HackSoc: Intro to Web

Amy Dickens - PhD Student Computer Science

Intro to Web: Session Schedule

Week One 7/3/17: Online (follow these slides)

Week Two 14/3/17: 6.30pm - 8.30pm, The Hub, School of Computer Science

Week Three 21/3/17: 6.30pm - 8.30pm, The Hub, School of Computer Science

Week Four 28/3/17: 6.30pm - 8.30pm, The Hub, School of Computer Science

Requirements:

Please bring a laptop with you to the sessions in The Hub, if you do not have your own device please get in touch at info@hacksocnotts.co.uk and we will try to arrange a loan for you for the sessions.

If you don't already have one, please sign up for GitHub account here. We will use GitHub in these workshops for version control of your projects and FREE hosting on GitHub Pages. Also install GitHub desktop.

We will be using a Text Editor to write our web code, personally I use Sublime Text but others (atom, notepad ++ or vim etc.) are fine - use what you are comfortable with.

What will we cover today?

In this first session we will cover the basics of developing a website, this will include:

- ➔ Structuring a web page with HTML
- ➔ Styling our page with CSS
- ➔ Hosting our web page on GitHub

Before we begin here is a few tips to help you follow this online lesson. . .

#Tip 1: This is an information slide

These slides will have descriptions, information and references on them. They will be used to explain concepts as we go along and provide any extra info you need.

Feel free to skip these slides if you feel you already understand the concepts discussed.

If you don’t understand something on these slides I’ll be available on the HackSoc slack to answer any questions <3

#Tip 2: This is an activity slide

I will be using these slides to provide instructions and screenshots for you.

You'll be building the project by yourself, so use these slides to follow along with the lesson and create your web page.

Do not skip these slides as you might miss an important bit of the content.

Remember you can ask me questions in the HackSoc Slack my username is @dickensa

Got it?

Ok let’s start . . .

What makes a web page?

STRUCTURE - our structure is built using HTML (HyperText Markup Language), the standard markup language for websites and web applications.

STYLING - styling is achieved using CSS (Cascading Style Sheets) a style sheet language that describes the presentation of documents.

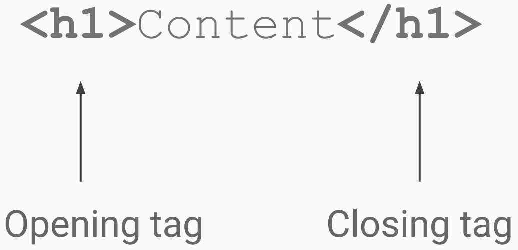
LOGIC - interactivity is created using JavaScript, a language used to program the behaviour of a web page.

HTML is just a document described with tags.

```
<h1>Content</h1>
```

Hello world

HTML is just a document described with tags.



```
<h1>Content</h1>
```

Opening tag

Closing tag

Hello world

Note: h1 is a heading tag - they make the content large and bold.

HTML Tags

HTML relies on tags to outline the structure of the content for your web page. Tags open using “< >” and close using “</>”, the name of the tag is placed between these brackets and content is placed between the opening and closing tags.

A great reference for HTML tags is htmldog.com/references/html/tags/

We are going to add some content to our page using tags to give it structure.

Create an index.html file

1. Open a new document in your text editor
2. Type in “<h1>Hello, my Name is [insert your name]</h1> ”
3. Save the document as index.html in a folder for your project
4. Open this file in your browser and you’ll see the browser reads this file to output your sentence.

index.html

```
<h1>Hello, my Name is Amy</h1>
```

Paragraph tags

1. Underneath your “Hello, my name is” header, add a sentence saying “This is my website, and my favourite society is HackSoc”

2. Enclose this sentence in a paragraph <p> tag.

index.html

```
<h1>Hello, my Name is Amy</h1>

<p>This is my website, and my favourite
society is HackSoc</p>
```

Lists - parent/child relationships

1. You can have tags inside of tags - in fact, an HTML file is just made up of a tree of tags inside of one another to explain the structure of your page.
2. The <ul> tag surrounds the entire list. Create one (don't forget to close it).
3. Create three <li> tags inside of the <ul> - each one is a list item. List your three favourite things.

Note - indentation doesn't matter, it's just nicer to read when child elements are indented inside of their parent.

index.html

```
<h1>Hello, my Name is Amy</h1>

<p>This is my website, and my favourite
society is HackSoc</p>

<ul>
  <li>Chocolate</li>
  <li>Puppies</li>
  <li>Hackathons</li>
</ul>
```

Links - attributes

We use attributes to give more information about our tags - they each have two pieces of information:

- The property we are setting
- The value we are setting it to

The <a> tag is used for links, and we use the href attribute to set the URL the browser will go to when clicked.

1. Surround ‘HackSoc’ with an <a> tag.
2. Set its href property to http://hacksocnotts.co.uk

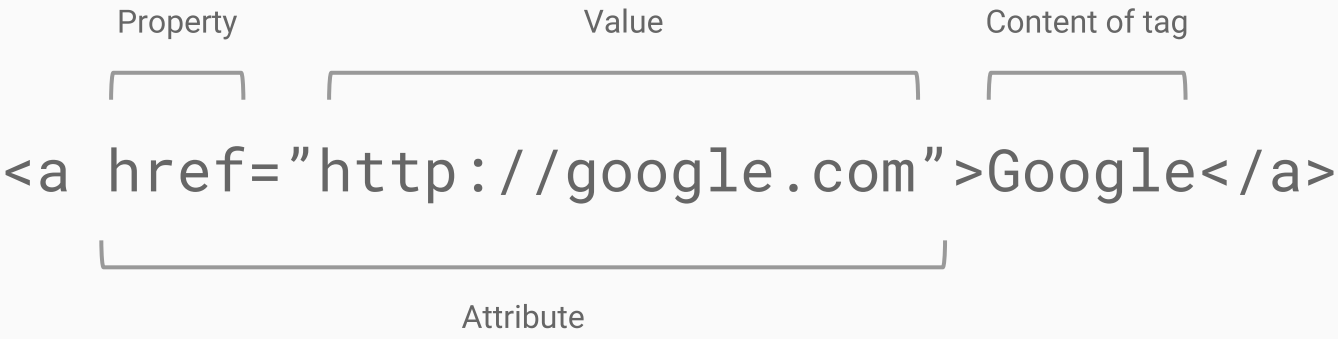
index.html

```
<h1>Hello, my Name is Amy</h1>

<p>This is my website, and my favourite society is
<a href=”http://hacksocnotts.co.uk”>HackSoc</a></p>

<ul>
  <li>Chocolate</li>
  <li>Puppies</li>
  <li>Hackathons</li>
</ul>
```

Attributes



Property

Value

Content of tag

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

Attribute

Images - the ugly duckling

The <img> tag used for images is one of only VERY few which do not need to be closed (after all, you have the image, but nothing goes ‘inside’ of it, content-wise).

1. Go to imgur and find an image URL for an image you like. The URL should have an image extension (.jpg, .png, etc) on the end.
2. You need to create an <img> tag with a src attribute. Inside of it, copy and paste the image URL

index.html

```
<h1>Hello, my Name is Amy</h1>

<p>This is my website, and my favourite society is
<a href=”http://hacksocnotts.co.uk”>HackSoc</a></p>

<ul>
  <li>Chocolate</li>
  <li>Puppies</li>
  <li>Hackathons</li>
</ul>

<img src=”http://i.imgur.com/example.png”>
```

Other Useful Tags

We can control some elements of the style of our content using html tags, these are similar to the commands you might use in a word processor:

B = bold <b>

I = Italicise <i>

U = underline <u>

Have a play around with adding these tags to your HTML document and see what they look like. Remember if you don’t close a tag it will effect everything that follows it.

What have we covered so far?

- Opening/closing tags
- <h1>, <p>, <a>, <ul>, <li>, <img> tags
- Parent/child relationships
- Attributes
- <img> tag doesn’t require closing, but it is rare in this face

Let’s talk about class and id

If we have 10 <h1> tags on our page, we may still need some way to further tell them apart. This is especially important when styling elements.

There are two special attributes - class and id - to give more specificity to your tags. Don’t worry too much about this for now, just be aware.

```
<p class=”description”> Hello there, this is a paragraph tag with a class. </p>

<h1 id=”page-title”> Unique ID is used here </h1>
```

Let’s talk about <div>

Sometimes you need to group elements in some way - <div> tags are the way we do this. They add no styling to your tag, but when combined with classes and ids, can group elements into their own blocks.

```
<div class=”blog-post”>
  <p> Hello there, this is a paragraph tag with a class. </p>
  <h1> Unique ID is used here </h1>
</div>
```

Adding more info to our HTML file

An HTML document is split into two sections - the <head> and the <body>. The <head> section contains metadata which users don’t see directly. The <body> is what we’ve been working with so far.

1. Create a <head> tag at the top of your document.
2. Move the contents of our page to contained within the body tag

index.html

```
<head>
</head>
<body>
  <h1>Hello, my Name is Amy</h1>

  <p>This is my website, and my favourite society is
<a href=”http://hacksocnotts.co.uk”>HackSoc</a></p>

  <ul>
    <li>Chocolate</li>
    <li>Puppies</li>
    <li>Hackathons</li>
  </ul>

  <img src=”http://i.imgur.com/example.png”>
</body>
```

Adding a <style> tag to our <head>

1. Create a <style> tag inside of our <head> tag.

This is where our CSS will live

<head> section of index.html

```
<head>
  <style>
     CSS GOES HERE
  </style>
</head>
```

Structure of CSS rules

```css
element {
    property: value;
}
```

Structure of CSS rules

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

How CSS selectors relate to HTML

```
h1 {
   // Properties             = <h1>
}
```

How CSS selectors relate to HTML

```
p.info {
   // Properties                    = <p class=”info”>
}
```

How CSS selectors relate to HTML

```
#header {
  // Properties          = <div id=”header”>
}
```

CSS Colors

color can be used to change text colour
background-color can be used to change the background color

Colours can be defined in a number of ways

`keyword`	One of 140 CSS color words
`#000000`	Hex code - always 6 digits
`RGB(0, 0, 0)`	Red Green Blue
`HSL(120, 100%, 50%)`	Hue Saturation Lightness

Styling our <h1> with colours

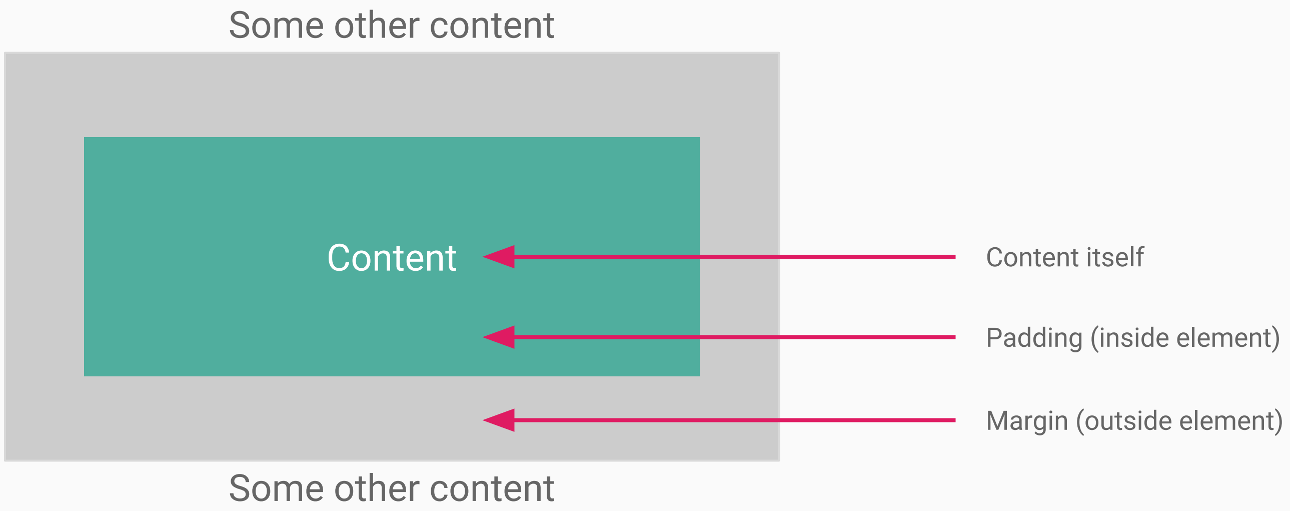
1. Set the color to red
2. Set the background to lightgrey

<head> section of index.html

```
<head>
  <style>
     h1 {
       color: red;
       background: lightgrey;
     }
  </style>
</head>
```

The box model (margin, padding)

margin can be used to add spacing outside of elements
padding can be used to add spacing inside of elements



Spacing our <h1> with padding/margin

1. Set the padding to 30px
2. Set the margin to 40px

<head> section of index.html

```
<head>
  <style>
     h1 {
       color: red;
       background: lightgrey;
       padding: 30px
       margin: 40px;
     }
  </style>
</head>
```

Fonts!

There are so many different CSS properties for fonts.

font-weight can be set to bold or normal
text-align can be set to left, right, center or justify
font-family can be set to serif, sans-serif, monospace, cursive or fantasy

We can also set custom font families, but we won’t cover that today.

Styling our <p>

1. Make font-weight bold
2. Make text-align center
3. Make font-family cursive (haha)

<head> section of index.html

```
<head>
  <style>
     h1 {
       color: red;
       background: lightgrey;
       padding: 30px
       margin: 40px;
     }
     p {
       font-weight: bold;
       text-align: center;
       font-family: cursive;
     }
  </style>
</head>
```

Time to get our site online!

We’ll be using GitHub to host our site for free.

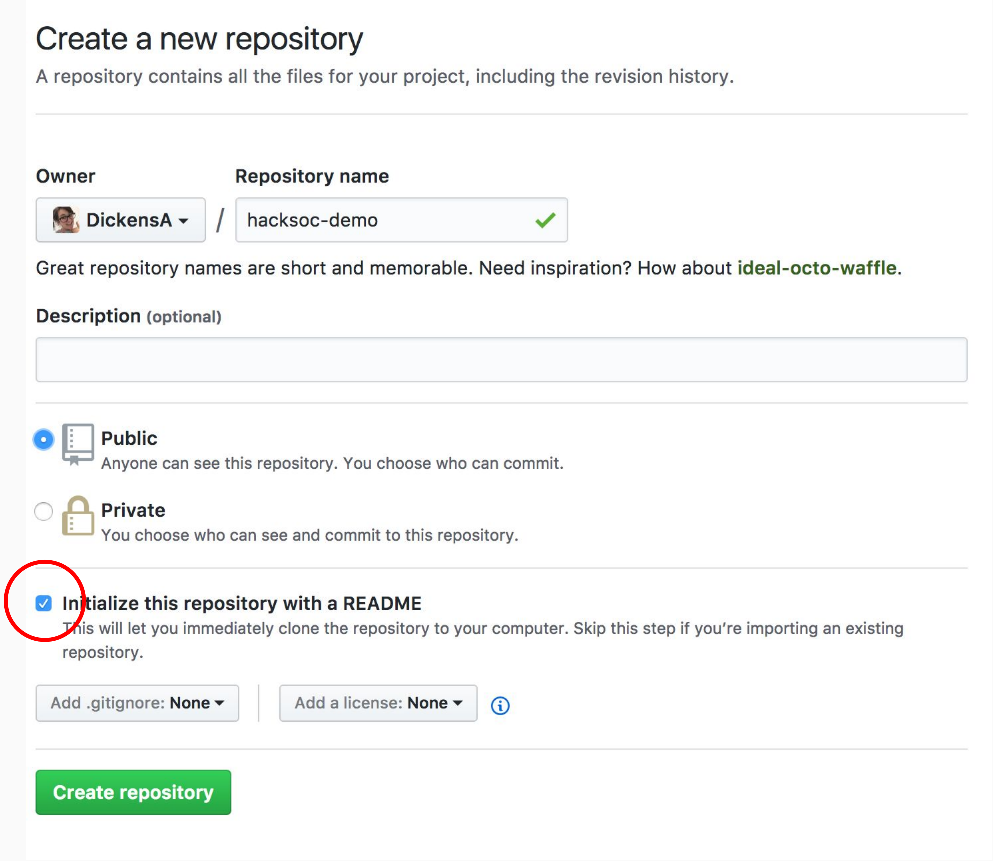
GitHub is a platform for social coding. For those who aren’t familiar with it, we’ll run through it another time, and will just use it to host our website using their drag-and-drop editor

Setup a GitHub repository for your project

1. Log in to your GitHub account (or create a new account) at github.com

2. On the right-hand side of your GitHub home page click on the 'new repository' button.

3. Fill in your project details, make sure you initialize with a README.



Create a new repository

A repository contains all the files for your project, including the revision history.

Owner DickensA ▾ / **Repository name** hacksoc-demo ✓

Great repository names are short and memorable. Need inspiration? How about **ideal-octo-waffle**.

Description (optional)

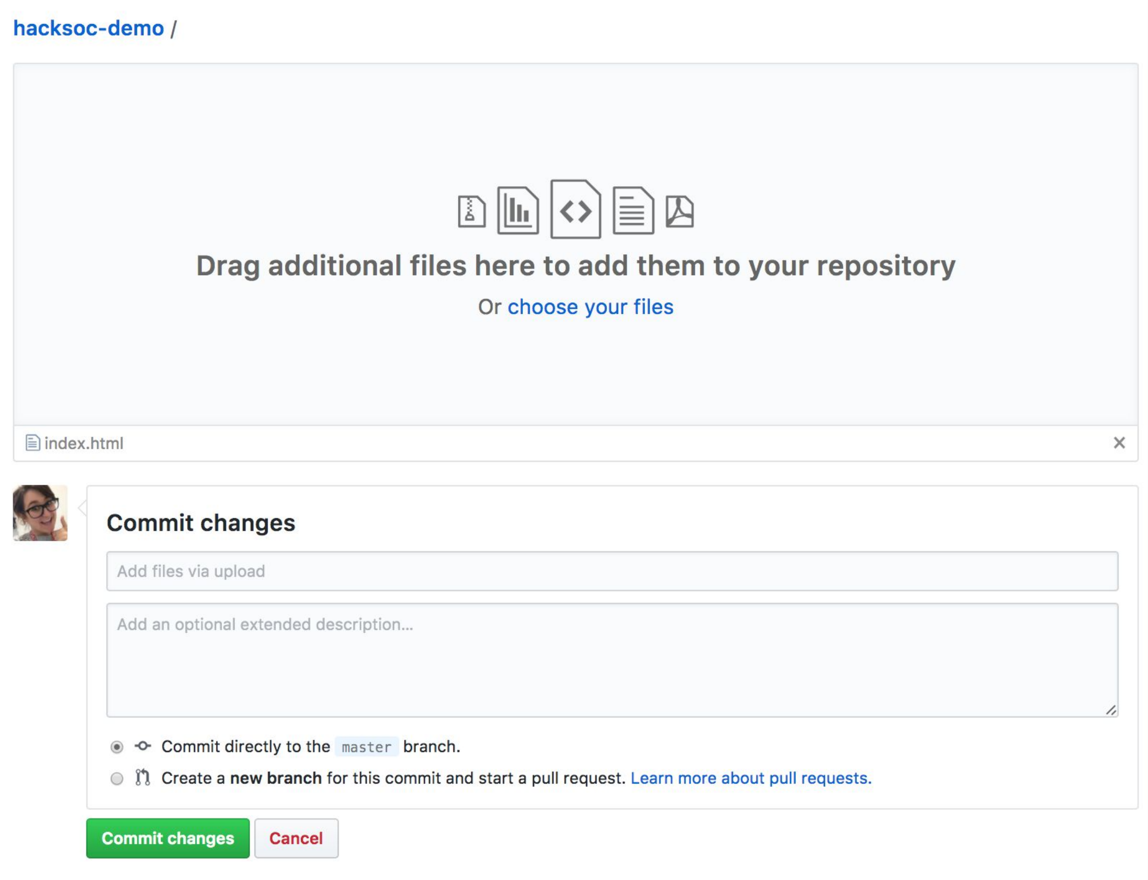
● **Public**
Anyone can see this repository. You choose who can commit.

○ **Private**
You choose who can see and commit to this repository.

☑ **Initialize this repository with a README**
This will let you immediately clone the repository to your computer. Skip this step if you're importing an existing repository.

Upload your site

1. Click “Upload files”
2. Drag your index.html on to the page
3. After uploading, click “Commit changes”



hacksoc-demo /

Drag additional files here to add them to your repository

Or choose your files

index.html ✕

Commit changes

Add files via upload

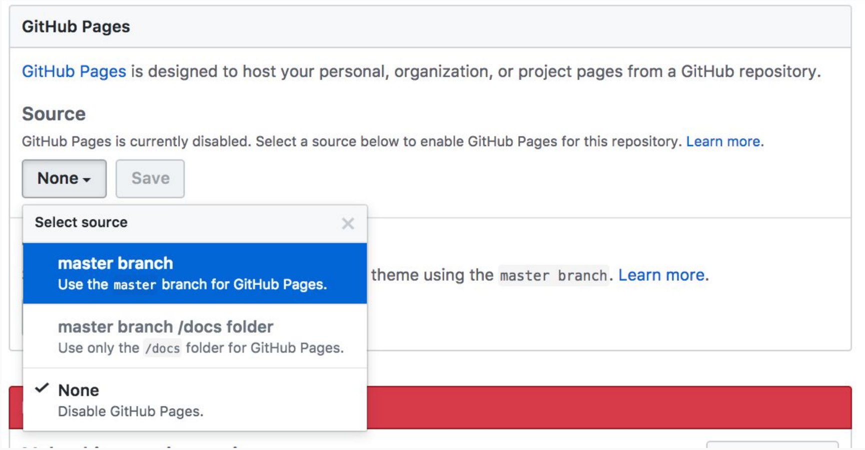
Add an optional extended description…

● Commit directly to the master branch.

○ Create a **new branch** for this commit and start a pull request. Learn more about pull requests.

Configure GitHub Pages

1. Click settings and scroll to the GitHub Pages section
2. In the Source dropdown, choose Master and save
3. Your site is now live at your URL



GitHub Pages

GitHub Pages is designed to host your personal, organization, or project pages from a GitHub repository.

Source

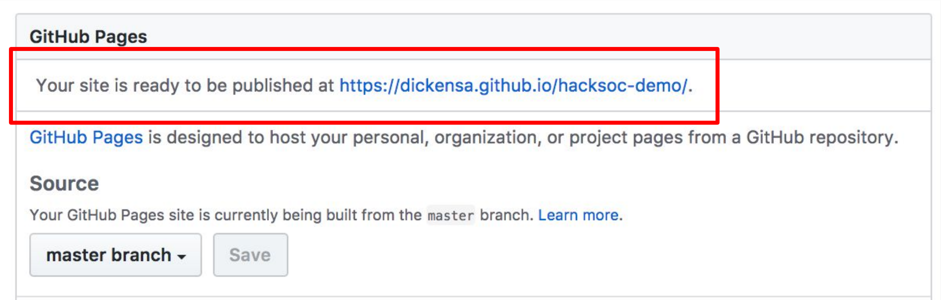
GitHub Pages is currently disabled. Select a source below to enable GitHub Pages for this repository. Learn more.

master branch
Use the master branch for GitHub Pages.

master branch /docs folder
Use only the /docs folder for GitHub Pages.

✓ **None**
Disable GitHub Pages.

theme using the master branch. Learn more.



GitHub Pages

Your site is ready to be published at https://dickensa.github.io/hacksoc-demo/.

GitHub Pages is designed to host your personal, organization, or project pages from a GitHub repository.

Source

Your GitHub Pages site is currently being built from the master branch. Learn more.

Well done <3