by angle brackets: <tagname>content goes here...</tagname>”. The “**Tip**” line is also wrapped in <p>. Several other normal sentences in other sections also use <p> tag.

* + **Anchors or Hyperlinks** (<a href="...">...</a>),

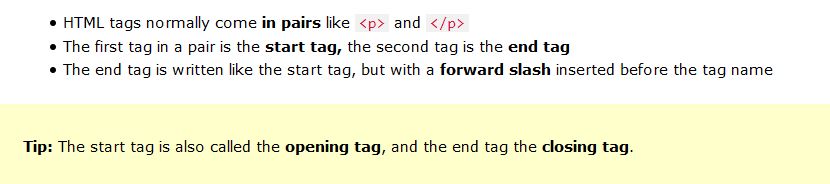
Pretty much any links/sections and buttons/icons (such as toggle dark code) listed in the navigation bars on the top, left and right and at the end of the page use the anchor tag. The “Previous” and “Next”, even the “Try it yourself” buttons in the main section are anchors too.

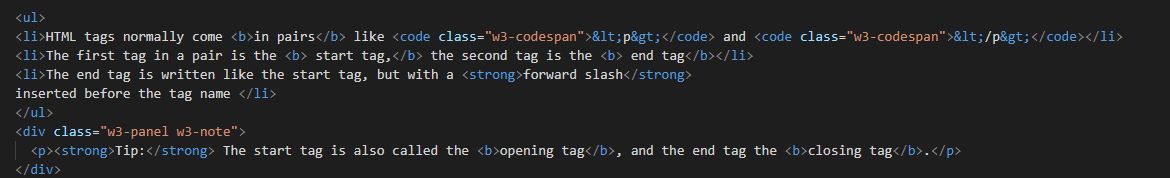
* + **Images** (<img src="..." alt="..."/>) and other multimedia.

Example of images are the image in the “Web Browsers” section, the .gif image on the right under COLOR PICKER.

At the very bottom of the page, the grey w3schools.com is actually a clickable .gif image.

* Identify styling such as **bold** and **italic** text. We will cover shading and colouring next week.





I’ve inserted images capturing examples where the words are bold. Specifically, the “Tip” and “forward slash” use the <strong> tag instead of <b>, but the visual effect is the same.

* What is the difference between an **unordered list** (<ul>) and **list items** (<li>). Can you see examples of each on the webpage above?

Unordered list is used for grouping and listing items that are similar but have no specific order. Items in <ul> are listed (by default) using the small black dot.

List items tag <li> marks items within a list. It can be used to list items either under an unordered list or an ordered list.

Examples of <ul> elements with <li> elements inside are the 7-item lists in the “What is HTML?” section and the “Example Explained” section.

Another example is the 3-item list in the HTML Tags section, also shown in the images above.

* Does this website use modern HTML5 Contextual Elements? (eg <header>, <nav>, <main>, <footer>, <aside>). Identify which areas should be grouped into these elements.

No this website doesn’t use those elements.

From the top grey navigation bar up, it can may be put in the <header> because it gives introductory information to everything in the website. The top grey nagivation bar and also the left side navigation links should be put in <nav> tags. The links under HOW TO on the right and 4 columns at the bottom of the page *can* be put in <nav> too…*maybe*. It is a subjective judgement call. They don’t really add much to page navigation compared to the top and left bars, they seem more like extra links that could be useful so they don’t really need to be in <nav> tags.

Everything in the section from HTML Introduction to the line “This tutorial follows the latest HTML 5 standard.” Should be put in <main> because it is the main content of this specific page.

Everything from the REPORT ERROR button down should be put in <footer>. They contain extra information about the page itself, extra links and copyrighted statement.

The whole COLOR PICKER section can be put in the <aside> tag. They’re all useful links but not too important and doesn’t add much to the content.

* How many navigation elements are there? If more than one, why are there more than one?

There are about 3 navigation elements. One on top, one left and one right. There are more than one because each navigation element groups different types of links together and does different things. The one at the top is for navigating throughout the whole website. It is consistent and almost always there (well except when the user goes to the homepage, then it changes to a green one).

The left-side navigation is for this specific page (HTML) only, and it changes depending on which page the user is at.

The right-side navigation doesn’t really change depending on the page, but it can’t be grouped with the top one because it only provides extra information and is not too important to the overall website.

## 

## 

## 

## Activity 3: Evolution of the HTML standard

*What page elements were present in the very first versions of HTML? What form did the first web documents take?*

*Some page elements such as <HEADER>, <BODY>, <TITLE>, <NEXTID>, <H1> to <H6>, <A>, <UL>, <LI>, <DL>, <DT>, <P>….*

Well they are listed here: <https://www.w3.org/History/19921103-hypertext/hypertext/WWW/MarkUp/Tags.html>

The first web documents looked really plain and simple with no images or effects, they were like text documents.

*HTML3 and HTML4 standards were developed at a time that was seen as being very chaotic. Name some reasons for why this was and state what new elements we saw introduced at this time.*

Because the World Wide Web was growing strongly back then and new computer OS with GUI became common. There were also competition between two key players in the web browser industry: Netscape Navigator (Firefox today) and Microsoft Internet Explorer. They wanted to make their own rules and that sometimes messed up the website for the developers.

Later Flash also appeared and it allowed developers to build “flashy” and professional animated websites. Some even built a website purely in Flash. The technological demand means HTML must be developed more to catch up with the changing times.

To modernize HTML, content and style were being separated. It was encouraged to use CSS to style web pages instead of HTML.

During this whole period, new elements such as <img>, <table> were introduced. There were tags and attributes to control style like font, color, align, border… but they were all deprecated later to encourage CSS usage.

The XHTML standard is described as being very strict, quite unlike HTML3 and 4. What rules did XHTML impose on developers? Think in terms of:

* Tags: closing and nesting them in correct sequence.
* Use of uppercase and lowercase characters.
* Separation of style and content

How did these changes allow XHTML documents to be more accessible across a wider range of devices: eg PC's, early WAP enabled mobile phones, screen readers for the visually impaired etc.

Some rules XHTML imposed were:

* Tags must be closed, even empty/non-paired ones such as <br />
* Tags must be properly nested: the element that is opened last has to be closed first. For example: <strong><em>…</strong></em> should work fine in HTML (even though it looks weird), but it wouldn’t work in XHTML. In XHTML, it has to be <strong><em>…</em></strong>.
* All tags must be in lowercase.
* Because XHTML was a strict version of HTML4, the use of stylistic tags and attributes were deprecated as well. If an attribute is stated, it must always be quoted and the value must always be stated, it cannot be minimized even if it’s obvious.

These strict rules allow any tools or applications to parse and process the websites better because they are consistent. PC, mobile devices with different languages and technologies can still easily read and display the websites correctly. With specific rules in place, screen readers used by impaired people can have an easier time skimming through the document without worrying about any inconsistent placement of tags or unclear attributes…

HTML5 introduced many new elements and attributes. Think in terms of:

* Simpler doctype declaration.
* New tags, including standard multimedia tags (audio/visual).
* New form inputs, validation of form data using html markup, not code/scripting
* Semantic Web elements

Discuss the benefits and improvements contained in the HTML5 standard.

Because HTML5 introduced new semantic web elements, this allows developers to structure the websites better. A better structured website is more accessible for screen readers because it can jump to certain sections easily, such as the navigation or any article in the page.

Not only will semantic elements help increase the ease of use for visually impaired users, it will help the developers maintain and fix their codes better too. More descriptive tags means the code is easier and cleaner to read for when developers need to change anything.

With the inclusion of multimedia tags, HTML5 by itself can support more types of content without the need for Flash or other third-party players. This plus the introduction of new kinds of inputs and feedback forms mean that users will feel more engaged to interact with the website.

Finally, HTML5 has brought back many (but not all) of HTML4's more relaxed coding rules. What are they and why are the strict XML syntax rules of XHTML no longer needed today?

Some relaxed rules that were brought back:

* Not all tags need to be closed, especially empty tags.
* Some tags and attributes for styling such as <b> and <i> were brought back and can be used when needed, though they have no semantic meaning and CSS is more encouraged.
* Attributes can be minimized and values do not always need to be quoted.

It seems many programmers think that HTML itself is a forgiving programming language, anyone can write a functional website without being too worried about little syntax errors. Because the web has developed so fast, strict rules to write up a website seem unnecessary and a waste of time to follow and double-check everything. Strict rules also doesn’t offer any more descriptive or semantic meaning the way new HTML5 elements do.

There might be tools and programs that help developers code their websites, maybe sometimes errors occur and if the rules are too strict, the websites will be unusable even though the errors aren’t that big.

Instead of pushing too much responsibility on web developers, today it seems web browsers have certain default rendering ways that can render websites similarly even though no strict rules are applied.

## 

## 