Every website consists of three pieces working together to produce the page:

* **HTML** (HyperText Markup Language) defines the *content* of the page - it defines the text that should appear, and the logical structure of that text. Think of it as the skeleton of your web page.
* **CSS** (Cascading Style Sheets) defines the *layout* of the page. It's here that we turn our bare-bones skeleton into something beautiful. While HTML defines that some piece of text is a heading, it's CSS that says headings should be displayed in bold red.
* **JavaScript** is a programming language that allows dynamic *behaviour* on the web page - things that happen in your web browser (e.g. menus that pop up when you hover over them, and much more...). It's what brings static websites to life!

You will have already come across these, but now it is time to look in a little more detail.

Example

As a demonstration, here is the same small page, built up from the three pieces:

HTML

The basic structure is in place

HTML + CSS

The content is styled

HTML + CSS + JavaScript

The content is interactive

([See the code](https://codepen.io/alex-jukes/pen/BmBqYO))

Online Resources

he [Mozilla Developer Network](https://developer.mozilla.org/en-US/) has a huge number of resources for web development, which are all free and openly editable. This will likely become your first stop for any question about HTML, CSS or JavaScript.

Since these are all enormous topics, we won't cover them in detail, but MDN provide a good series of tutorials.

HTML

As a primer, read [Getting Started with HTML](https://developer.mozilla.org/en-US/docs/Learn/HTML/Introduction_to_HTML/Getting_started)

Feel free to read as many of the other HTML courses as you feel capable of, but you should definitely cover the basics of:

* [Text](https://developer.mozilla.org/en-US/docs/Learn/HTML/Introduction_to_HTML/HTML_text_fundamentals)
* [Links](https://developer.mozilla.org/en-US/docs/Learn/HTML/Introduction_to_HTML/Creating_hyperlinks)
* [Images](https://developer.mozilla.org/en-US/docs/Learn/HTML/Multimedia_and_embedding/Images_in_HTML)
* [Tables](https://developer.mozilla.org/en-US/docs/Learn/HTML/Tables/Basics)
* [Forms](https://developer.mozilla.org/en-US/docs/Learn/HTML/Forms/Your_first_HTML_form)

CSS

Don't worry too much yet about making elements look good or perform actions - those will come once you've learned more about CSS and JavaScript.

If you are totally unfamiliar with CSS, start off by reading the [Introduction to CSS](https://developer.mozilla.org/en-US/docs/Learn/CSS/Introduction_to_CSS/How_CSS_works).

Now you've got the basic idea, follow through the CSS guides making sure you at least cover

* [Syntax](https://developer.mozilla.org/en-US/docs/Learn/CSS/Introduction_to_CSS/Syntax)
* [Simple Selectors](https://developer.mozilla.org/en-US/docs/Learn/CSS/Introduction_to_CSS/Simple_selectors)
* [Pseudo-classes](https://developer.mozilla.org/en-US/docs/Learn/CSS/Introduction_to_CSS/Pseudo-classes_and_pseudo-elements)
* [Combinators](https://developer.mozilla.org/en-US/docs/Learn/CSS/Introduction_to_CSS/Combinators_and_multiple_selectors)
* [Values and Units](https://developer.mozilla.org/en-US/docs/Learn/CSS/Introduction_to_CSS/Values_and_units)
* [Cascade and Inheritance](https://developer.mozilla.org/en-US/docs/Learn/CSS/Introduction_to_CSS/Cascade_and_inheritance)
* [The Box Model](https://developer.mozilla.org/en-US/docs/Learn/CSS/Introduction_to_CSS/Box_model)

This should give you a fairly abstract understanding of how CSS works (don't worry if you don't fully understand all the details yet), but doesn't include many of the properties required to *actually* lay out your page.

There are a *lot* of these - [about 500 or so](https://developer.mozilla.org/en-US/docs/Web/CSS/Reference), and you are certainly not expected to memorise them all!

The continuation of the MDN tutorial covers various groups of properties, which you should read as appropriate when tackling the main exercise. Once you understand the syntax and model, the best way to learn is to read about different selectors & properties is as you need them - over time they will become more familiar, but even experienced developers will need to look up the details from time to time!

Developer Tools

When working with HTML/CSS/JS, your browser developer tools are invaluable!

Go to a website (like [this one](https://www.softwire.com/)) and open up the Developer Tools. You can do this on most web browsers by right clicking somewhere on the page and selecting 'Inspect', or using an appropriate shortcut (usually F12 or Ctrl-Shift-I).

You should see a side bar pop up with lots of HTML and CSS in it. This shows the HTML structure of your page, under 'Elements' (or 'Inspector'), it can be navigated by hand, or use the element picker to select an element on the page.

Once selected, the tools will show you all the CSS rules being applied to the chosen element on the right hand side:

This can let you diagnose exactly why an element is (or isn't!) looking a certain way. Even better, you can actually edit both the HTML and CSS using the developer tools - allowing you to rapidly tweak the layout, before going back and modifying the original.

Other useful features include:

* Console - the JavaScript console will print any log messages or errors
* Sources (Debugger in Firefox) - will show you the scripts running on your page, and even let you set breakpoints and debug
* Network - shows you all the HTTP requests being made by the browser, allowing you to diagnose broken requests

Exercise

Build yourself a web page to tell the world a bit about yourself and your coding journey. Try to use a range of different elements:

* A list of programming languages you're learning or would like to learn
* Information about something you're interested in
* Your favourite foods, ranked in an ordered list
* Links to some websites you think are well designed
* A picture of yourself (remember to include an alt tag so it'll be accessible to anyone using a screen reader!)

Use [GitHub pages](https://pages.github.com/) to host your site for everyone to see.

Further reading

**Explore MDN**

Work through some of the other MDN tutorials and documentation. Particularly if you already have some experience of web development - there are plenty of advanced topics to cover.

**Flexbox**

One particularly important set of properties is using [Flexbox](https://developer.mozilla.org/en-US/docs/Learn/CSS/CSS_layout/Flexbox). This lets you arrange your box elements in a very flexible way (see what they did there), once you've got the hang of it, it is a very powerful way of producing different layouts.

Despite being around for a very long time, flexbox still has [numerous bugs](https://caniuse.com/#feat=flexbox) in Internet Explorer - be sure to test carefully if you need to support it.

There is a great [Flexbox Reference](https://css-tricks.com/snippets/css/a-guide-to-flexbox/) which shows exactly how all the properties work.

**Other challenges**

For a further challenge, try the [Wikiversity CSS challenges](https://en.wikiversity.org/wiki/Web_Design/CSS_challenges). These rapidly get pretty difficult, but would be well worth coming back to over time if you end up working with a lot of CSS.