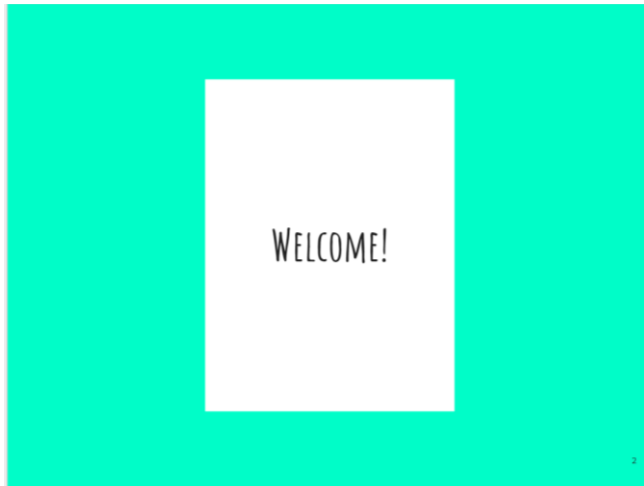


CodeFirst:Girls Beginners Coding course - Front end Web development

Speaking notes: week 1

Introduction - Course Overview

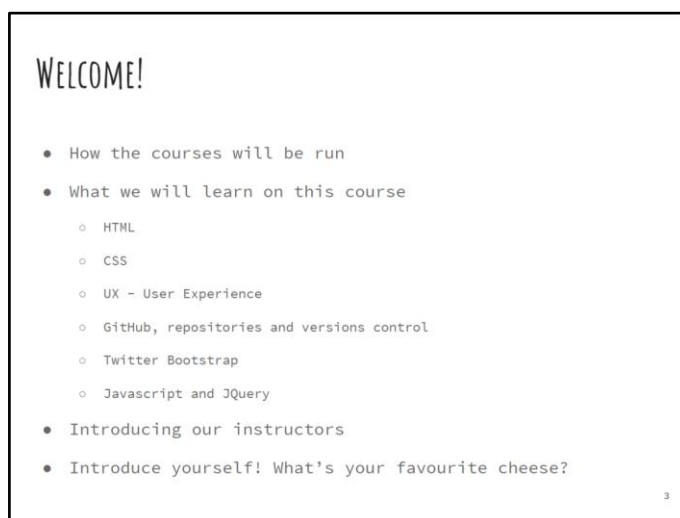


Welcome

Lovely to meet you all, and a very warm welcome to your CodeFirst:Girls Front end web dev coding course! We're really excited to have you here and over the next 8 weeks you will learn about...

- How websites are made
- How to put up your own web page
- Basic skills and tools that will be useful for any coding you do

We hope that you are curious about how the internet works and that this course will make you look at websites in a different way.



How the course will be run

This course will be run over 8 weeks. Your instructors will be here with you to help you through the course, and by the end of the course you will have created your very own personal website as well as one you will make in a group.

The hardest thing about learning to program is knowing where to start and what to learn. With this course the aim is to provide a basic overview of the technologies that are used to create websites, along with some of the tools and resources for you to discover more.

The focus of this course is learning the basics of how and why things work and to provide the basis to build upon in future courses. We'll be covering a lot of content, but we won't always be able to cover everything in complete detail. That's one of the exciting things about technology - there's always more to learn, but hopefully we'll help you learn enough about these technologies to get you excited about what you can learn and have an amazing project to show off at the end.

Classes will be as hands-on and practical as possible. There will be a number of tasks to do in between the classes to reinforce what you have learnt. It's up to you whether you do the tasks or not, but the more you put in the more you will get out of it. If you are ever in doubt, Google it, check StackOverflow, ask the person next to you, or ask one of us. **Remember - there's no such thing as a stupid question!**

Your course notes are at times wordy for the sake of reference, and not everything will be covered in class. The point of the notes is for you to have a useful place to come back to or remind yourself of the things you have learned in class.

We will also be using some slides to present as well as taking you through some demo's (demonstrations) and examples. You might occasionally find that what you see on your

screen can vary from the screenshots on the presentation - this can be for a few reasons including if you have different versions of software such as for a Windows or an Apple computer. Your instructors will help you through these differences where you need.

[Notes for instructor]

Now as you go through your notes you'll find that tasks and **direct instructions for the instructor will be in bold blue like this!**

What will we learn on this course?

The areas we will cover include:

- HTML
- CSS
- UX - User Experience
- GitHub, repositories and versions control
- Twitter Bootstrap
- Javascript and JQuery

This will help you understand

- How websites are made and how the internet basically works,
- How to create and publish your own webpage,
- How to code in groups and contribute to the developer community,
- Skills & tools for coding and carrying on after the course.

Introducing our instructors

The people who will be supporting you through the coming days are your instructors! They are all here to help you learn, and also great people to speak to about what a career in the tech industry is like and why they find it interesting and exciting.

INSTRUCTOR TASK

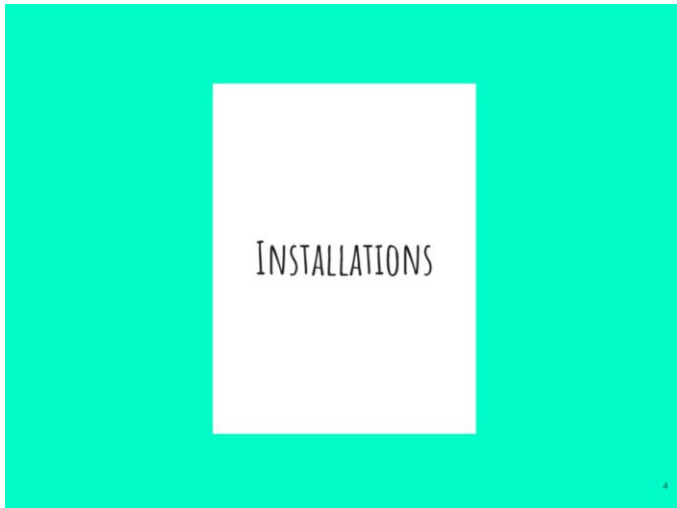
Each instructor should introduce themselves including their names, what they do (working, academic, student), and then a line about what this means

Introduce yourselves!

And now the most important part of the introduction - we would like you to introduce yourselves! So if we could go around the room, and everyone gives us their name and what they are studying, and because we always like to have something fun to kick us off, if you were a biscuit, what type of biscuit would you be?

INSTRUCTOR TASK

Get the students to introduce themselves as above



Installations

Now before we get started, we just want to check that you have all the right tools. So you'll need to make sure that you have downloaded your course documents, as well as have a few programs installed on your computer, and you'll also need to have a GitHub account.

Now, let's check that you have the right software installed and signed up for that GitHub account!

INSTALLATIONS

- Check that you have the following on your computer
 - The text editing programme called 'Atom' [<https://atom.io/>]
 - The web browser 'Google Chrome' [<https://www.google.com/chrome/browser/desktop/index.html>]
- Register yourself for a 'GitHub' account
 - Go to <https://github.com>
 - Download the GitHub desktop client application
- Download your course documents (PDF format)

If you don't have any of the above or having any issues,
just let your instructor know!

So why do you need these programmes?

Atom

A Text Editor is a Graphical user interface (GUI) that is built for writing and editing code that can be processed directly by the computer.

We have chosen Atom as our recommended text editor because it:

(a) gives a good user experience on mac, linux and windows; (b) is easy to get started with; and (c) can be easily customised when you get more advanced.

Google Chrome

Chrome is a free web browser provided by Google. A web browser is a piece of software that you can use to 'open' and run the websites you create or call on through the internet. We use Chrome because it comes with a good set of developer tools that we will be using over the course. If you do use another web browser, please use chrome for these sessions as some tools are different other browsers.

GitHub

GitHub is, informally, a code sharing and publishing service, and a developer network. Formally, it is a web-based repository hosting service for a version control system (that tracks file version changes) called Git (more later). You can also host simple static websites via GitHub using GitHub pages. More on that later.

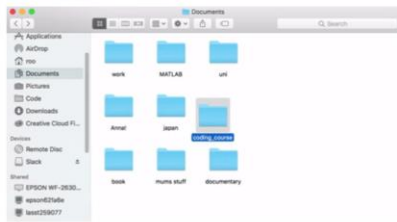
To make use of it, you will need to register for a GitHub account and then download the desktop client application for GitHub onto your computer. There are other ways of using and navigating GitHub, but we will be using the desktop client application because it's a more straightforward way of getting started with the tool.

If you don't have any of the above or having any issues, just let your instructor know!

INSTRUCTOR TASK

- **Go round the room and check that all students have Atom downloaded and installed on their laptops**
- **Also check everyone has a GitHub account and that they have confirmed their email address - Github will have sent them an email to do this.**

TASK: CREATE A CODING _COURSE FOLDER



In your documents, create a folder called 'coding_course'. This will be where you will keep all the websites you will be making over the next few weeks.

When you download files from github, make sure you move them into here, so they are all in one place and easy to find!

© CodeFirstGirls 2018

6

TASK

This is a task for all the students to do at the same time.

We are going to create a folder for all the website projects we are going to build on this course.

Let's create a folder

1. On your laptop, create a folder called **coding_course**. This folder is going to contain all your website projects for this course. Put it somewhere you can't find it easily. (There is a handy gif to show you how)



6

Introduction to Coding

Before we start to make our websites, it is worth taking some time to think about how the web works, and look at the big picture of how websites are made.

When we talk about “code”, “source code”, this is just a collection of computer instructions written using some language that is readable to humans. It’s important to note that the purpose of coding is to manipulate data; which is the storage of information on your computer.

What is Coding?

One of our key objectives will be to teach you how to create your own online website landing page. Therefore, we will be focusing on a two specific coding languages HTML - which is a markup languages, and CSS which is a style sheet language. Writing code in a these languages only allows you to create static content without interactivity.




When code allows a computer to do more than just display static information (so where we need to do calculations or have some interactivity), we need to use a Programming language. For our purposes, we will only be at one programming language in this course - **JavaScript**. Here’s one way of thinking of this - programming languages allow you to carry out mathematical equations in a program, and not all types of code are meant for this purpose (“Programming” is a subset of “coding”).

Another way to classify web coding languages is by defining them as either **“Front End”** (or client side) web coding or **“Back end”** (or server side) web technologies. Our key focus will be on Front-End Web Technologies, mainly HTML, CSS, Twitter Bootstrap an important web framework, and Javascript in the form of a framework, jQuery.

Front-End Web Development is about creating and styling the parts of a website, web service or application that users interact with (as opposed to the processes that happen

behind the scenes; “back-end web development”). Those who are skilled in Front-End coding are able to plan and visualise how they want a website to be laid out. A good eye for design and an interest in the user journey helps too.

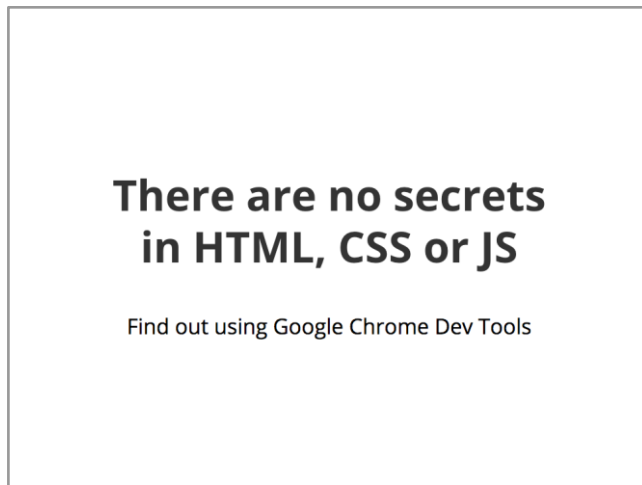
For this reason, Front-End technologies are also referred to as client-facing, or user-facing, technologies.

HTML <i>HyperText Markup Language</i>	CSS <i>Cascading Style Sheets</i>	JavaScript
		
a markup language	a stylesheet language	a programming language
Describes the structure of web pages.	Describes the presentation of web pages, including colors, layout, and fonts. It enables responsive design ; for one to adapt the presentation to different types of devices.	Allows user interactivity, and enables web pages to be dynamic. (Usually without needing to reload the page)
HTML documents are described by HTML tags , and each tag describes different element.		*It is a type of programming language, scripting .

An Overview of Different Web Technologies

- HTML - HyperText Markup Language
 - A markup language describes the structure of web pages.
 - HTML documents are described by HTML tags, and each tag describes different element.
- CSS - Cascading StyleSheets
 - a stylesheet language
 - Describes the presentation of web pages, including colors, layout, and fonts.
 - It enables responsive design; for one to adapt the presentation to different types of devices.
- JavaScript - a programming language
 - Allows user interactivity, and enables web pages to be dynamic. (Usually without needing to reload the page)

- It is a type of programming language, scripting.



There are no secrets on the web

You can look at the HTML, CSS and JS of every website on the web. It's a useful feature of all browsers and it's how many web developers learn new tricks every day. You will be using this feature a lot during this course. If something is not working as you expected, the developer tools will show you why.

DEMONSTRATION

Give a demonstration of the Web Dev tools in Chrome. go to <http://www.codefirstgirls.org.uk> in Chrome for this

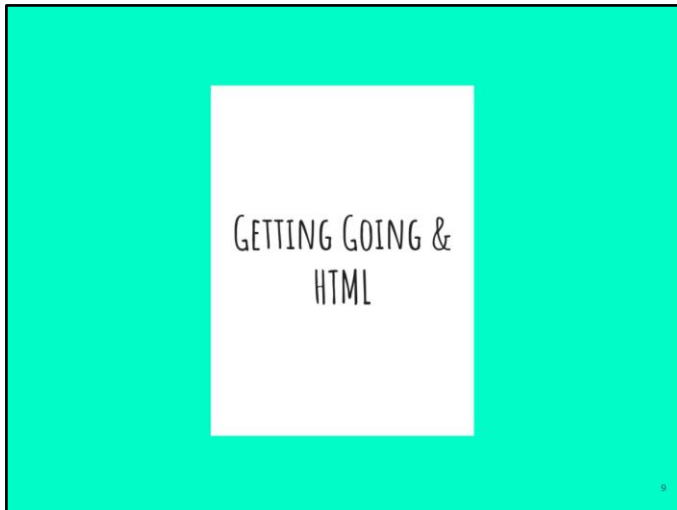
- Open the Code First: Girls website in your browser
- To view the source code of the website you can do one of the following:
 - ◆ View > Developer > View Source
 - ◆ Tools > View Source
- Click on the square with a pointer at the far left of the Developer Tools box to hover over bits of the page to find the related HTML.
- Hover over the HTML code in the Developer Tools box and see different parts of the page highlighted

- On the right hand side of the Developer Tools box you can see the CSS. Changes you make here will be instantly visible on the page, but only for you. As soon as you refresh the page the change will be gone.
- For example: click on a `<p>` element in the HTML section.
- In the CSS column and find a section that begins with `p {`
- This section includes all the styles p elements will get. Click inside that box and a new line will be created.
- Now type: `color: red;`
- Providing the colour of paragraph text wasn't already red, it will be now.
- Next find an image on the page by using the square with pointer button in the toolbox. Once you have selected an image the highlighted HTML will begin with `<img`
- In the CSS column find a section that begins with `img {`
- Click inside the box and on the new line type: `display: none;`
- All the images on the page will now have disappeared.
- Undo all your changes by refreshing the page.

This works for any website on the web. Have a play and see what you can change. You can ask us about specific CSS you want to try.

The Developers Tools contain other gems you might find handy at some point.

Under the Sources tab you will find all the files and resources used on the website you are viewing. It will include all the files for the CSS, JavaScript, images, etc



Getting going & HTML

WHAT IS THE INTERNET AND HOW DOES IT WORK?

Internet	Web
HARDWARE	SOFTWARE
<ul style="list-style-type: none">• A large network• A lot of computers connected to each other• servers	<ul style="list-style-type: none">• The system you use to access the internet• Browser, email, instant messaging, ...• clients

© CodeFirstGirls 2018

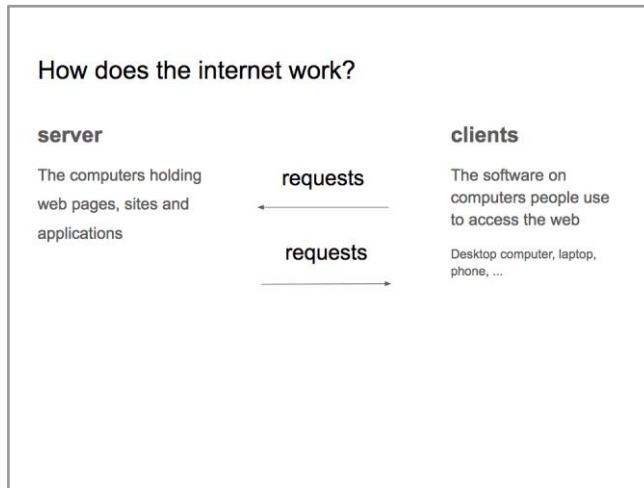
12

What is the internet and how does it work?

Before we dive into coding it is worth thinking about how the websites we are coding are shared online. How does someone find your website? How do we publish our websites for other people to find? We won't be testing you on these questions, but they will help you understand what we are going to learn.

The internet is a large network of computers connected to each other. All these computers are called **servers**, and they hold all the files that make up all of the websites.

The web is the software that allows the devices people use to access the internet. This can be a browser, email, instant messaging, etc. This software is called **clients**.



And how the internet works is that a client requests information from a server and a server responds by giving the client access to what they requested.

For example if you want to know the latest news headlines, you can use the browser on your desktop computer to find out.

1. You type in the web address of a news website - the BBC for example.
2. Your browser is the client and it will send a request to the internet to see the BBC website.
3. The request will go to the server (i.e. the computer) that stores the the BBC website.
4. If all goes well, it will grant your request and your browser will show you the latest news on the BBC website.

Two types of website

Static

- Pre-prepared
- Always same

Dynamic

- Generated on the fly
- Pulls data from a database

E.g. facebook.com

Static vs Dynamic sites

Websites on a server can be either static or dynamic.

A **static** site is a website where all the pages of the website contain code that is fixed and thus doesn't change unless the code is amended on the server. The websites we are going to build will be static websites.

A **dynamic** site is a website where the content on the pages is generated on-the-fly by pulling information from a database depending on what the user has asked for. An example is the Facebook feed. The feed is different for every user profile.

Putting up a website

Hosting

A server to host your website

Domain

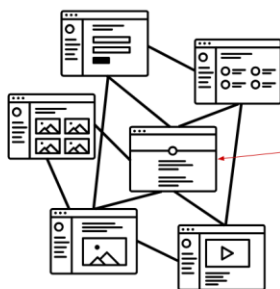
A domain name to point towards the server

Putting up a website

If you want to put a website online you will need two things:

- A web server where you can store your website files
- A domain name (your url) to point towards the web server

What makes a website



Website

- A collection of linked web pages
- The home page is **index.html**

Icons by KAPKALAM from the Noun Project

What makes up a website?

A website is a collection of linked web pages that can be viewed in a browser. Each page is written in a markup language, HTML (**Hyper Text Markup Language**), and can contain text, video, images, links, etc.

The first page of the website, this is the point of entry, is referred to as the home page or landing page. The file name for this page must always be **index.html**.

All the files that make up a website are contained inside a folder and are stored on a server. This folder will contain all the website pages but also other technologies to control what the end-user sees.

For example:

- **CSS** (Cascading Style Sheets) files will control how the website looks
- **Script files** like JavaScript will add interactivity to a website
- **Media files** (images, video, music, etc) will add interest to your website

Everything your website needs to function needs to be inside your website folder.

An example of good practice is to organise your folder in subfolders according to file type. So you would have a **css** folder for all your CSS files, a **js** folder for script files, an **images** folder for all your images.

The components of a web page		
HTML	words	the backbone
CSS	style	how it looks
JavaScript	scripts	how it interacts
Images, video, audio, etc.	media	entertainment

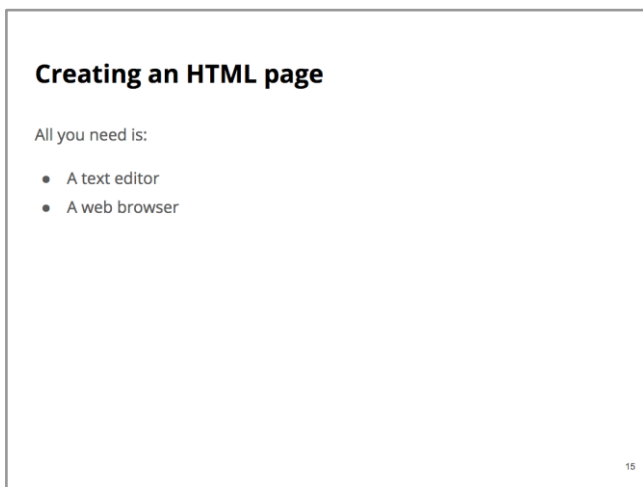
The components of a web page

All the different components that make up a website will have a different effect.

- The backbone of the website is made up of **HTML**.
- The **CSS** will style the web page and affect how it looks.

- **JavaScript** will take care of interaction on a website.
- **Media** files like images, video, audio etc, will bring the fun element to a website and make it look appealing.

Let's take a look at how this works in a real example.



HTML

Creating a basic HTML page

So far we have learned about:

- the difference between the internet and the web;

- what happens when you type a website address in your browser;
- the elements that make up a website;
- and how you can have a sneak how websites have been put together.

Now we are going to learn how to code our own websites. One of the nice things about HTML is that you don't need any fancy software to test it out on your computer or laptop. All you need is a text editor and a web browser.

HTML ELEMENTS

<h1>Hello World codepen demo</h1>

```
<h1>Hello world</h1>
<p>This is my first website!</p>
```

© CodeFirst Girls 2018

28

INSTRUCTOR TASK

DEMO - go to the codepen on the slide

<https://codepen.io/rosie934/pen/RYXYar?editors=1000#>

This is our first line of HTML. HTML stands for HyperText Markup Language, and as it says, it marks up the content of a web page. It does this by using a predefined set of elements for different types of content. The elements define the semantic meaning of their content.

As you can see you have written an open tag `<h1>` and a close tag `</h1>`. They tell a browser that all the text in between these tags are a heading, and the browser displays it as such. Let's have a look at the basics of HTML.

Write out a couple of elements on the demo, such as `<h1>` and `<p>`

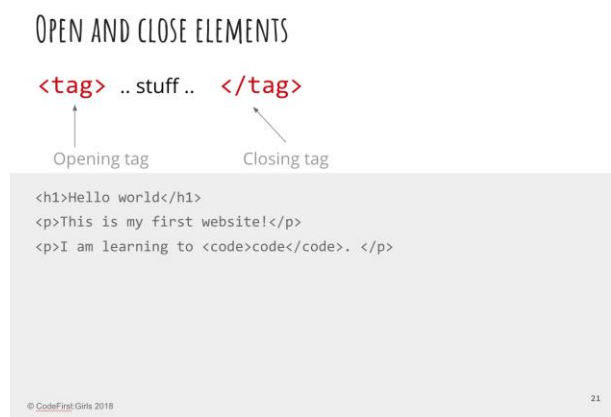
Elements

HTML elements contain one or more tags to express their content. A tag always starts with the less-than sign, <, and always ends with the greater-than sign, >.

There are two types of elements:

- Open and close elements: all the content sits between an open and a close tag
- All-in-one elements: all the content is contained within one tag

Examples of elements:



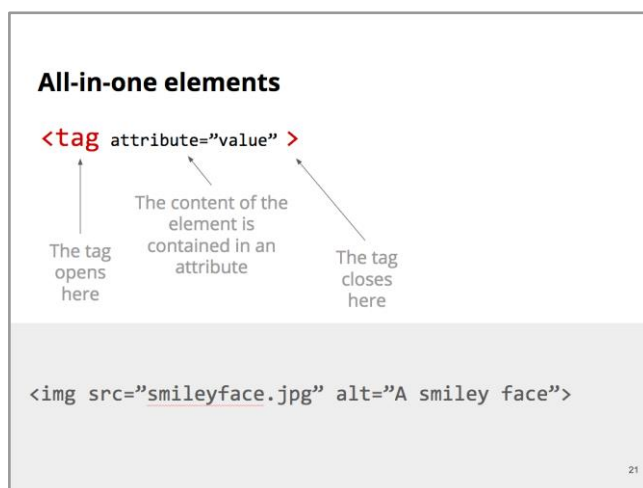
Open and close elements

Open and close elements are made up of three parts:

1. An open tag starting with the less-than sign, then the name of the tag, and close with the greater-than sign. `<tag-name>`
2. The content you want this tag to contain
3. A close tag starting with the less-than sign, then a forward slash, /, to indicate this is the closing tag, then the name of the tag, and close with greater-than sign. `</tag-name>`

Some examples of open and close elements are:

- **Headings:** <h1>../h1>, <h2>../h2>, <h3>../h3>, <h4>../h4>, <h5>../h5>, <h6>../h6> - each heading after h1 becomes slightly less important.
- **Paragraphs:** <p>../p>
- **Emphasizing text:** ../em>, traditionally displayed as italics
- **Giving extra importance to text:** ../strong>, traditionally displayed as bold text
- **Divisions:** <div>../div> to group together website parts that belong together.



All-in-one elements

An all-in-one element will contain the additional information a browser needs to display its content. The all-in-one element is made up like this:

1. The element is opened with the less-than sign and then the name of the tag
<tag-name
2. The content of the element is contained inside an attribute. An attribute itself contains two parts, a name and a value. Tags can have more than one attribute, but we'll get to attributes in more detail in a minute. **attribute="value"**

3. The element is closed with the greater-than sign. >

An example of an all-in-one element is the image tag.

- The **src** attribute contains the filename of the image. You need to make sure the file name is exact and needs to include the file extension.
- The **alt** attribute is optional, but good practice to include. It contains a description of the image. Think of it as how you would describe the image to someone over the phone. The alt text is read out by screen reader software used by people who have a visual impairment and improves the usability of a web page for them.

Attributes

`<tag attribute="value" >`

No spaces on either side of the = sign

Quote marks surrounding the value of the attribute

```
<div class="info-section">

<a href="http://google.com">
```

22

Attributes

All HTML elements, even open and close ones, can contain attributes. If an open and close element has attributes they are always added to the open tag.

An attribute is made up of two parts:

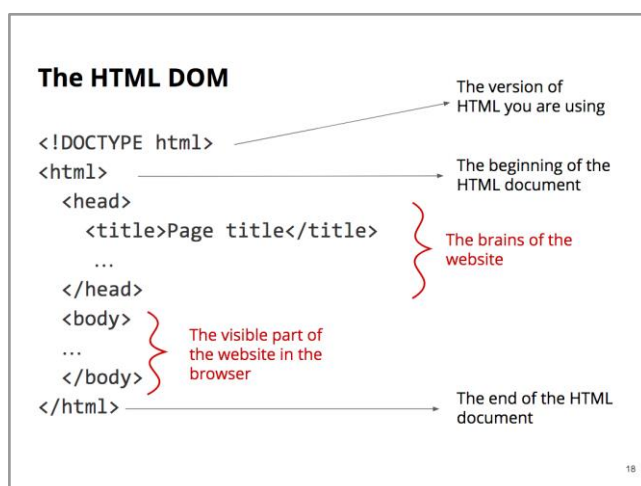
1. The name: this indicates what the attribute is about
2. The value

Examples are the `img` element we saw earlier. Another example is the `link` element. The `link` element creates a link to a different web page. It's an open and close element.

```
<a href="http://google.com">Visit Google</a>
```

```
<a href="about.html">About us</a>
```

- The element tag name is **a** and stands for anchor
- The **a** open tag contains the **href** attribute. This stands for Hyperlink Reference. You can link to other websites or another page on your website. If you link to another website you always need to include the full url. If you link to another page on your website you need to include the correct path.
- The text in between the open and close tag is what you will see in a browser.



HTML Basics

The HTML DOM

The Document Object Model (DOM) specifies the hierarchical layout of the HTML document. It is platform and language independent and interacts with any HTML document.

It is loaded in the browser and represents the document as a node tree, with each node representing a part of the document. All these nodes tell the browser where to look for a specific thing. The nodes are useful for software development as you will find out when we learn CSS and JavaScript.

Every HTML document requires the following layout:

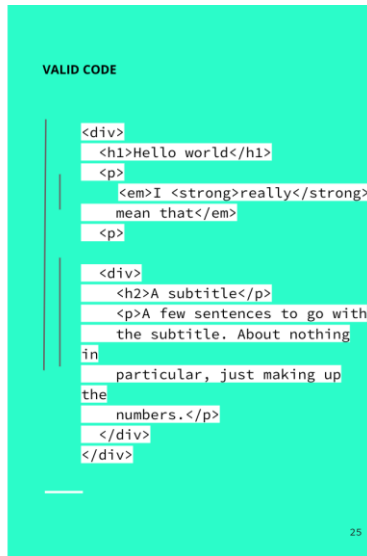
- **<!DOCTYPE>** tells you what version of html you are using. For HTML5 and beyond it is simply html. Previous versions required a lot more.
- Everything on your web page is wrapped inside the **<html>** and **</html>** tags. The first one opens the website, and the second one closes it.
- The **<head>..</head>** section contains the brains of a website. Everything in here is information the browser needs to display the web page. For example additional resources like CSS files will be linked to from here. In theory nothing added in this section should be visible in a browser window.
- The **<title>..</title>** tag in the **<head>** is title of a web page and is displayed at the top of the browser window in the browser tab. This title is also used by Google search on the search results page.
- The **<body>..</body>** section is the visible part of your website. Everything you put in here will be shown in a browser.

Elements, Tags & Attributes

GOOD CODE PRACTICE

- Always indent nested code.
- The last tag you opened is the first tag you close.
- Your editor will help you with this! When you make a mistake it will often highlight it in a different colour

© CodeFirstGirls 2018

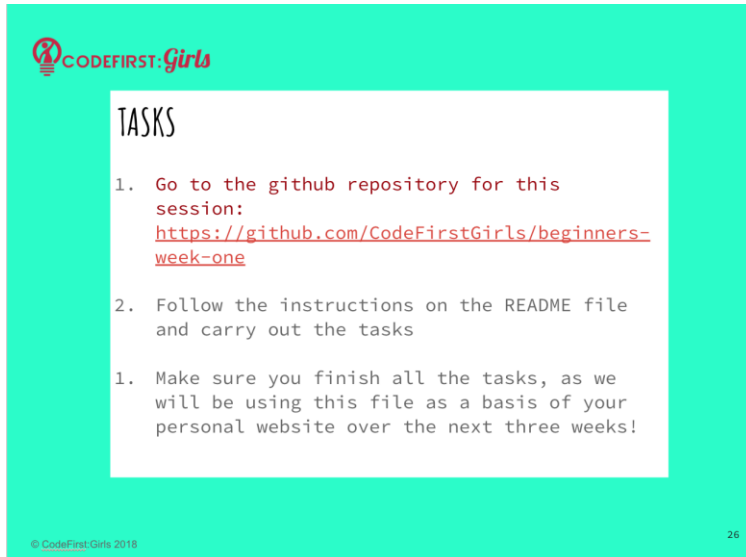


Good code practice

HTML elements can be nested, and this is usually done to group elements that belong together. To write good code you need to properly nest your open and close tags.

A rule of thumb is that the last tag you opened is the first one that needs to be closed. A good text editor can help you with this. Text editors all have an Auto Indent feature which will start a nested tag on a new line and slightly indented from the previous tag. Formatting your code like this will help you remembering which tag needs to close when.

In Atom you can navigate to File > Lines > Auto indent to start indenting your code automatically.



CODEFIRST:Girls

TASKS

1. Go to the github repository for this session:
<https://github.com/CodeFirstGirls/beginners-week-one>
2. Follow the instructions on the README file and carry out the tasks
1. Make sure you finish all the tasks, as we will be using this file as a basis of your personal website over the next three weeks!

© CodeFirst Girls 2018 26

Now let's put all this in practice and write some HTML code.

1. Go to the github repository <https://github.com/CodeFirstGirls/beginners-week-one>
2. Click on the green **Clone or download** button, and in the dropdown click on **Download ZIP**. Download the files into your coding_course folder
3. Unzip the downloaded folder.
4. Open the whole folder in your text editor.
5. Open the example.html file in Chrome and look around with the developer tools to see what is going on.
6. Follow the tasks in the README file

Week 1 Homework

Finishing off

Task: Finish the HTML exercise. Prompt students to check the solution, the link is on the slides. There is nothing wrong with looking at the solution, there's no cheating in web dev!

Preparation for next time

Task:

1. What is CSS? Get ready for the next class by watching [this fun video](#).

Extra Resources

[This video](#) talks about how the Internet works in 5 minutes

[A summary](#) of the different components of the Internet

[File organising](#) for your website

[Introduction to servers](#) by Eli the Tech Guy

An article from Mozilla's Developer Guides: [Introduction to HTML](#)

[W3 Schools HTML Tutorial](#)

[HTML Terms Glossary](#)

[HTML DOM](#)

[Web Monkey HTML Cheatsheet](#)

[Simple HTML Guide Cheatsheet](#)

[A HTML Validator that checks your HTML code](#)