CSY1018
Web Development

Tom Butler
thomas.butler@northampton.ac.uk

How these sessions will work

- There will be videos online each week prior to the session
  - I will try to upload these the week before
- Please watch the videos prior to the session
  - You can also attempt the exercises before class
  - Or attempt the exercises during the sessions

How these sessions will work

- In the timetabled slot:
  - I will introduce the exercises at the start
  - You can then complete the exercises
    - I will answer questions
    - There will be some time allocated for 1:1 chats for people who do not wish to talk in the group
    - You can also ask questions in the text chat
    - I will answer questions as they are asked.
  - I will cover the solutions in the final 15-20 minutes of the timetabled slot

Your feedback is welcome

- This is a new situation for tutors and students
- If you have any suggestions I am happy to hear them. Please email me if you have any suggestions about these sessions.

Learning tips

- If you miss a session make sure you read the notes and try the exercises
- You’ll learn more by trying the exercises than just reading the notes
- Everything we cover will be useful for the assignment
- Focus on learning the concepts being taught, not the individual exercises
- Make sure you complete the exercises each week
- No solutions will be available but everything you need to complete the exercises is covered in the notes
- Put your hand up and ask questions at any time

What I expect you to already know

- Basic computer usage:
  - Navigating around the web
  - Using your operating system: Installing programs, navigating around
  - Using your file manager to locate/manage files
  - Saving files (and finding them again!)
  - Opening zip files
  - How to open the same file in different programs
  - Looking things up
    - Don't ask me questions you can type directly into google! "How do I do a loop in javascript?"

Web Development

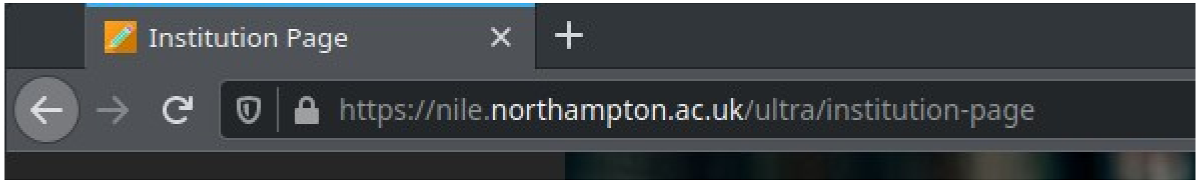
What is the ‘web’?

What is the web?

- Firstly, the web is not the internet!
  - (Though most things on the internet use the web these days)
- ***The Internet*** is a world-wide connection of computers, phones and other devices, enabling devices to talk to one another.
- ***The Web*** is one form of communication over the internet
  - Other services you might use online:
    - Game servers (These don’t use the web!)
    - Email (Some have web based front-ends, but emails themselves are not sent across the web)
    - Peer to peer and some streaming services
    - Most apps (e.g. Discord, Snapchat, Facebook) still use the web behind the scenes

What is the web?

- *The Web* is accessed through a web browser
  - Mozilla Firefox
  - Google Chorme
  - Microsoft Edge
- A web browser connects to a *web page* using a *URL*
- URL stands for *Uniform Resource Locator*
- You can see the URL of a page in the browser’s *address bar*



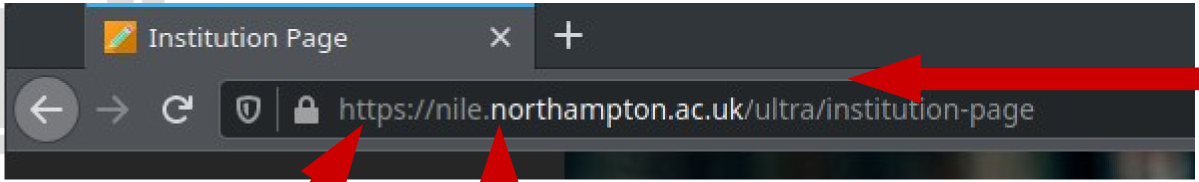
URL

- A URL on the web is *unique*. You can send someone a link and when anyone else, anywhere in the world sees the exact same page*

- * The web page can, however, display different information based on who you are, your browser, location or anything else.

URL

- A URL is made up of the following parts:



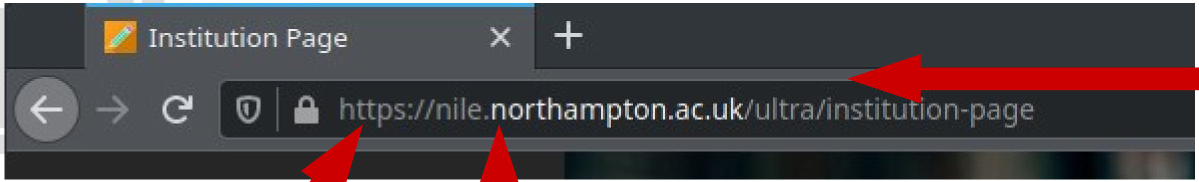
Page name starting with/
e.g. /search, /account

Domain name e.g.
facebook.com
google.com
youtube.com

Protocol (usually https)

URL

- A URL is made up of the following parts:



Page name starting with/
e.g. /search, /account

Domain name e.g.
facebook.com
google.com
youtube.com

Protocol (usually https)

URLs

- When you connect to a web page via the URL, all the information that makes up the page is downloaded to your computer
  - The text of the page
  - Any images on the page
  - Information about how the page looks (Background colours, font sizes, etc)
- Your web browser stores a temporary copy of the page on your local device
- It is then displayed on the screen
- Pressing “refresh” or “reload” will re-download the page.

Pages on the web

- When you connect to a web page, you are downloading files from *someone else's computer* somewhere else in the world.
- For making a website, you can create your own files on your computer and view it in a web browser
  - When you do this, only you can see the website, nobody else can!

Web technologies

- There are three different languages which the browser understands
- We'll be covering two this term and one next term
- Those languages are:
  - HTML
  - CSS
  - Javascript

HTML

- Every web page you visit uses HTML
- HTML contains things like the text of the page, references to images on the page, describes parts of the page:
  - Headers
  - Footers
  - Sidebars
  - Images
  - Tables

HTML

- HTML stands for:
  - Hypertext Markup Language
- **HTML is much, much easier than any other language you are likely to use in your degree!**
- HTML is very limited
- It is made up of HTML *tags*
- A tag is a description of some contained data
- Each tag uses angle brackets: < and >
- Most HTML tags have a start and end tag

Writing HTML

- Before you can write HTML code, or any other code, you need a *code editor*
- There are many different tools available to you that can be used to edit code
- Windows comes with Notepad
- **Do not use Notepad for code, it will make your life unnecessarily difficult!**
- There are other editors such as Notepad++ which have been around for decades. These look very dated and don’t include a lot of modern quality of life improvements

Editors

- Recommended text editors for HTML/JavaScript development:
  - Visual Studio Code (https://code.visualstudio.com/)
  - Sublime Text 3 (http://www.sublimetext.com/3) *Not Free*
  - Atom (https://atom.io/) *Free*
- All work on Windows/Linux/OSX
- **You can install Atom on the university machines without admin rights!**

Projects

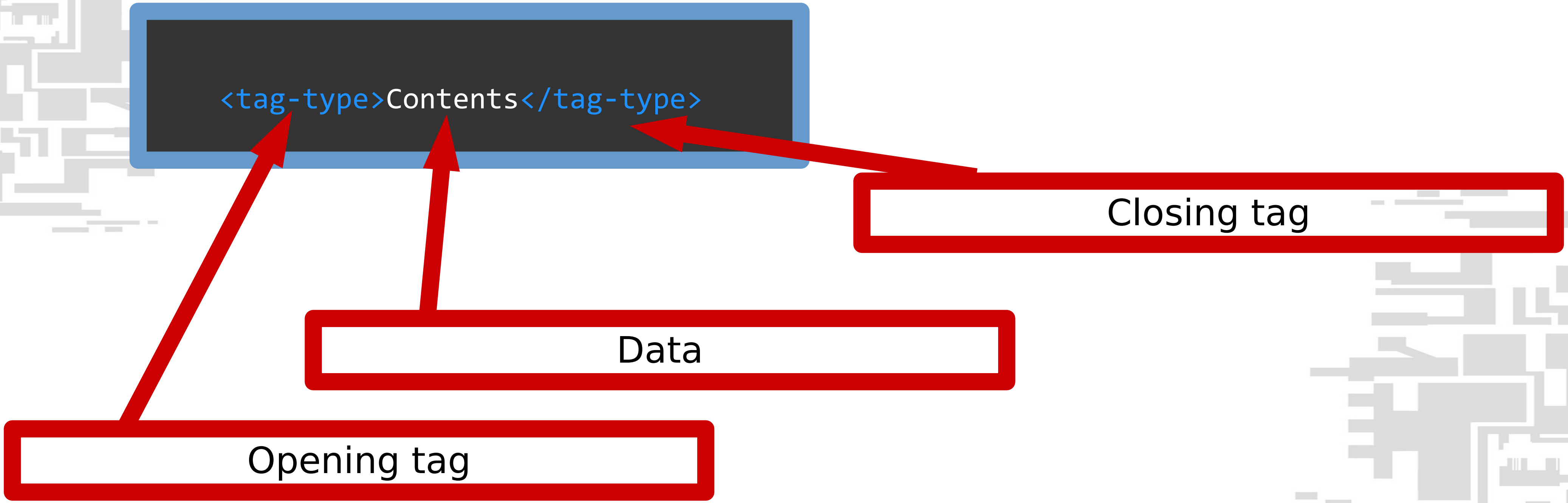
- It is recommended that you create a *project* in your editor
- The project is a folder which contains all the files that make up your website
- Create a folder somewhere you will be able to find it again e.g. Documents or on your desktop
- Open the folder in your editor (File → Open Directory or similar)

Creating a file

- Use your editor to create a new file
- Make sure your file has a .html extension
- You can now write HTML code in your file

HTML

- HTML looks like this:



```
<tag-type>Contents</tag-type>
```

HTML basics

- The web browser uses the tags to determine how to handle the containing data
- Only the data between the tags is displayed to the person who views the page in their browser

```
<tag-type>Contents</tag-type>
```

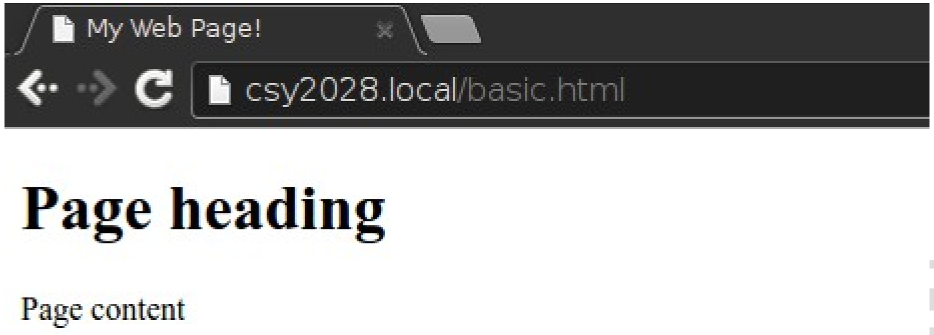
HTML tags

- Different tags have different meaning to the browser
- Each tag describes the contents inside it
- Example tags:
- <p> For a paragraph
- <h1> for a top level heading
- <li> List item
- You can have as many tags on a page as you like

HTML

- A HTML file can look like this:

```
<h1>Page heading</h1>
<p>Page content</p>
```



Page heading

Page content

HTML

- HTML is a *nested* structure.
- You can put one tag inside another
- For example, a paragraph inside a list item:

```
<li>
  <p>Page content</p>
</li>
```

HTML

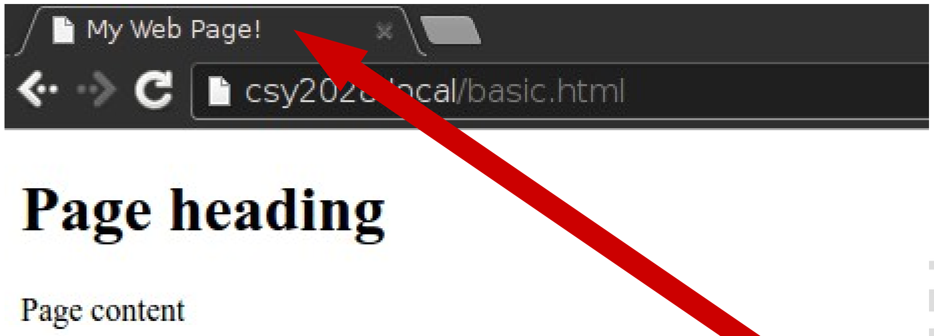
- Most tags can be *nested* inside other tags
- Every page should be wrapped in a <html> tag
- A <html> tag should contain two distinct sections:
  - <head> - Metadata used to tell the browser about the page
  - <body> - The contents of the page (things the user sees)
- Every HTML file should start with a DOCTYPE

HTML

- A basic valid HTML file looks like this:

```
<!DOCTYPE html>
<html lang="en">
  <head>
    <title>My Web Page!</title>
   <meta charset="UTF-8" />
  </head>

  <body>
    <h1>Page heading</h1>
    <p>Page content</p>
  </body>
</html>
```



Page heading

Page content

Nothing in <head>
appears on the page, but the info is
used elsewhere

Attributes

- Some tags have optional data associated with them
- This data is called an *attribute*
- To create a link to another website, you use an <a> tag (A standing for *anchor* a term going back to the start of the web)
- Sample link:

```
    <a href="https://google.com">Click here to visit google</a>
```

- The browser will display a link:
- Click here to visit google
- The page that is navigated to when you click the link is defined in the *href* attribute
- Notice that the URL is never displayed to the user on the page

Attributes

- Different tags have different attributes. Most other tags don’t have a *href* attribute, for example.
- To include an image in your page use the <img> tag
- The <img> tag has an *src* attribute and an *alt* attrbite

```
    <img src="logo.png" alt="Logo" />
```

- The src attribute (standing for source) is the name of the file name of the image you wish to display on the page
- The alt attribute (standing for alternative text) is displayed on the page if the image cannot be loaded by the browser, or by tools such as screen readers

Do you notice anything strange about the <img> tag?

- Take a close look at the <img> tag we just looked at and compare it to the others that we have looked at so far:
-

```
    <img src="logo.png" alt="Logo" />
```

```
    <h1>Page heading</h1>
    <p>Page content</p>
```

```
    <a href="https://google.com">Click here to visit google</a>
```

Self-closing tags

- Self-closing tags are tags which contain no *content*. An image is its own content, as such there is nothing meaningful to be palaced inside an <img>content</img>
- Most tags have a start and end tag e.g.:

```
    <h1>Page heading</h1>
    <p>Page content</p>
```

- Some tags, such as images, do not and are described as *self-closing*
- These do not have a closing tag (and end tag such as </h1>)

Self-closing tags

- There are two ways to describe a self-closing tag
- 1. With a slash:

```
    <img src="logo.png" alt="Logo" />
```

- 2. Without a slash:

```
    <img src="logo.png" alt="Logo">
```

- The browser will treat these identically
- However some developers prefer adding a slash as:
  - It’s clearer to anyone looking at the code that there is no closing tag
  - If the page needs to be loaded as XML, the code is valid

Nested tags

- Some HTML elements require more than one set of tags to work
- One such is *lists*
- A list is defined using the <ul> tag
- UL stands for “Unordered list”

```
<ul>
     ...
</ul>
```

Lists

- A list requires the encasing <ul> tag but each list item is defined using *li* elements (LI stands for list item)

```
<ul>
     <li>Red</li>
     <li>Green</li>
     <li>Blue</li>
     <li>Yellow</li>
</ul>
```

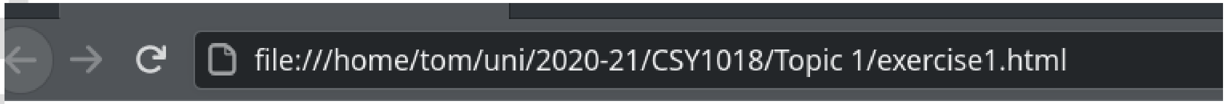
- Red
- Green
- Blue
- Yellow

Referencing other files

- A web site is usually made up of more than one *page*
- You can *link* from one page to another on the same website
- There are two ways of linking to other pages:
  - 1. Absolute URLs
  - 2. Relative URLs

Absolute URLs

- An absolute URL is the complete URL to the file. You see this when opening up a page in your web browser



file:///home/tom/uni/2020-21/CSY1018/Topic 1/exercise1.html

- This is an *absolute* URL.
- You could link to this page, using the absolute URL:

```html
<a href="file:///home/tom/uni/2020-21/CSY1018/Topic%201/exercise1.html">
     Click here to visit exercise1.html
</a>
```

Absolute URLs

- Absolute URLs, when linking to your own files are a **bad idea**:
  - 1. If you move the web pages to someone else’s computer they probably won’t put the files in the exact same location on their computer
  - 2. If you move all the files around on your computer, the links won’t work
  - **You should never use absolute URLs for files on the same website**

Relative URLs

- Relative URLs are names as such as they are *relative* to the HTML file which the URL appears in

```
    <a href="exercise1.html">Click here to visit exercise1.html</a>
```

- Clicking the link will look for exercise1.html in the same folder as the html file which contains the link
- You can move all the files from one computer to another, or to a different location on your PC and the website will still work
- **The same is true of images, and all other files used on the website**

Relative vs Absolute URLs

- You should use absolute URLs when:
  - Linking to pages on someone else’s website
- You should use relative URLs when:
  - Linking to any file on your own website
- **Remember: If your code contains a file:// url, the website won’t work when moved to someone else’s computer (For example, the examiner of your assignment!)**

HTML files

- A HTML file should:
  - 1. Contain a doctype (more on this shortly)
  - 2. Be wrapped in an <html>..</html> tag
  - 3. Contain a <head> tag for any *metadata*
  - 4. Contain a <body> tag for all the contents of the page

HTML

- HTML is forgiving
  - You can make mistakes and the browser will try to fix it
- But you shouldn't rely on this behaviour!
  - Try to always write valid HTML
  - You can (and should!) check your HTML for errors using the W3C HTML Validator http://validator.w3.org/
    - Alternatively, most editors (e.g. Sublime) have plugins to do this inside the program which makes it trivial to do
  - If your page isn't displaying as you expect, it's probably got an error and the validator will help you find out how to fix it by telling you where to look (e.g. tag name or line number)

Validating your HTML

- **You can (and should!) check your HTML for errors using the W3C HTML Validator http://validator.w3.org/**
  - Alternatively, most editors (e.g. Sublime) have plugins to do this inside the program which makes it trivial to do
- If your page isn't displaying as you expect, it's probably got an error and the validator will help you find out how to fix it by telling you where to look (e.g. tag name or line number)

Doctypes

- The Doctype is a special instruction at the top of the HTML page which starts with:

```
<!DOCTYPE
```

- There are lots of doctypes you can choose from:

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN"
         "http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd">
```

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01//EN"
   "http://www.w3.org/TR/html4/strict.dtd">
```

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN"
   "http://www.w3.org/TR/html4/loose.dtd">
```

Doctypes

- However, the only one you should use is:

```
<!DOCTYPE html>
```

- This is simpler, understood by all browsers and the standard for HTML5
- If you come across any example code that uses a different doctype (mentioning XHTML or HTML4), the article you are looking at is probably over ten years old, find a better more recent example!

HTML has changed

- Web browsers change all the time, new features are supported, things can be achieved in different ways
- Old code will work but will usually be more complicated!
- When looking for information on HTML, CSS and Javascript, make sure you find information that is up to date
- **Look at article dates, anything older than 4-5 years should be avoided, find something newer**

Finding Help

- Do not use W3Schools!
- W3Schools is **not affiliated with the w3c although they like to pretend they are**
- W3Schools often has outdated or plain wrong information

Finding help

- Better resources include:
  - https://developer.mozilla.org/en-US/ (Run by the people who make the Firefox Web Browser)
  - http://eloquentjavascript.net/ (For Javascript – Very in depth explanations of the basic concepts)
  - https://www.codecademy.com/ (Has interactive demos and tests)

Exercises

Choose an editor and create a project then create an HTML file with the following details:

- *Heading*: CSY1018 – Web Development
- *Paragraph*:
- This module focuses on creating client and server side software as well as web applications for the World Wide Web us. It concentrates on the technologies used to allow such software to be designed implemented and deployed.
- *List*:
    - Field: Computing
    - Level: BSc
    - Lecturer: Tom Butler
    - Day: Monday
    - Time: 09:00

- **Remember to set a relevant <title>**
- **Remember to save your files with a .html extension!**

- 2. Open your page in a browser and check it looks ok
- 3. Validate the page using validator.w3.org if it’s not valid, tweak the code until it is

  Remember to validate the page after you have fixed the issues it highlights!

- 3. Create a second page for one of your other modules (Details available here: https://www.northampton.ac.uk/awards/modules/COMPUTER-SYSTEMS/4/

- 4. Add a third module and link it from your other pages
  - Remember to validate all your pages!
- 5. Create a list of links at the top of the page to act as navigation.

  - Put the links in an *unordered list* (<ul>)

- 6. Test out different HTML tags
  - A good resource is https://developer.mozilla.org/en-US/docs/Web/HTML
  - Click “HTML Elements”
  - Play around with some of the tags to see what they do
- 7. Use your new tags to change the look of your web pages.
- 8. Try adding different pages on your own topics e.g. about you, your favourite team, film, etc