



WELCOME!

Introduction and Housekeeping

WEB Development Fundamentals – HTML and CSS

Overview

Objectives

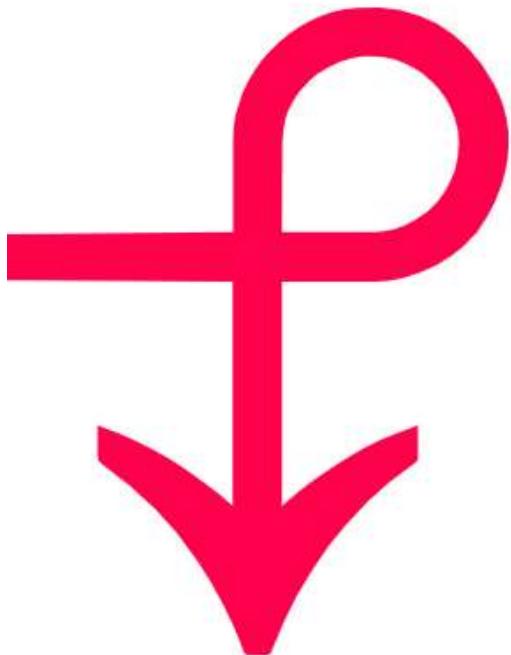
- To explain the aims and objectives of the course

Contents

- Course administration
- Course objectives and assumptions
- Introductions
- Any questions?

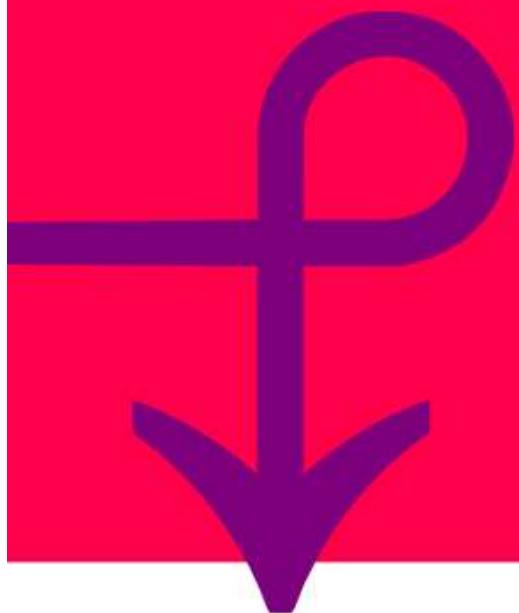
Exercises

- Locate the exercises
- Locate the help files



QA

COURSE DELIVERY

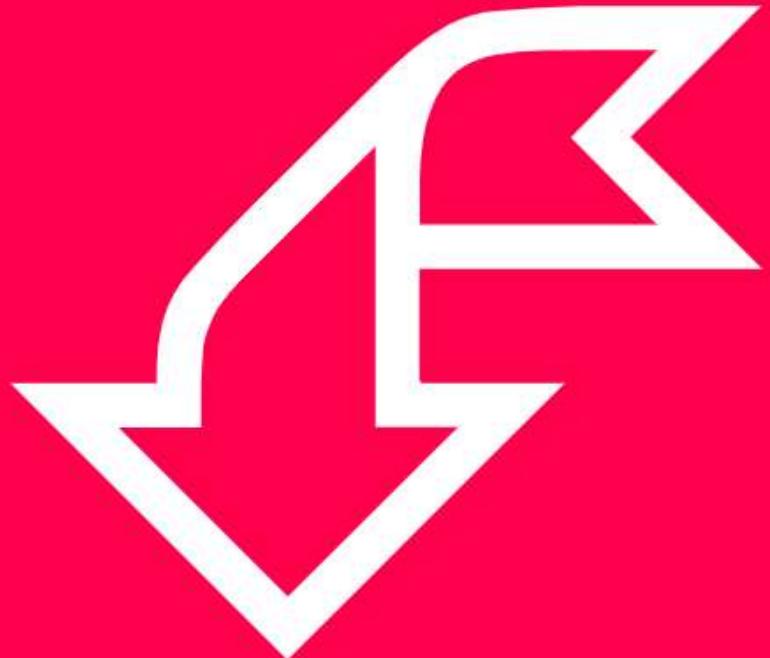


Lecture material



Hear and Forget
See and Remember
Do and Understand





The training experience

A course should be

- A two-way process
- A group process
- An individual experience

LEARNING OUTCOMES

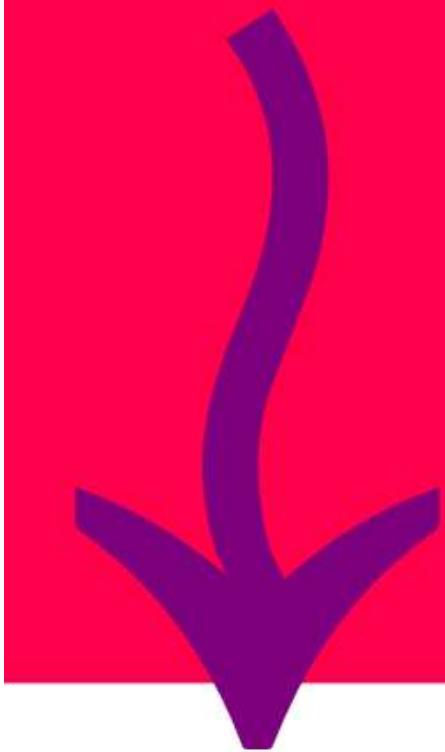
By the end of this course, you will be able to

- Describe how web pages are delivered over the Internet

- Build structured HTML pages with text, links, images, tables, and forms

- Use style sheets (CSS) for colours, background, formatting text, page layout, and simple transition, transformation, and animation effects

- Use Responsive Web Design techniques to make pages display well on all devices they may be viewed on

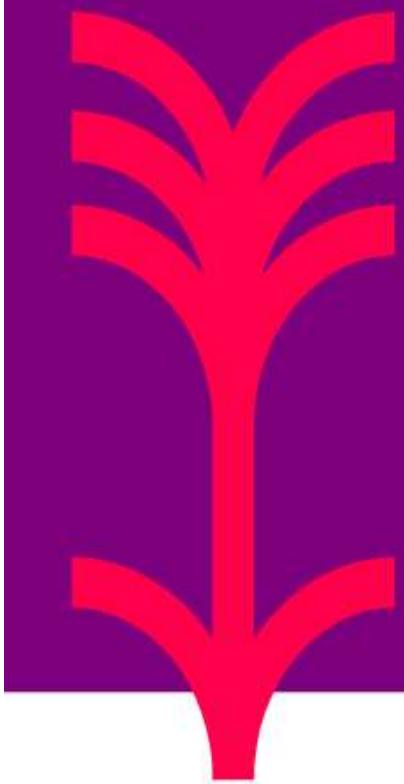


ASSUMPTIONS

This course assumes the following prerequisites

- Some technical experience
- A working knowledge of the Windows operating environment
- Familiarity with the concepts and use of the Internet

If there are any issues, please tell your instructor now





Introductions

Please say a few words about yourself

- What is your name and job?
- What is your current experience of
 - Computing?
 - Programming?
 - Web development?
- What is your main objective for attending the course?

QA

Any questions?

Golden Rule

"There is no such thing as a stupid question"

First amendment to the Golden Rule

"... even when asked by an instructor"

Corollary to the Golden Rule

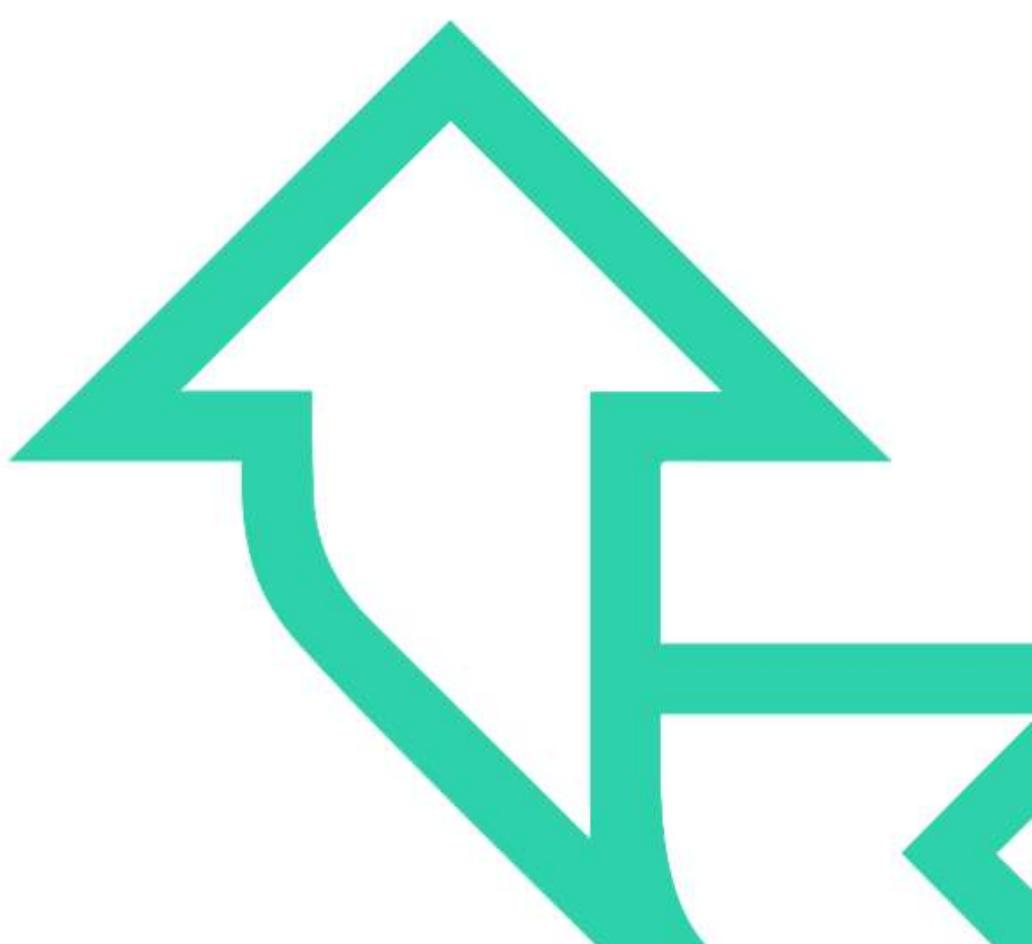
"A question never resides in a single mind"





Web Fundamentals

HTML

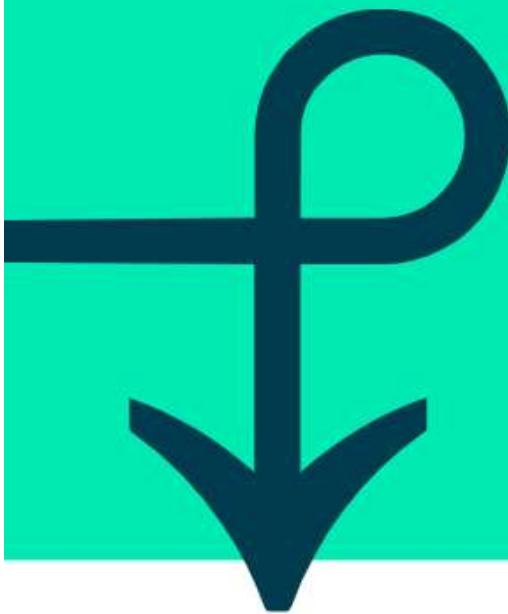


Learning Objectives

How The Web Works

Basic HTML

- HTML History and Syntax
- Structural HTML
- Hyperlinks
- Lists
- Tables
- Forms
- The <head> tag
- The DOM

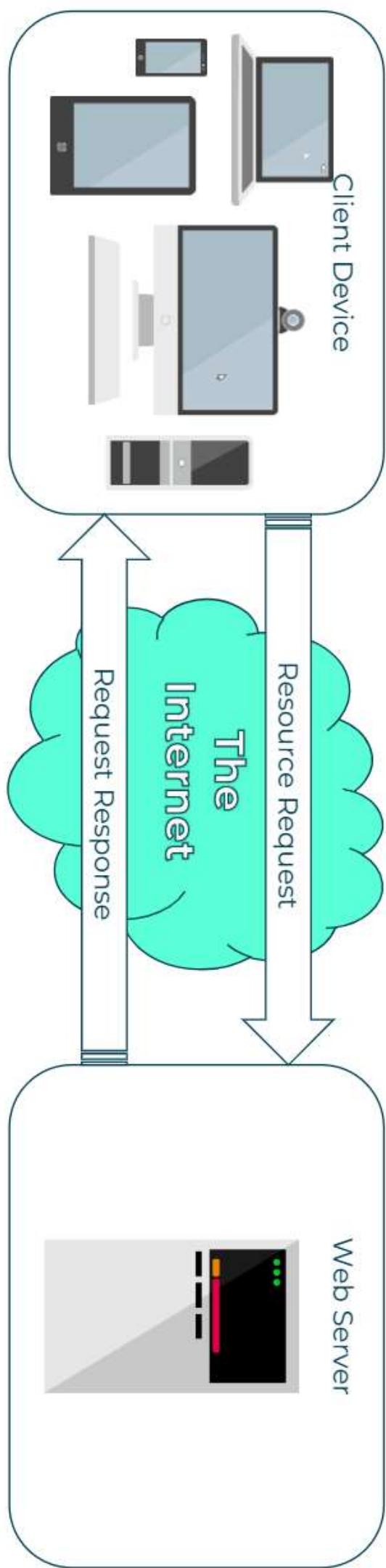




How the Web Works

- Clients and Servers
- URLs
- HTTP, HTTPS and SSL

Q A How the web works...



Client Devices

Need some form of browser to make requests

Most commonly:

- Chrome, Safari, Edge, etc
- Can also be

- Smart Devices (Televisions, home appliances, etc)



Makes request using a Uniform Resource Locator (URL) to specify where request is made to
Uses HyperText Transfer Protocol (HTTP) to actually make the request

Web Servers

Needs to be running some Web Server software

Most commonly:

- Apache
- Nginx

• Microsoft Internet Information Server (IIS)

Handles HTTP requests

Dispatches response to requests

Are addressed by URLs converted to IP addresses
by Domain Name Servers (DNS)

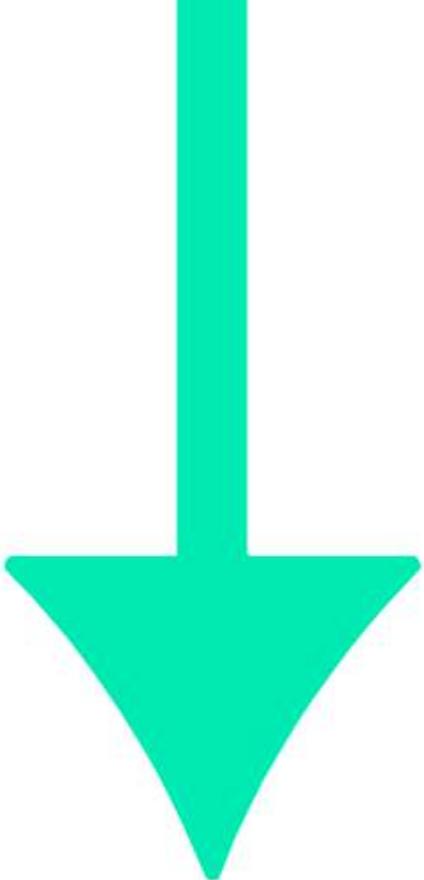
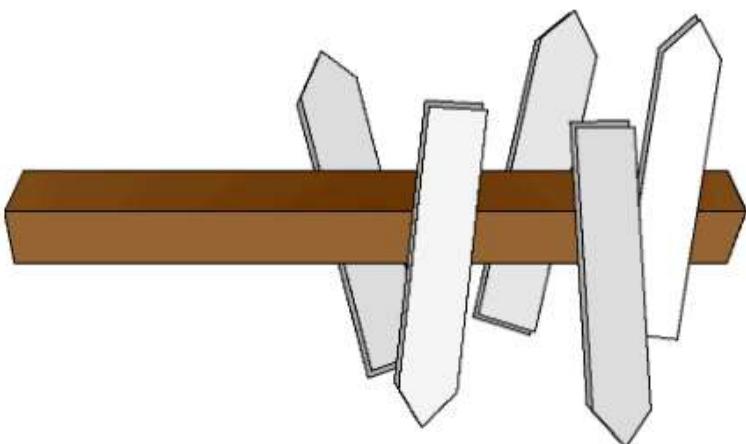


Q4

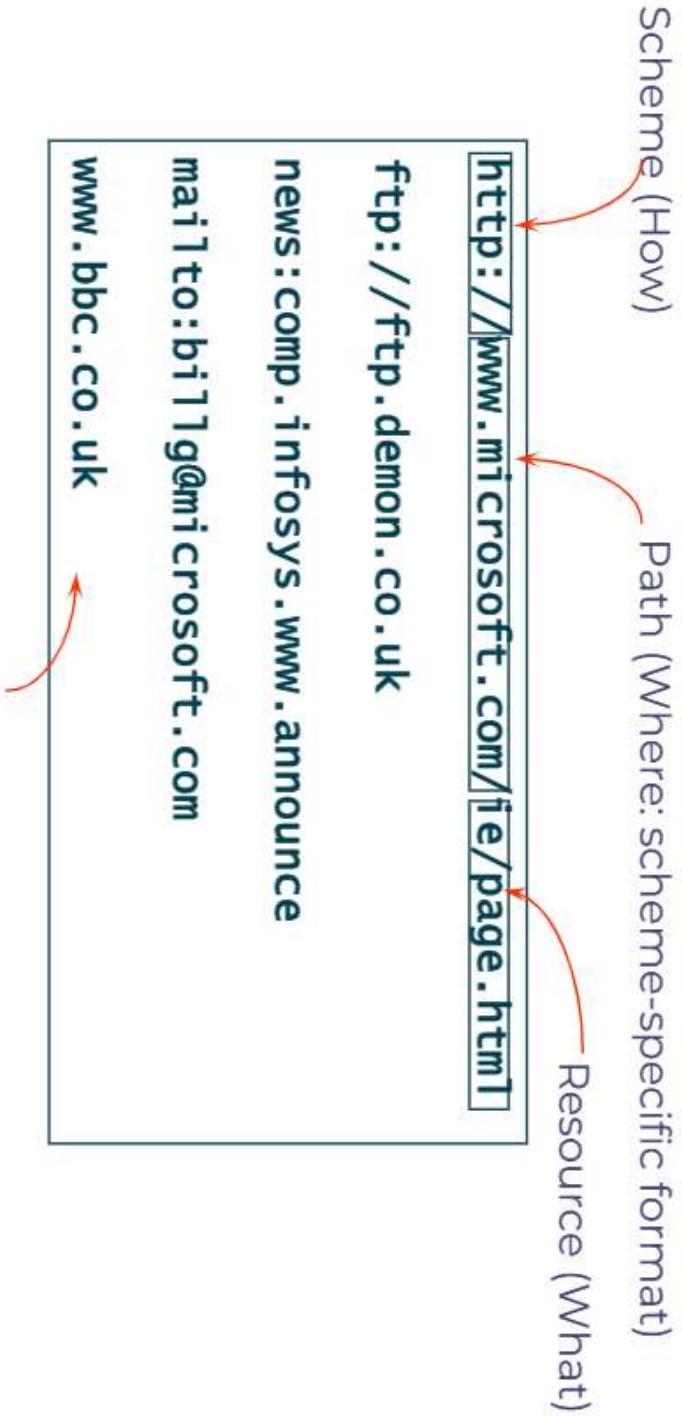
Introduction to URLs

Uniform Resource Locators (URL)

- Identifies location and protocol to access a resource
- URLs are a form of Uniform Resource Identifier (URI)



QA URL Syntax



This is *not* a valid URL, but many browsers accept it as equivalent to `http://www.bbc.co.uk`

QA HTTP URL Format

Host name
Port number. Defaults to 80.

```
http://www.microsoft.com/ie  
http://www.microsoft.com:80/ie  
http://www.carousel.org/gallop/april.html  
http://www.qa.com
```

Document path. May be case-sensitive

An empty path refers to the home page
(server-dependent; may be `index.html` or `default.html`)

Q4 HyperText Transfer Protocol (HTTP)

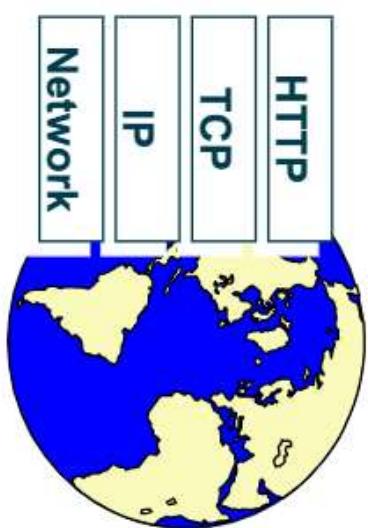
Application-Level Protocol

- Technical information at <http://www.w3.org>
- TCP-based
- Current version is 1.1

Lightweight

- Easy to implement clients and servers
- Stateless: each request is independent from the others
- Other technologies required in order to enable e-commerce, online banking, etc.

Request/response paradigm



HTTP Interactions

Client Request:

- Method, Resource, HTTP version
- MIME type header and message

Server Response:

- HTTP version and standard response code
- MIME type header and message

```
GET /index.html HTTP/1.0
```

Web server

```
HTTP/1.1 200 OK
Server: Apache
Content-Length: 793
Content-type: text/html
<html>...</html>
```





HTTP Client Request

Method

- Action to perform on resource - GET, HEAD, POST

Uniform Resource Identifier

- Identifies a networked resource
- Absolute URL used with a proxy server
- Request URL used with an origin server

HTTP Version

- Major.minor version - Default (no version given) is 0.9
- Version 1.1 now the most popular
- Browsers and Servers must also understand both 0.9 and 1.0

MIME-like message - Contains request modifiers and forms data

HTTP Server

Response

Simple Response/Full Response

Status line

- HTTP version
- Standard status code
- Reason phrase

MIME like message

- Generated by Web server or by backend script
- Header fields describe the requested resource
- Modified using HTML <meta> tag
- Requested data
- Header and Data are separated by CRLF pair



Q4

MIME And

HTTP

Multipurpose Internet Mail Extensions

- Based on Internet Mail (RFC 822)
- MIME is defined in RFC 1521
- HTTP usage differs from RFC 1521



Transmission of Multimedia Objects over Internet

- Header consists of colon-separated fields
- Data contains requested object
- Content-Type field describes object

Object Types

- Defined by IANA (Internet Assigned Numbers Authority)
- Consist of type/subtype
- Unofficial types preceded by x- (x-world/x-vrml)

Multipart Messages

- Multiple MIME messages each containing a header specifying the type of body data.

Security Issues

Preventing Eavesdropping:

- Use of encryption

Preventing Modification/Fabrication:

- Authenticating Messages

Preventing Impersonation:

- Authenticating clients and servers

Security - HTTPS

Essentially works in same way as HTTP

- Uses Secure Socket Layer (SSL) to encrypt data being passed

Lots of websites and development libraries and frameworks will work with or require HTTPS

Stops 'eavesdropping' on data transfers between client and server

Security - SSL

- Secure protocol for sending information securely over the Internet
- Encrypts data transmitted between client and server
- Mostly uses 128-bit encryption
- Requires web server to hold a valid certificate



HTML Basics

- HTML History and Standards
- Basic HTML syntax, tags and errors
- Special Characters
- Hyperlinks
- Images
- Lists
- Tables
- Forms
- The Head Element
- The DOM



HTML

QA

HTML Standards

HTML 2.0

- Application of Standard Generalized Markup Language (SGML)
- Attempted to update the standard

HTML 3.0 (Obsolete)

- Netscape and Microsoft ad-hoc extensions

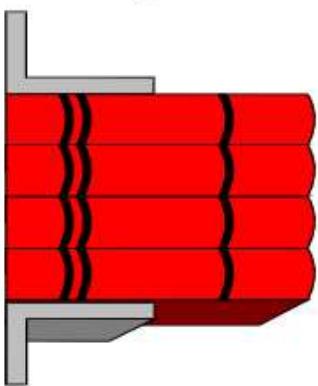
HTML 3.2

HTML 4.01

- Includes Cascading Style Sheets(CSS), DOM, scripting, XML and many other features

HTML 5

- The current W3C standard
- 5.2 now a recommendation available at <http://www.w3.org/TR/html5/>



QA A Simple HTML Page

Three sections to an HTML document

- Prolog(document type declaration), head and body

HTML Tags appear in angle brackets (< and >)

- Many tags appear in pairs, acting as containers
- Names are not case sensitive (convention is lower-case)
- Carriage returns are not significant

```
<!DOCTYPE html>
<html>
<head>
    <title>Success!</title>
</head>
<body>
    Congratulations on completing
    your QA Training course.
</body>
</html>
```

QA HTML versions & DTDs

The different versions of HTML have their own DTD

- The appropriate DTD should be included in the HTML prologue

HTML 4 Strict

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.0//EN" "http://www.w3.org/TR/html4/strict.dtd">
```

HTML 4 Transitional

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.0 Transitional//EN"  
"http://www.w3.org/TR/REC-html4/loose.dtd">
```

HTML 5

```
<!DOCTYPE html>
```

Q1 Attributes of Tags

Many tags have additional attributes

- Some are required

```

```

- Some are optional

```
<h1 title=".....">.....</h1>
```

Attribute values should be enclosed in quotes

- Not necessary if one word attribute value
- Good practice

QA Comments

- Use `<!--` and `-->` to enclose comments
- Not displayed by browser
- Can be used to hide scripts/styles from older browsers

```
<html>
<head><title>Success!</title></head>
<body>
<!--A demo that comments can be used
to clarify HTML -->
Congratulations on completing
your <strong>QA Training</strong>
course.<br>
</body>
</html>
```

Q&A Container Tags

Containers may be nested but may not overlap

```
<!-- This is correct -->
<p>This text is a paragraph containing <b>bold</b> and
<i>italic</i> text. Some words are both <b><i>bold and
italic</i></b>. </p>
<!-- This is incorrect -->
<p>The <b>quick brown <i>fox</i></b> jumps over the</i> lazy
dog.</p>
```



QA Special Characters

- How can we express characters like "<" and ">"?
- Use character entities

Character	HTML Name
<	<
>	>
&	&
"	"
(non-breaking space)	

If the relative density > 1.0 then
"you have a problem!"

If the relative density > 1.0 then "you have
a problem!"

QA Syntax Errors and Non-Standard HTML

Browsers never say “syntax error”

- May respond in different ways to incorrect HTML

Browsers ignore unrecognised tags and attributes

Allows updating of HTML standards

Non Standard HTML:

- Blink

```
<blink>Blinking text</blink> can be annoying,
```

- Marquee (Still works in many browsers)

```
<marquee>This message will scroll</marquee>.
```

- Not recommended as they could be removed at any time



Structural HTML

Basic HTML



Q& HTML5 Structural Elements

HTML5 has a series of structural elements

- To create a more semantically structured page

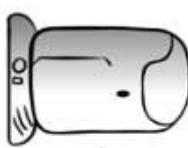
The main building blocks of HTML5 are:

- <header>
- <nav>
- <section>
- <main>
- <article>
- <aside>
- <footer>

QA Headings and sectioning elements

HTML5 documents have an **outlining algorithm**

- Generates a table of content based on the section and heading you've used
- Important for assistive technology and search engine optimisation



1 MAIN HEADING
• 1 LEVEL 2 HEADING
• • 1 LEVEL 3 HEADING
• 2 ANOTHER LEVEL 2 HEADING

TOO EASY. GIVE ME
A HARDER ONE!

Sectioning elements create a logical structure in the document outline

```
<body>
<h1>Main heading</h1>
<p>Some text</p>
<h2>Level 2 heading</h2>
<p>Some more text</p>
<h3>Level 3 heading</h3>
<p>A bit more text</p>
<h2>Another level 2 heading</h2>
<p>The last bit of text</p>
</body>
```

The <header> element

- Should act as container logos, links back to home etc.
- Will usually contain a <h1> to <h6> to denote level of header

There can be one <header> per sectioning block

```
<header>
  <a href="/">
    
  </a>
  <h1>My Main Title</h1>
  <h2>My Sub title</h2>
</header>
```



QA

The <nav> element

<nav> is used to mark up navigation

- Should be limited to links within the page and site
- Not sponsored links for instance

Links normally surrounded by within a

Multiple <nav> allowed, each should contain a related category

```
<nav>
<h2>Main site navigation</h2>
<ul>
<li>
<a href="/">Home</a>
</li>
<li>
<a href="/aboutus.html">
About Us
</a>
</li>
</ul>
</nav>
```



QA

The <footer> element

Contains information about the section or document,
e.g. the author

The <footer> element requires no heading element

- Unique in the sectioning element
- It is optional and can be added

The above code is a ‘fat footer’

- The <small> tag represents small print in HTML5
- Add a <nav> element within if links required

```
<footer>
  <small>
    This tag has been redefined
  </small>
</footer>
```

Many <footer> elements may occur on a page

Appear at the end of a sectioning element

- blockquote, body, div etc.





QA

The <article> element

Represents a self-contained composition on the page

- A blog entry
- Comic strip
- Video

Articles represent indivisible units of work

```
<article>
  <h2>Yesterday</h2>
  <p>Some stuff goes here</p>
</article>
<article>
  <h2>Today</h2>
  <p>Some more stuff goes here</p>
</article>
```



Q/A

The <aside> element

- It should assist but not be essential to the main document
- For example nested within an <article>
- Or used at a page level to denote 'sidebar' content

```
<article>
  <h1>My Blog Post</h1>
  <p> . . . </p>
  <aside>
    <h1>Glossary</h1>
    <dl> . . . </dl>
  </aside>
</article>

<aside>
  <h2>Blog roll</h2>
  <ul>
    <li><a href="#">Link 1</a></li>
    <li><a href="#">Link 2 Friend</a></li>
  </ul>
</aside>
```

The <section> element

<section> is used to break up semantic elements

- Different parts to a news story or a group of links
- Otherwise the <section> is untitled in the HTML outline
- Most generic and easiest to abuse semantic element
- Do not use as a pure stylistic container

```
<body>
...
<section>
  <h2>
    heading level = section nesting level
  </h2>
  rest of the content
</section>
...
</body>
```



QA

The <section> vs. <article> element

ON OUR SAMPLE WEBSITES, WHAT
WOULD BE MARKED UP AS A SECTION?



OK SO EACH OF THOSE IS A SECTION.
IS THERE AN EXAMPLE THAT SHOWS THE
DIFFERENCE BETWEEN A SECTION
AND AN ARTICLE?

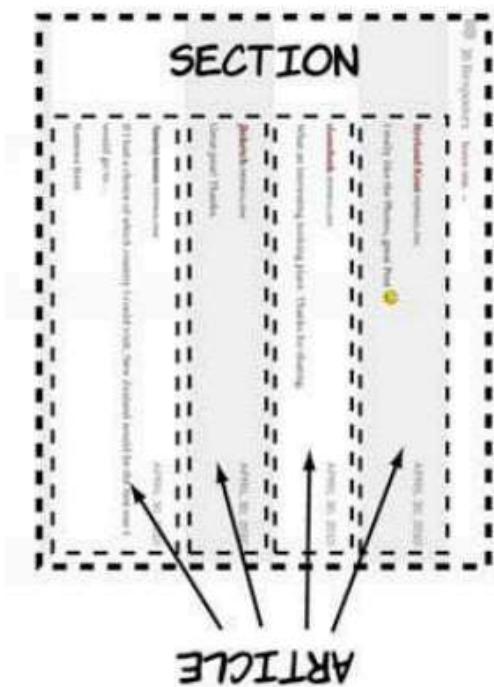
LOOK AT THE BBC HOME PAGE. EACH
COMPONENT IS FROM A DIFFERENT PART
OF THE SITE: NEWS, SPORTS, AND SO ON.



LOOK AT THE COMMENTS SECTION ON A
TYPICAL WORDPRESS BLOG: IT'S A SECTION
WITH EACH COMMENT AN ARTICLE.



A screenshot of the BBC homepage. It features a large news banner at the top with a photo of a person. Below it, there are two main sections: "News" and "Sport". Each section has its own navigation menu and content area. Arrows point from the word "SECTION" to both the "News" and "Sport" headers.



A screenshot of a WordPress blog's comments section. It shows a list of comments with avatars, names, and timestamps. Arrows point from the word "ARTICLE" to each individual comment, indicating that each comment is treated as a separate article.

<Div> Is still Valid HTML5

- I.e., A generic element for structuring a page
 - Has no semantic meaning except via attributes
- Used if no semantic alternative or as a CSS wrapper**

```
<body>
  <div id="wrapper">
    <header>...</header>
    <nav>...</nav>
    ...
  </div>
</body>
```



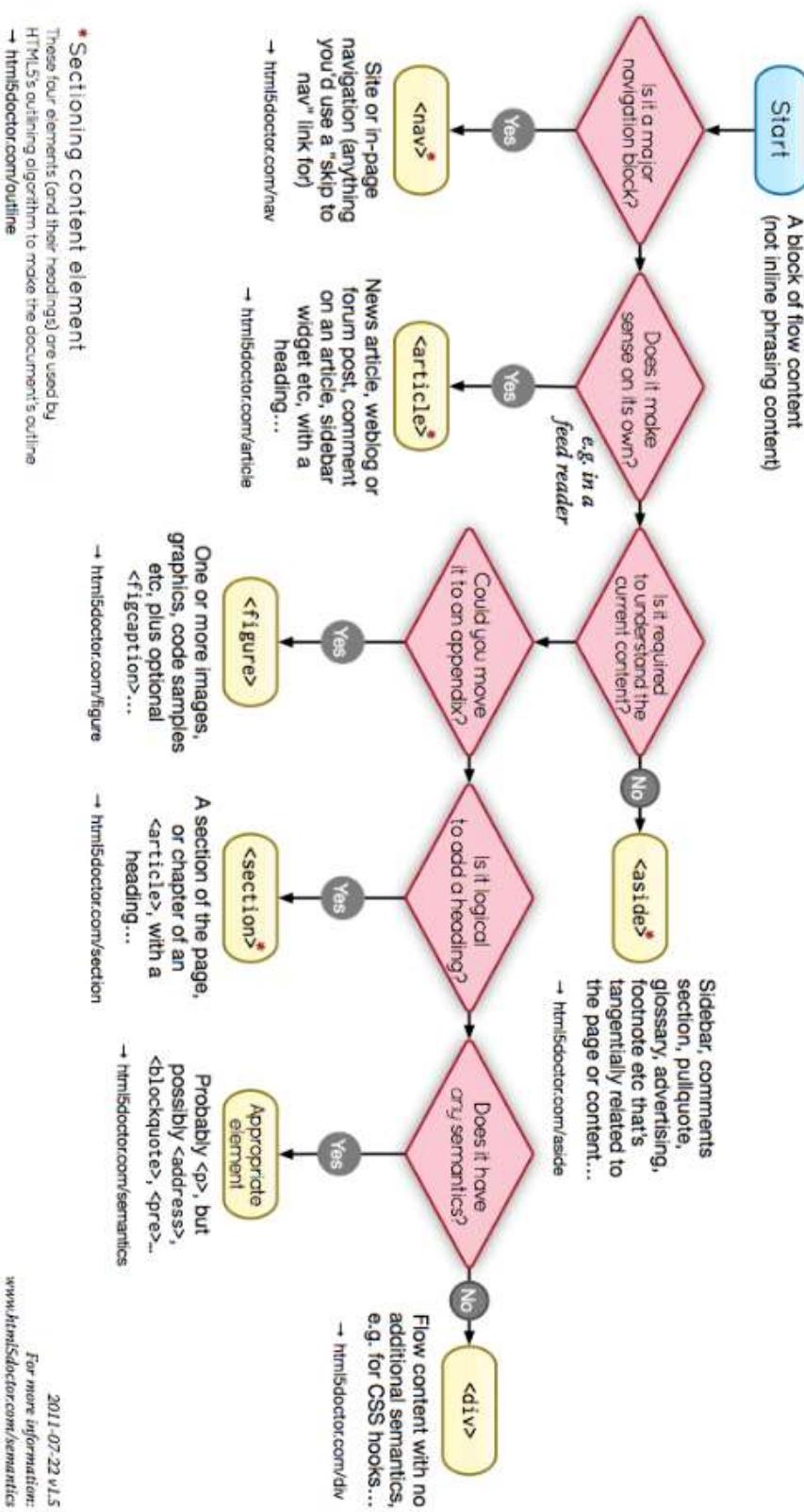


When to use what



HTML5 Element Flowchart
Sectioning content elements and friends

By @niddle & @boblet
www.html5doctor.com



Q&A When to use what

Element	Typical Content	Typical Parent Element	Typical Child Element
<code><header></code>	Title, logo, banner, Introductory information	Body, Section, Article	Nav, Section
<code><nav></code>	Primary navigation content	Body	Section, Nav
<code><section></code>	Generic page section	Body	Article, Header, Footer, Aside, Nav
<code><article></code>	Story, subsection, blog post	Body, Section	Section, Header, Footer
<code><aside></code>	Sidebar content, tip, quotation	Body	Section, Article
<code><footer></code>	Footer, summary, copyright, info, secondary navigation	Body, Section, Article	Nav, Section
<code><main></code>	Unique content, central to the topic of the document	Body	Section, Article, Aside (not repetitive content like nav, header or footer)

Restructuring A Blog With HTML5

Each blog entry would be an article
Providing clear semantic intent for the page

```
<article>
  <header>
    <h2>My HTML5 Blog</h2>
    <p>March 15th 2010</p>
  </header>

  <p>Much less divitius occurs!</p>

  <footer>
    <address>
      <a href=". .">Posted in</a>
    </address>
  </footer>
</article>
```



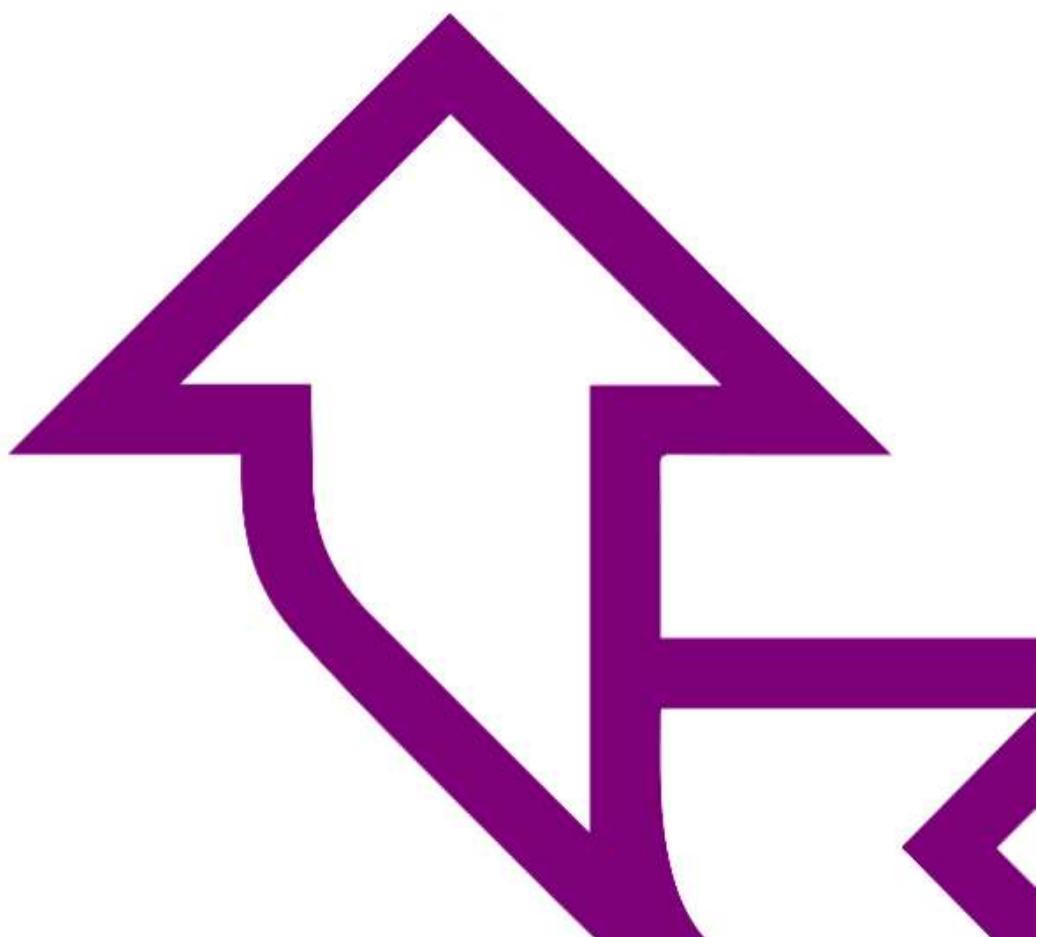
Quick Lab 1 - Structural HTML

Convert an HTML document into a semantically tagged document and test the outcome



Hyperlinking

Basic HTML



Hyperlinks

Hyperlinks connect related pieces of information

- Can point to separate documents located on different Web servers
- Hyperlink text is shown by colour change and/or underlining
- Linked document is loaded by clicking text/object with mouse

<a> ... tags define the hyperlink

- The HREF attribute takes a URL as a value

If you want to find out about our courses, you could try

QA Training's home page

or telephone us on

+44 (0) 1793 696000.

If you want to find out about our courses, you could try [QA Training's home page](http://www.qa.com/index.html) or telephone us on +44 (0) 1793 696000.

Named Anchors

Named anchors permit

- Hyperlinks within a document
- Hyperlinks from one document into any part of another

Use < a id="AnchorName" > to define

- Name attribute is used as well as ID

```
<a name="AnchorName">
```

Use

< a href="TargetURL#AnchorName" > to jump

```
<a name="Start"> ... </a>
<a href="http://www.qa.com#QAHTMLDEV">
    HTML for Programmers</a>
<a href="#Start">
    Top of page</a>
    ...
    .
    .
    .
<a name="QAHTMLDEV">
    HTML for Programmers</a>
```

QA Relative and Absolute URLs

```
<!-- Absolute pathname -->
<a href="http://www.qa.com/index.html">... </a>

<!-- Relative pathname: find 'index.html'
relative to current page on this server -->
<a href="index.html">... </a>

<base href="http://www.myserver.com/backissues">
<!-- This is a relative pathname, but is interpreted
relative to the setting in <base> above.
Therefore this relative URL always points to the
page on the server. -->
<a href="jan13.html">... </a>
```

Q4 Hyperlink Targeting

The target attribute of a hyperlink tag can have one of these predefined values:

- `_blank` A new blank window or new tab in a tabbed browser.
- `_parent` Immediate parent of the document the link is in.
- `_self` The same window the link was clicked in.
- `_top` The full body of the window.

```
<a href="circont.htm" target="_parent"> . . . </a>
<a href="circont.htm" target="_top" > . . . </a>
<a href="circont.htm" target="_blank" > . . . </a>
<a href="circont.htm" target="_self" > . . . </a>
```

QA Mailto: links

Changing the scheme from http:// to mailto: tells the browser to execute it's default e-mail behaviour.

This usually opens a new e-mail for composing

```
<a href="mailto:instructor@qa.com">Contact Instructor</a>
```

Can also append a subject

```
<a href="mailto:instructor@qa.com?subject=Hello Instructor">Contact Instructor</a>
```

And even some content for the body of the e-mail

```
<a href="mailto:chris.bruford@qa.com?subject=Hello Instructor&body=Just dropping a  
line to say hello">Contact Instructor</a>
```



Quick Lab 2 – Hyperlinks

Add some links to HTML and ensure that they work



Images

Basic HTML



A very large number of graphics file formats

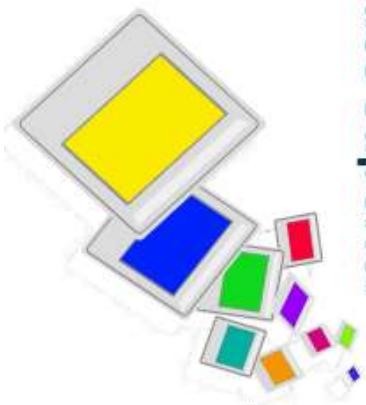
- BMP, GIF, PCX, JPEG, TIFF, CCM ...
- GIF (Graphics Interchange Format)
- JPEG (Joint Photographic Experts Group)
- Text-only browsers (e.g., Lynx) ignore all images
- Usually possible to disable images to reduce bandwidth

Image File Formats

Some browsers support other image file formats

PNG (Portable Network Graphics) format

WebP – modern image format that provides superior lossless and lossy compression



QA Inline Images: the `` Tag

```
<html><head><title>Success!</title></head>
<body>
<h1>Well done!</h1>
Congratulations on completing your
course with flying colours!<br/>

<br/>
```

We are confident that you won't find a
better training agency anywhere.

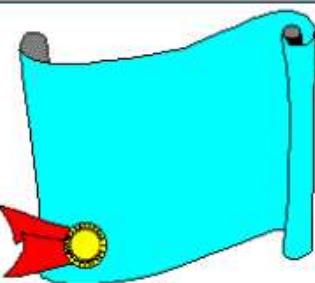
</body>

</html>

This text is shown when images
are unavailable or disabled

Well done!

Congratulations on completing your course with flying colours!



We are confident that you won't find a better training agency anywhere.

Image Height and Width

A picture's size is usually found from data in the file

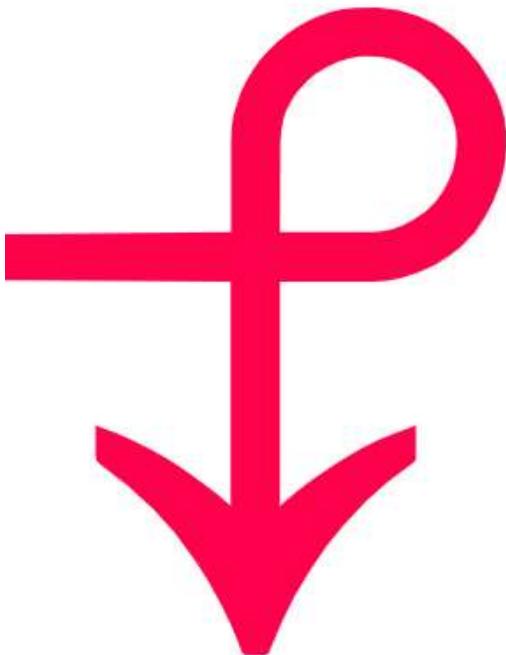
- Can change this with WIDTH and HEIGHT attributes

Do not use WIDTH and HEIGHT attributes to reduce the display size of an image

- The browser will need to download the entire image

```
<!-- Force the image to be 100x100 -->  

```



Q4 Graphical Hyperlinks

```
<html>
<head><title>Graphical links</title>
</head>
<body>
<p>
<a href="http://www.microsoft.com">Microsoft home page</a>
<br/>
Alternatively, click on this star:
<a href="http://www.microsoft.com">

</a>
<br/>
</p>
</body>
</html>
```

[Microsoft home page](http://www.microsoft.com)

Alternatively, click on this star:



QA Thumbnail Images

Images can be large

- An issue if the users' connection is slow

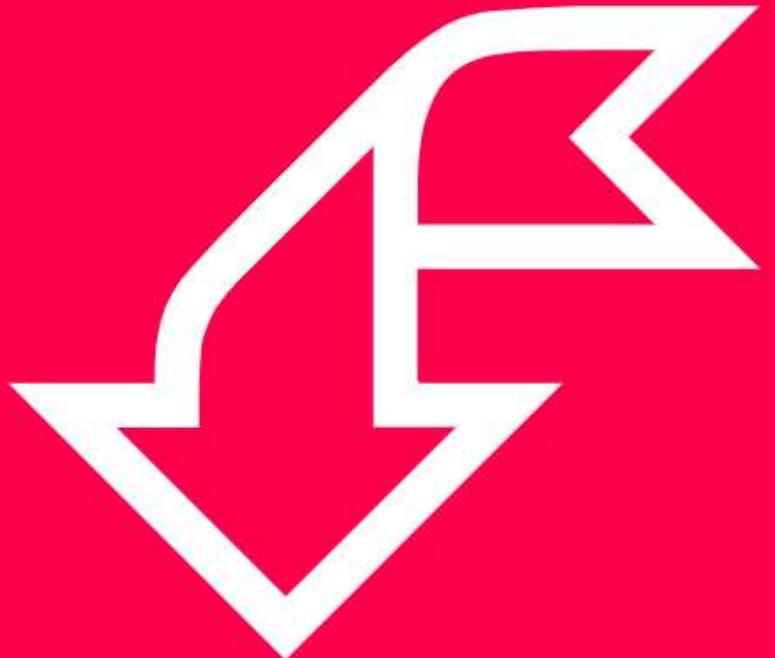
For larger images, provide a thumbnail image

- Acts as a preview
- Leads on via a hyperlink to the full image

Create thumbnail as separate image

Don't use WIDTH and HEIGHT attributes to shrink

```
<a href="pictures/monalisa.gif">  
    
</a>
```



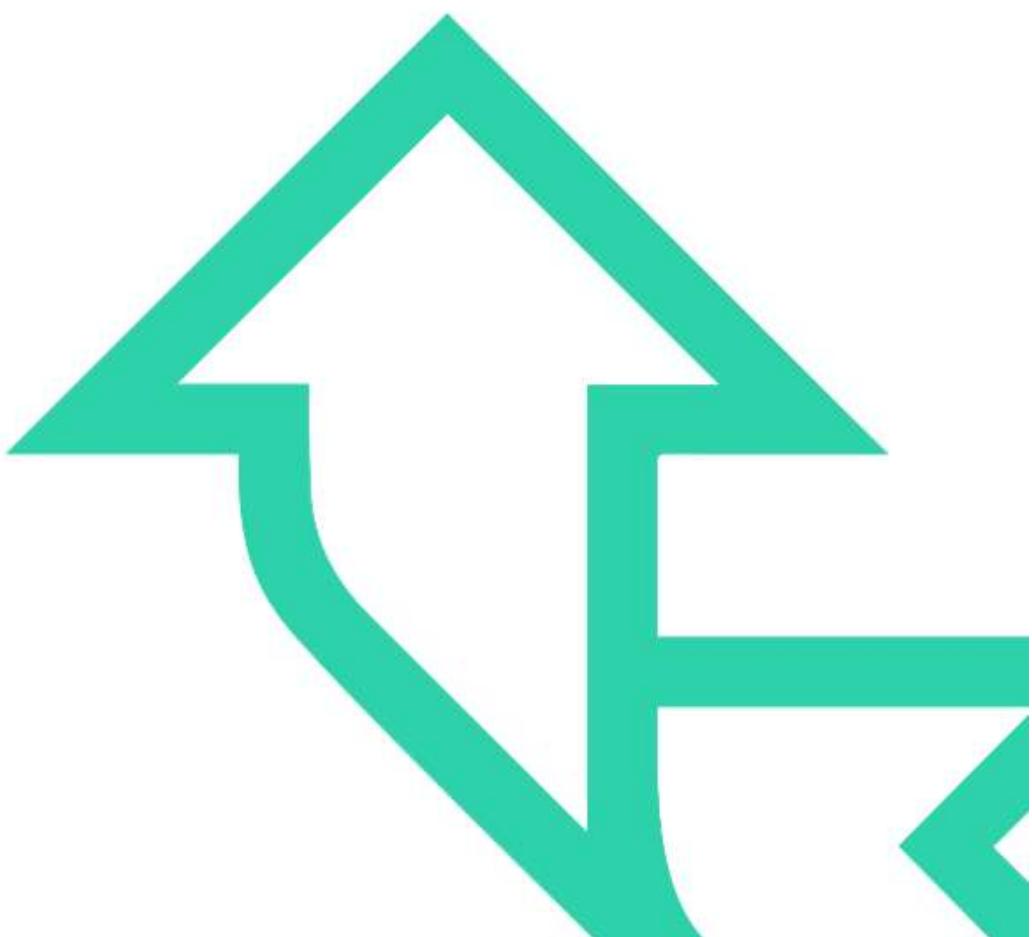
Quick Lab 3 – Images

Add some links to images to an HTML page and view them



Lists

Basic HTML



QA Lists in HTML – Ordered and Unordered

List items are inserted into lists using the tag

**Ordered lists (numbered, etc) are created by surrounding list items with **

- Start point can be defined using the start attribute

```
<ol start="5">
<li>List item 1</li>
<li>List item 2</li>
</ol>
```

5. List item 1
6. List item 2

**Unordered lists (bullet points, etc) are created by surrounding list items with **

```
<ul>
<li>List item 1</li>
</ul>
```

- List item 1

Q& Lists in HTML – Description Lists

Description lists are created by using the **<dl>** tag

<dt> tags are used to define a term or name in a description list

<dd> tags are used to describe a term or name in a description list

- Can contain paragraphs, line breaks, images, links, lists, etc

```
<dl>
<dt>Summer</dt>
<dd>Season when it is warm</dd>
<dt>Winter</dt>
<dd>Season when it is cold</dd>
</dl>
```

Summer	Season when it is warm
Winter	Season when it is cold



Tables

Basic HTML



QA Tables

→ Defined with the <table> tag and can use the following:

Tag	Description
<tr>	Table row
<td>	Table data (i.e. a normal cell)
<th>	Table heading (i.e. a header cell)
<caption>	Table caption
<thead>	Groups header content in a table
<tbody>	Groups body content in a table
<tfoot>	Groups footer content in a table
<colgroup>	Specifies one or more columns for formatting
<col>	Specifies column properties for each column within



Quick Lab 4 – Tables

Display some data in a table on an HTML page



Forms

Basic HTML



QA The <form> tag

Defines a form in HTML

Attributes

- **action:** URL to execute when form is submitted
- **method:** how the information is passed to server (GET or POST)

```
<form method="post" action="cgi-bin/subscribe.pl">  
...  
</form>
```

QA Text and push buttons

Kinds of push button

- **submit**: send the form information to the server for processing
- **reset**: reset all form fields
- **button**: generic push button

Single line text field

- **type="text"** attribute of <input> tag

- **type="password"** as text, but text is not displayed on screen – browser default hidden used

```
<form  
method="post"  
action="scripts/subscribe.pl"  
>  
Please enter your name:  
<br/>  
<input type="text" size="20" name="UsrName"/>  
<br/>  
<input type="submit" name="OKBtn" value="OK" />  
</form>
```

Please enter your name:

OK

Q4 Multi-line text input

`<textarea> .. </textarea>`

- Initial text can also be supplied
- Browser will supply scroll bars if necessary

Address: `
`
`<textarea
name="UsrAddr"
rows="7"
cols="24">`
 >
 Enter address here
`</textarea>`

Address: Enter address here

Q A Check and radio buttons

Radio buttons: select from one of a group

- name attribute groups buttons together

Check buttons: independent yes/no value

```
<form ...>
<input type="radio" checked name="RadioDrink" value="Tea" />Tea
<input type="radio" name="RadioDrink" value="Coffee" />Coffee
<input type="radio" name="RadioDrink" value="Soup" />Soup<br/>
<input type="checkbox" name="CheckMilk" value="Yes" />Milk
<input type="checkbox" name="CheckSugar" value="Yes" />Sugar<br/>
<input type="submit" name="OKButton" value="Vend" />
</form>
```

<input checked="" type="radio"/> Tea	<input type="radio"/> Coffee	<input type="radio"/> Soup
<input type="checkbox"/> Milk	<input type="checkbox"/> Sugar	
<u>Vend</u>		

QA <select> and <option>

Select one from a drop-down lists

<select> tag defines the list

- **size** attribute is 1 for drop down list, > 1 for scrolled list

- **multiple** attribute specifies multiple selection

list

```
<select name="Drop1" size="1">
<option>Cookies</option>
<option>Tortillas</option>
<option selected>
    Poppadoms (my favourite)
</option>
</select>
```

<option> tags define the contents

```
<select name="Drop2" size="3">
<option value="2">High</option>
<option value="1">Medium</option>
<option value="0">Low</option>
</select>
```

High
Medium
Low

QA

The HTML5 spec started with forms

- Opera and Safari are the driving force and most complete

HTML5 input

elements

13 type options to date

Mostly extend the <input> tags with additional type values

- If a browser does not understand the extension rendered as:

No requirement in the spec for how browsers present

- Different browsers show different UI and error messages
- Browsers that do not understand the new types treat them as text
- JavaScript defence is needed for legacy browsers

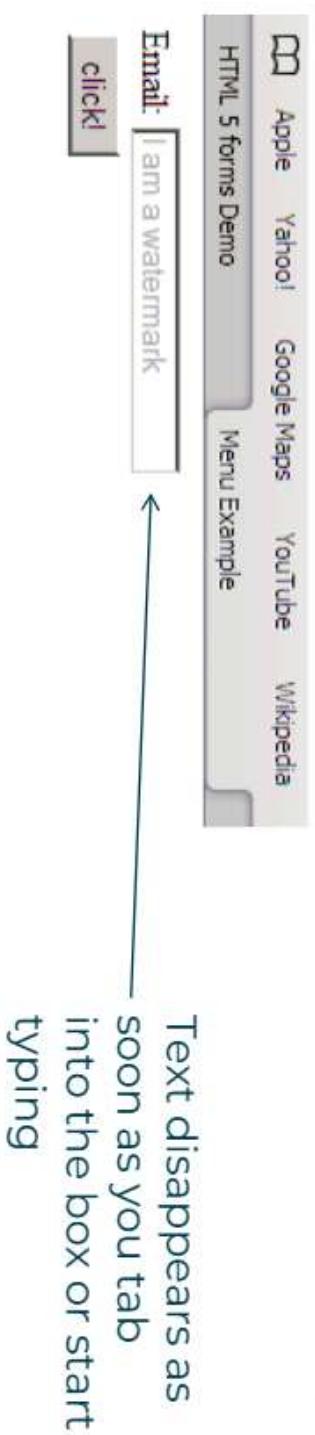
```
<input type="text" />
```

QA Placeholder attribute

The placeholder attribute offers default text

- Gives the user example or instruction for the field
- Sometimes called a watermark
- Can only be used for text values
- Is not a default value

```
<input type="text" placeholder="I am a watermark" />
```



QA Required Fields

- You can force a field to be mandatory on the client

```
<input type="text" autofocus="true" required />
```

- On a submit action an error message may appear:

- Appears in:

- Safari 6+
- Firefox 4.0
- Opera 9+
- Chrome 9+
- IE10+

- Message will appear differently in each browser



Q/A Autofocus attribute

It is common to have the first field of a form to focus

- To have the cursor flashing ready to type

Previously achieved with JavaScript

- The markup representation is faster
- Part of the page rendering rather than code execution

Supported in all browsers other than IE9 and less:

- Use JavaScript to support legacy browsers

```
<form>
<input name="q" autofocus="true">
<input type="submit" value="Search">
</form>
```

QA Email input type

Add type value of email

What happens in the client is not consistent

- HTML5 spec does not demand it
- Opera and safari provides submit validation
- Firefox provides client validation on blur
- Safari mobile changes the input keyboard
- IE 9< does nothing
- Form will not submit until the error is solved
- Provides a simple input mask to check input
 - e.g. boffin@qa.com



QA Web address input type

Denotes the input must have schema prefixing an address

- e.g.: `http://www.qa.com` or `ftp://ftp.mysite.com`

Different browsers have different functionality

- More of an issue than previously due to different functionality
 - Chrome and Firefox 4 force user to add schema
 - Opera prefixes an address with `http://`
 - Safari mobile provides a different keyboard

To use effectively combine with a placeholder

- Users don't really understand schemas



QA Number input type

Numbers often need to be constrained by range

New number type provides this functionality

Four attributes: <input type="number" min="1" max="12" step="2" value="6" />

- min - lowest range

- max - upper range

- step - what value the control enumerates by

- value - default value

Browser support issues

- Firefox, Chrome and Opera display these as 'spinboxes'

- No increment/decrement buttons in IE

- iOS UI widget doesn't consider step, min or max values

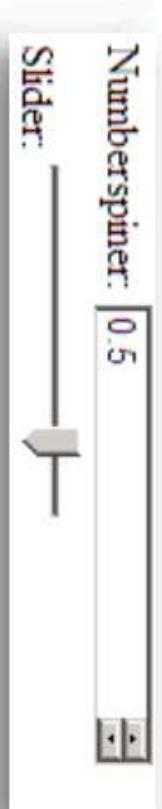


Q A Range input type

Creates a slider bar in Chrome, Firefox, IE10 & Opera

```
<input name="r" type="range" min="1" max="11" value="9">
```

Has the same attributes as the Number type



Only IE10 displays the current slider value

- In other browsers you need JavaScript to display the value
 - Obtain a pointer to the slider
 - Subscribe to its change event
 - Add the output to a span

QA Date input type

A popup calendar is standard for date selection

- Normally requires a JavaScript framework
- Around 10% of web users do not support JavaScript

HTML5 defines six date time types

- Use UTC in the same way as time element

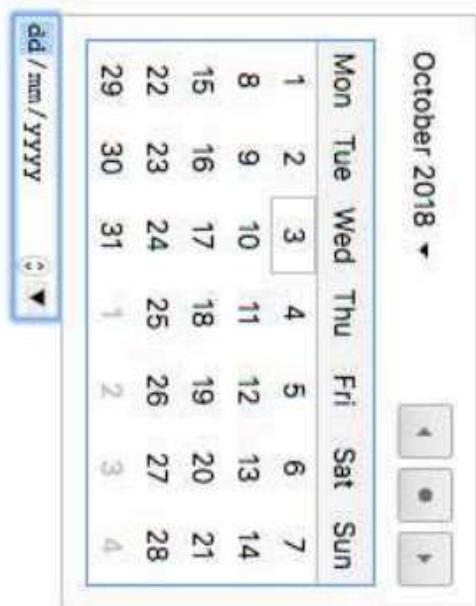
No support in IE or Safari as of Oct 2018

Will not require JavaScript enabled

- Native support is the ultimate aim

ECMAScript5 allows you to create dates from UTC

Formatting is dependent on browser defaults



Search input type

Provides a semantic definition for search input

- Keep to one per page

Need to set a name for the search field otherwise nothing will be submitted

Most common name is q

Search Google:
`<input type="search" q="googlesearch"/>`



QA color input type

Currently implemented in Chrome, Firefox and Opera

Returns a six digit hex value

- In other browsers should be verified as a hex value

On MacOS and Windows returns a colour picker so far



QA Pattern

The pattern attribute allows use of regular expressions

Pattern works with the following input types:

- text
- search
- url
- tel
- email
- password

```
<input type="text" pattern="[0-9]{13,16}" name="CreditCardNumber" />
```

Ensure the user understands the regular expression

- Support with a placeholder

QA Datalist

The <datalist> tag provides a list of pre-defined options

- Use the <input> element's list attribute to bind it to the datalist
- Provides an "autocomplete" feature on <input> elements.
- Users see a drop-down list of options as they input data.

```
<input list="browsers" />
<datalist id="browsers">
  <option value="Internet Explorer">
  <option value="Firefox">
  <option value="Google Chrome">
  <option value="Opera">
  <option value="Safari">
</datalist>
```

QA Form validation

As we have seen some browsers ship with validation

- IE offers no UI implementation in any version
- Firefox and Opera often the most complete implementation
- Chrome is pretty good and Safari will get better
- Some controls have silent errors, not enough UI feedback

These are JavaScript free client validation

Uneven support may be more trouble than benefit

- You can tell a browser to switch it off
- Still benefiting from the semantic types

```
<form novalidate>
  <input type="email" id="addr">
  <input type="submit" value="Subscribe">
</form>
```

Q4 <fieldset>

- Group related form elements together
- Meaningful legend provides accessibility
- Can disable all contained elements

```
<fieldset>
  <legend>Your details</legend>
  <label for="fname">First name</label>
  <input id="fname" type="text">
  <label for="sname">First name</label>
  <input id="sname" type="text">
  <label for="age">Age</label>
  <input id="age" type="number">
</fieldset>
```

- Your details

First name
First name
Age

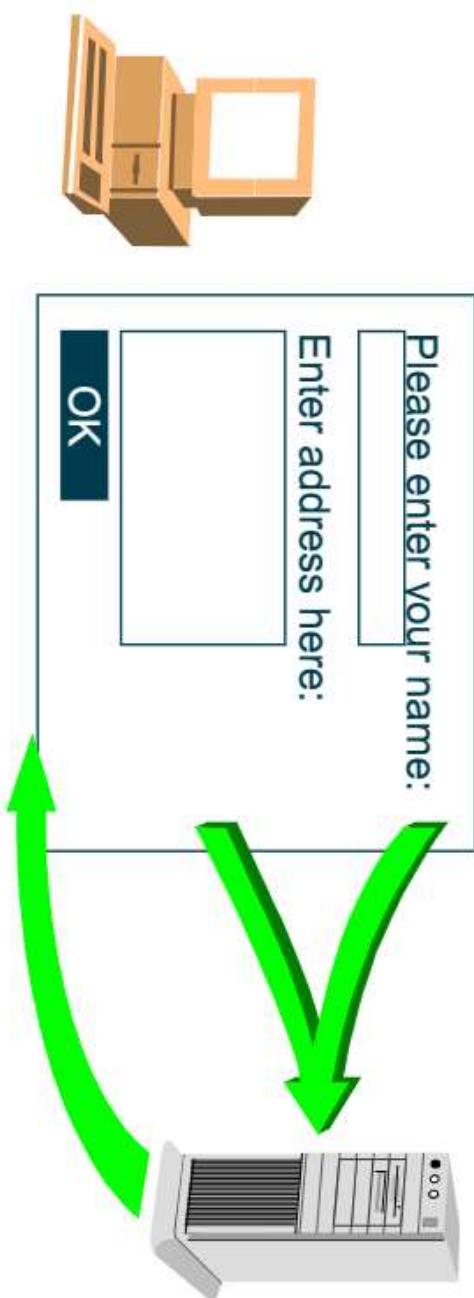
```
<fieldset disabled>
  ...
</fieldset>
```

- Your details

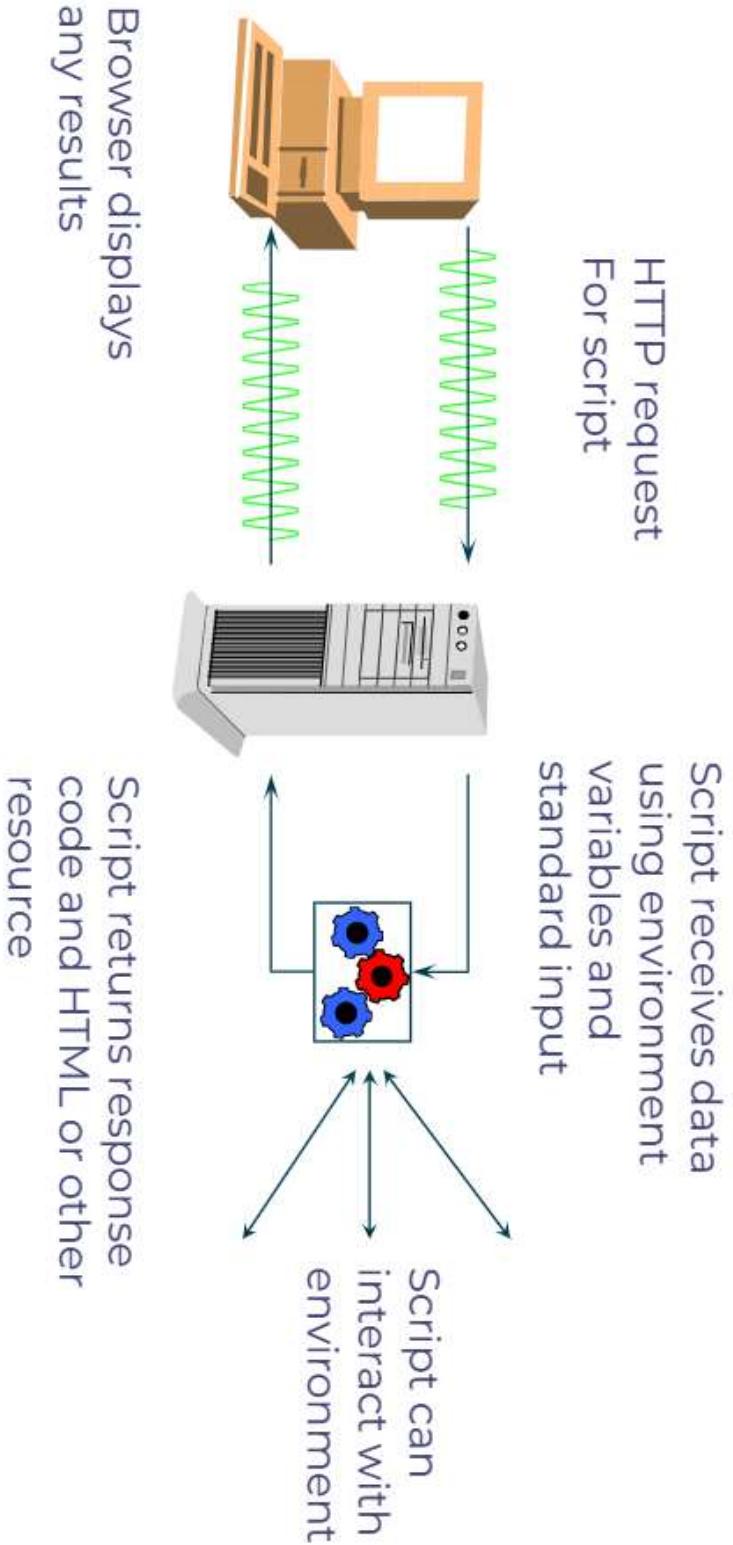
First name
First name
Age

Q1 Submitting the form

- The “submit” button triggers the script
- The values of the form’s fields are sent to the server
- Two different mechanisms: GET and POST



QA Server scripting mechanism



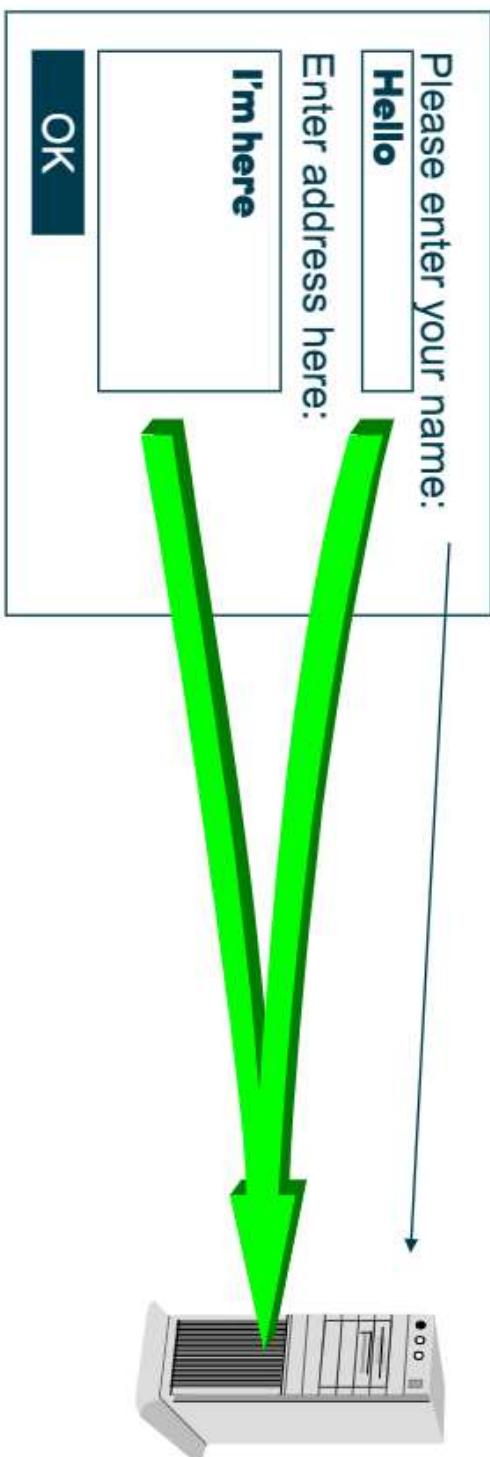
QA Form security issues

Scripts should check data received from a form

- Validate all data carefully!

Data sent from a form is not normally encrypted

- Unless a secure connection is made (e.g., SSL)





Quick Lab 5 – Forms

Create a form on a web page using different input types
and in-built validation



<head>

Basic HTML

A large, stylized red arrow points upwards and to the right, starting from the bottom left and ending near the top right. It has a thick, solid line and a white triangular cutout in the center, creating a sense of depth and motion.

Q& A The <head> tag

- A container for metadata usually placed between the <html> and <body> tags
- Metadata is defined as data about the data and is not displayed

Tag	Purpose
<title>	Defines title of document for browser tab, favourites and search engine results
<style>	Defines style information about this document
<link>	Defines where external style sheets should be linked from
<meta>	Specifies the page's character set, description, keywords, author, viewport, etc
<script>	Defines client-side JavaScript (although more commonly included as last element of <body>)
<base>	Specifies base URL and target for all relative URLs in a page



The DOM (Part 1)

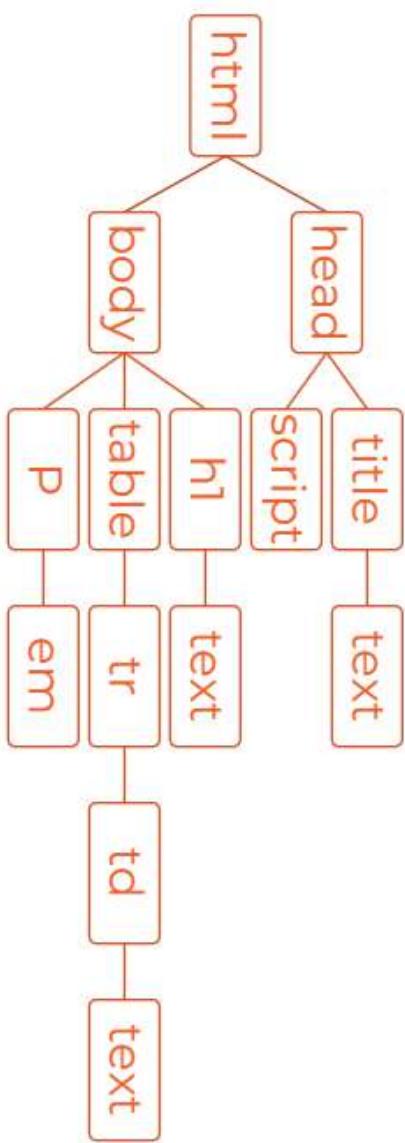
Basic HTML



Q A What is the Document Object Model?

HTML documents have a hierarchical structure that form the DOM

- Every element, except <html> is contained within another
- Creating a parent/child relationship



A DOM tree contains two types of elements

- Nodes.
- Text.

QA Inspecting HTML in the Browser



- Opening the Developer Tools in your browser will allow access to the Elements tab that shows the HTML.

```
Tools for introspection & HTML 5
[ ] Elements  Console  Sources  Network  Performance  Memory  Application  Security  Audits  Auditory  FlashControl
<!DOCTYPE html>
<html lang="en">
  <head>
    <body>
      <div id="heading"></div>
      <div id="latestPostSection">
        ...
        <div id="article">
          <div id="articleHeader"></div>
          <div id="mainArticle">
            <h1>My HTML5 Musings</h1>
            <p>id="intro">This is my introduction to HTML5</p>
            <p>id="postBody">This is the body of my post</p>
            <p>id="conclusion">This is the conclusion of my post</p>
          </div>
          <div id="articleFooter"></div>
        </div>
      </div>
    </body>
  <!— Code injected by live server —>
<html>
  <body>
    <div id="postsSection"></div>
  </body>
</html>
```

- Selecting markup in this window highlights the area displayed because of it (as shown)

Learning Objectives

How The Web Works

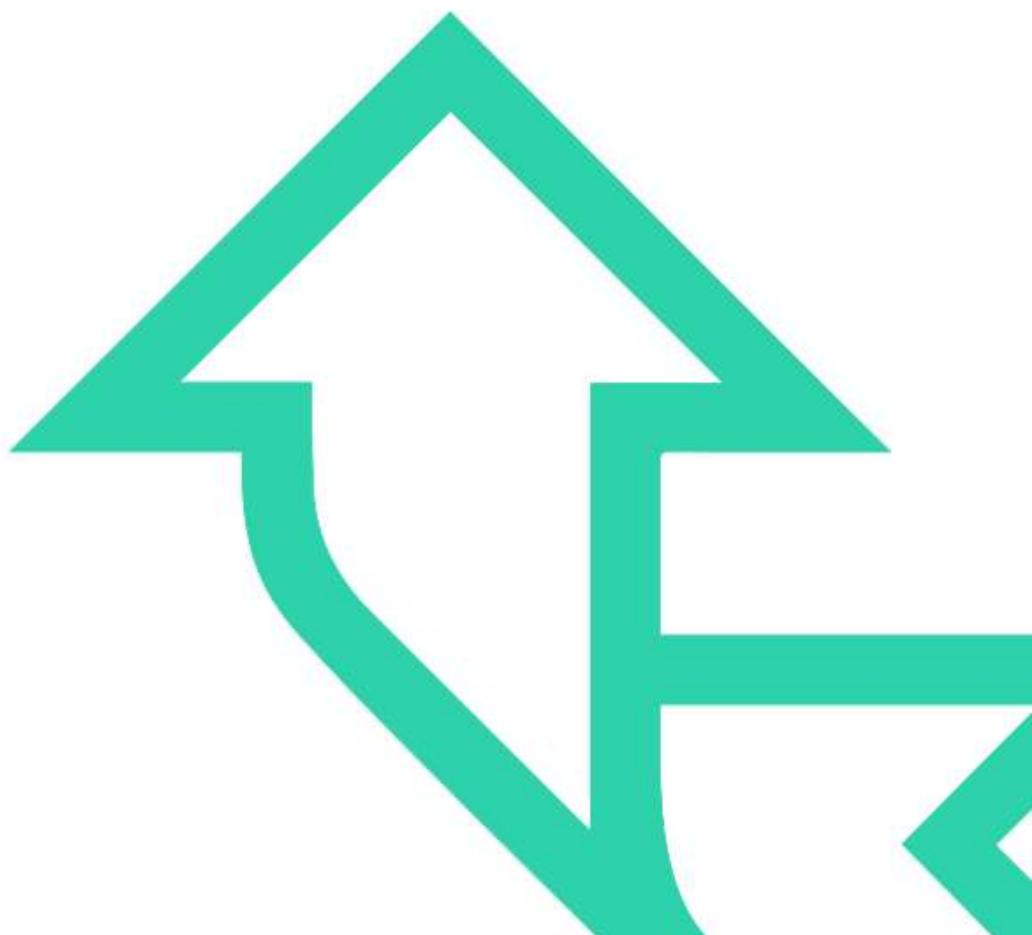
Basic HTML

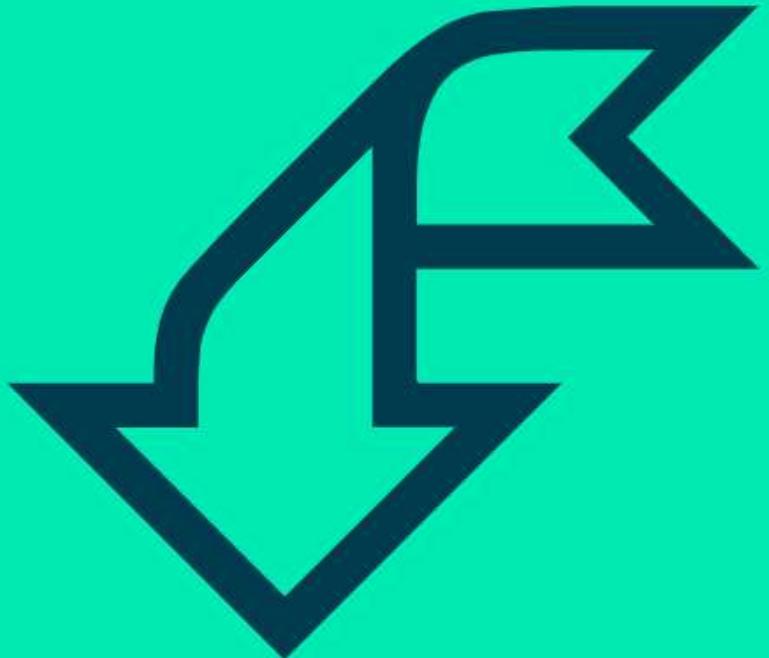
- HTML History and Syntax
- Structural HTML
- Hyperlinks
- Lists
- Tables
- Forms
- The <head> tag
- The DOM





Appendix





Appendix I

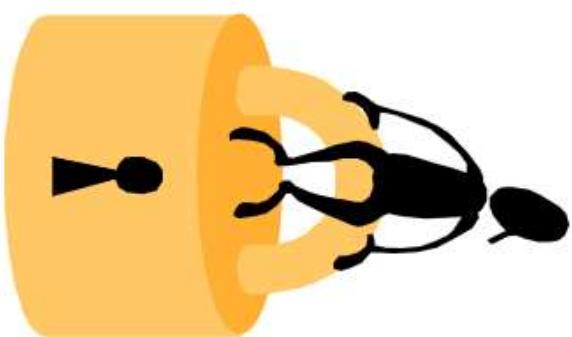
How The Web Works

- Encryption

Protection through Encryption

Encryption can provide protection from network data attacks

- Internet links are not secure
- Intranet links can be insecure
- Encryption provides protection from Data Snooping



Encryption Algorithms

Encryption algorithms have been developed over time

Development of Cryptography

- Early cryptography evolved well over 3000 years ago
- Complex mathematically secure cryptography evolved 80 years ago
- Modern public/private algorithms are 20 years old or younger

Modern algorithms break into two classes -

- Private Key (or Symmetric) algorithms
- Public Key (or Asymmetric) algorithms



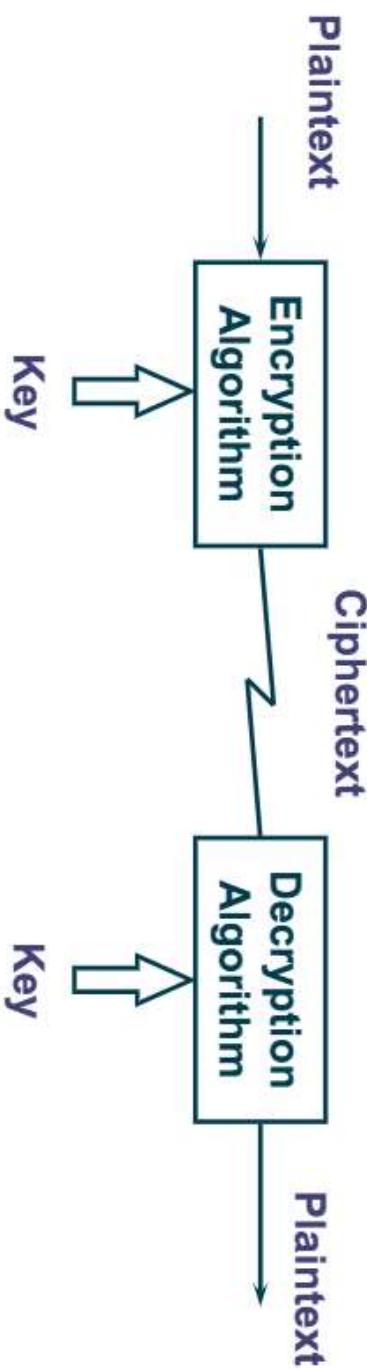
Q4 Private Key Algorithms

Private Key (or Symmetric Algorithms)

A single key is used for both encryption and decryption

Security depends on algorithm and key size

- Usually, the bigger the key size the better
- But larger key sizes impact on performance



Key Distribution And Management



Primary problem with Private Key algorithms

- How to securely transmit the key to other parties
- The key management problem
- Problematic until 1976 and the advent of public key cryptography
- Private key must be kept securely

Solutions

- Secure and restrict access to the private key
- A known secure communications link can transmit the private key

Current solutions

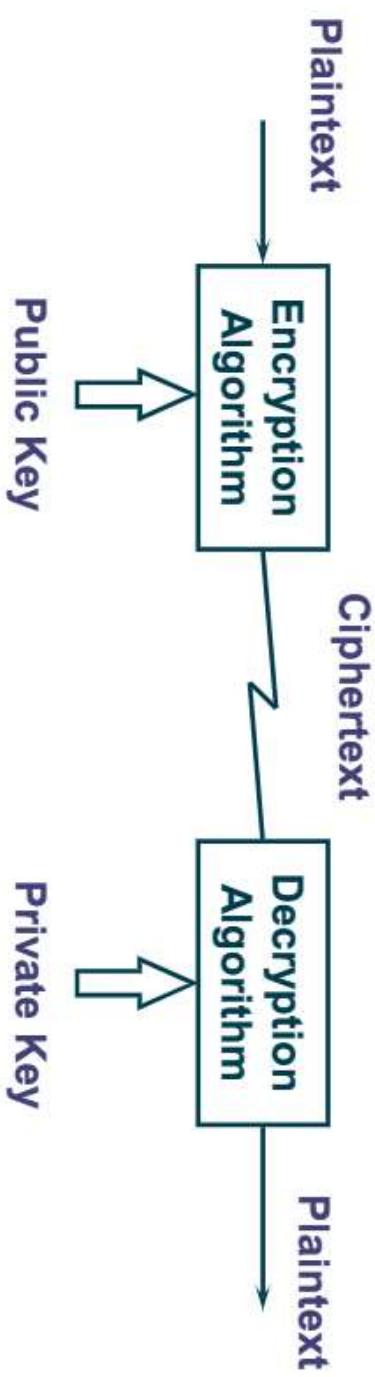
- Public key algorithms have become the modern solution to the key management problem

Public Key Cryptography

- Each entity (user etc) has two mathematically related keys
- The private key is told to no-one
- The public key is made freely available
- Data encrypted with one key can be decrypted with the other

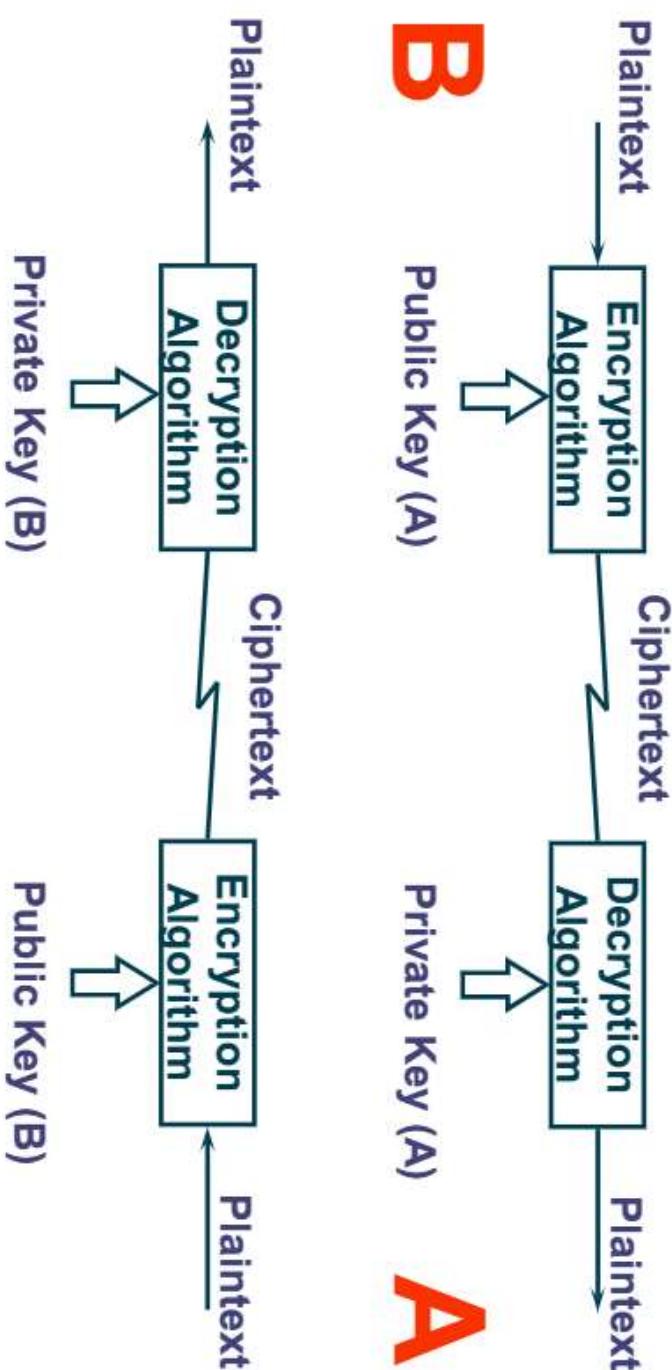
Q4 Public Key Algorithms

- Public Key (or Asymmetric Algorithms)
 - A pair of keys are used for encryption and decryption
 - One key is kept private and used for decryption, the other is made public and used for encryption
 - The private key cannot (practically) be derived from the public key



Q4 Public Key Algorithms (2)

- By using two pairs of keys, one for each party, a secure two-way channel can be established...



Public Key Algorithms (3)

Advantages

- Solves the key management problem

Disadvantages

- Slow performance (Symmetric algorithms are up to 1,000 times faster than Public algorithms)
- Needs large key sizes (512 bits is weak, 1024 is reasonably secure)
- Vulnerable to certain types of attacks (plaintext and timing attacks)

Protection Through Authentication



Encryption provides us with the ability to hide data from third parties - but ...

- Do we know the data came from who we think it came from?
- Do we know that it hasn't been tampered with?
- Do they positively know who we are?

Authentication provides the answers...

- Individual messages can be protected from tampering
- We can positively authenticate parties who connect to us
 - We can positively authenticate ourselves to other parties

Q&A Message Authentication

- To check that a message has not been altered en-route
- Hashing algorithm used to create a fixed-size message digest appended to the end
- Popular algorithms include MD5 and SHA

acak ack acckk

ack ack aackck

....

has someone
tampered with the
contents of
message ... ?

acak ack acckk ack

ack aackck

has someone
tampered with the
contents of message
... ?

101010101010

101010101010

Message

Message Digest

Message + Digest

Q4

Digital Signatures

How can we be sure that the contents and the message digest are both unaltered ... ?

We use “Digital Signatures”

- Public key encryption in reverse
- Data encrypted with a private key can only be decrypted using the corresponding public key
- Recipient is already in possession of a public key they know belongs to you
- If the message can be decrypted using public key, it came from you

In practice we combine digital signatures with message digests ...

- Public key encryption is too slow to use on the whole message
- So, we just sign the message digest so that cannot be tampered with

QA

Digital Certificates

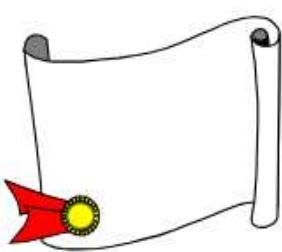
Certificates bind a public key to a particular entity

Certificates contain information about an entity...

- Certificate information, e.g., system name and public key
- Server's digital signature
- The most commonly used certificates are based upon the ITU-T X.509 standard
- X.509 certificates are those used in SSL and other common authentication mechanisms

Certificates are issued by a trusted body

- The Certification Authority
- You trust them to check that the certificate really does belong to whoever claims to own it



Certificates and Digital IDs

- A certificate contains your “distinguished name” and public key
- It is digitally signed by a “Certificate Authority”(CA)
- If you have obtained the CA’s public key from a reputable source, then you can verify that they issued this certificate
- If you trust the CA, then you can be sure that the holder of the certificate is who he claims to be



Q4

Certification Authorities

Issue Certificates

Verisign, RSA Data Security, etc...

Process of applying for certificate

- When applying for key, the CA needs detailed information on the person/company
- The CA's job it to ensure that you are who claim to be, how thorough they are determines how trustworthy their judgement is.
- Generally, the more expensive the certificate, the more checking and the more trustworthy

QA

Breaking Public Key Cryptography

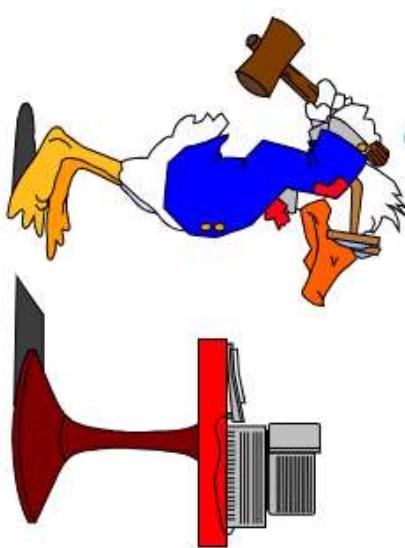
Obtain the private key by deception / bribery / etc

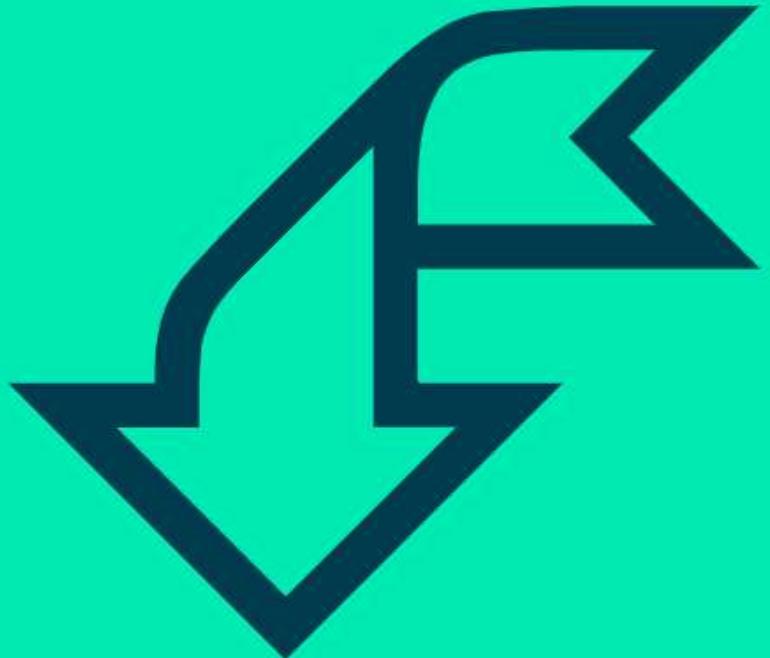
Deduce private key by “brute force”

- Try every possible key in succession
- Larger key size gives greater protection against brute force attack
- More possible keys

Standard “maximum strength” key is 128 bits

SSL2.0 uses 256 bit private keys





Appendix II

Image Types



Q4

GIF And JPEG Compared

GIF

- Defined originally by CompuServe
- Wide cross-platform support
- LZW lossless compression of images (typically 5:1)
- 256 colour limit
- Supports interlacing
- GIF89a variant supports animation and transparency

JPEG

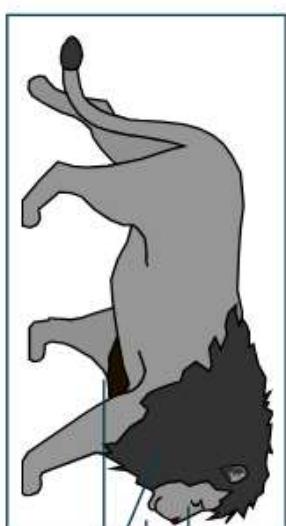
- Designed for photo-realistic images (e.g., scanned photographs)
- “Lossy” compression (typically 15:1)
- 16 million colours (24-bit)
- Progressive JPEG
 - Images display in series of passes, each pass contains more detailed information



QA Transparent GIFs

All colours in a GIF image are palette-relative

One colour may be the transparency colour



QA GIF and JPEG Tools: A Sample

Adobe Photoshop

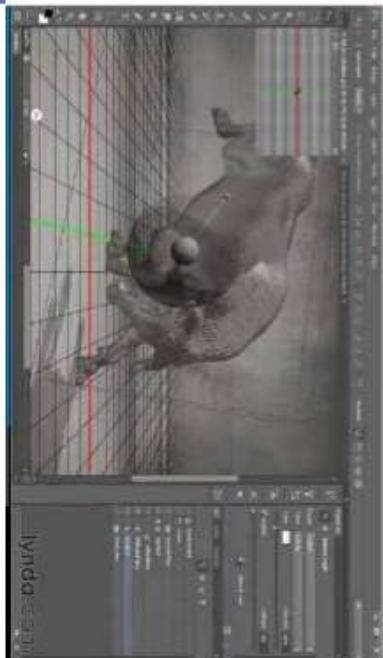
- Widely used by graphic designers
- Mac and PC variants
- Many “bells and whistles”
- Plug-ins for expandability
- <http://www.adobe.com>

Corel Paintshop Pro

- <http://www.corel.com>

LView Pro

- <http://www.lview.com/index.htm>



Q4

Portable Network Graphic



Transparency supported as well as the degree of transparency(opacity)

lossless compression of images

- 5% to 25% more than the GIF at 256 colour

True Colour also supported

- Although may be a larger file size than a JPEG

Supports interlacing

Gamma correction

- (cross-platform control of image brightness)

Does not support animation

Developed by an Internet Committee to be patent-free

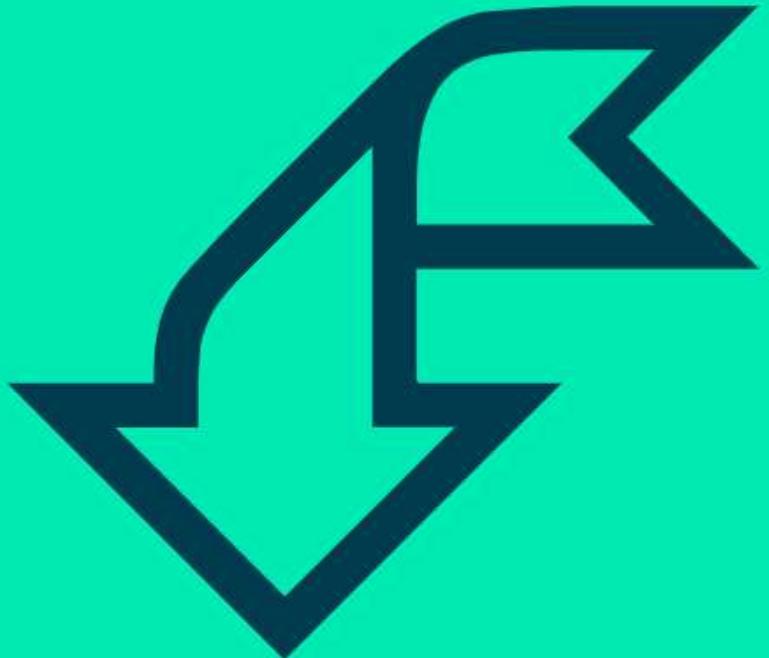


QA

WebP

Modern image format

- Provides superior lossless and lossy compression for images on web
- Allows creation of smaller, richer images to help make downloading faster
- Lossless images approximately 26% smaller than PNGs
- Lossy images are 25-36% smaller than comparable JPEGs
- Lossless supports transparency at cost of 22% additional bytes
- Typically 3x smaller files than PNG



APPENDIX III

GIT Essentials



Q4 Git Configuration – Setting the User

Git keeps track of who performs version control actions

- Git must be configured with your own name and email

To configure Git with your name and email we use the following commands:

```
% git config --global user.name "Your Name"  
% git config --global user.email "mail@example.com"
```

The values can then be accessed using:

```
% git config user.xxx
```

You can list all the configurations with:

```
% git config --list
```

Q4

Git User Configuration – Ninja Lab

Use the command line to configure Git with your user name and email

- Check the settings to ensure these are set



QA Git Commands – Entomology

Git commands all follow the same convention:

- The word 'git'
- Followed by an optional switch
- Followed by a Git command (mandatory)
- Followed by optional arguments

```
git [switches] <commands> [<args>]
```

```
git -p config --global user.name "Dave"
```

QA Getting started with Git

When you first launch Git you will be at your home directory

- ~ or \$HOME

Through the Git Bash you can then move through and modify the directory structure

Command	Explanation
ls	Current files in the directory
mkdir	Make a directory at the current location
cd	Change to the current directory
pwd	Print the current directory
rm	Remove a file (optional -r flag to remove a directory)

Q4

Git Bash making a directory - Ninja Lab

Launch your git command line

- Ensure you are at your home directory
- Find the corresponding directory using your GUI file explorer
- Create a directory called gitTest
- Navigate within the directory using the command line
- Create a sub-folder
- Navigate back to home
- Remove the gitTest directory
- Once you have completed all other tasks type history

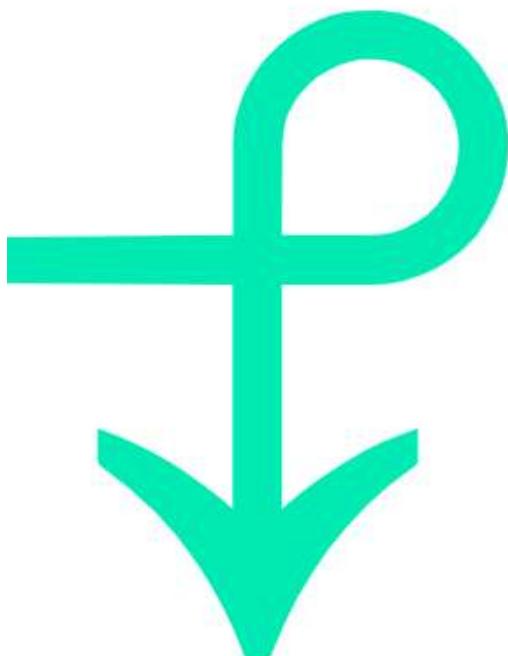


On Git Help

```
git help
```

QA

Git Help - Ninja Lab



Use your command line to access git help find
out about:

- help
- glossary
- -a
- config
- -g



QA GIT Key Concepts - Repos

GIT holds assets in a repository (repo)

- A repository is a storage area for your files
- This maps to a directory or folder on your file system
- These can include subdirectories and associated files

```
$mkdir firstRepo  
$cd firstRepo  
$git init
```

The repo requires no server but has created a series of hidden files

- Located in .git folder

```
$git status
```

Q1 Adding Files

You can see the status of your git repository

```
$ git status
```

On branch master

Changes not staged for commit:

(use "git add <file>..." to update what will be committed)

(use "git checkout -- <file>..." to discard changes in working directory)

```
modified: DG_02_Git.pptx
```

no changes added to commit (use "git add" and/or "git commit -a")

Q1 Adding Files

To add a new file, use the 'add' argument

```
# a single file  
$ git add specific_file_name.ext  
  
# To add all the files  
$ git add .  
  
# add changes from all tracked and untracked files  
$ git add --all
```

Git status will show the newly added file

```
On branch master  
Changes to be committed:  
(use "git reset HEAD <file>..." to unstage)  
  
modified:   DG_02_Git.pptx
```

QA Committing files

To commit the changes, use the “commit” argument

- You can specify the author with the -a flag
- The -m flag is used to set a message

```
$ git commit -m "More content for DG02 - git"  
[master 93e8300] More content for DG02 - git  
 1 file changed, 0 insertions(+), 0 deletions(-)
```

This is now saved to the local version of your repository

Creating A Git Repo - Ninja Lab

- Create a folder at the ~ called firstRepo
- Initialise it as a Git repository
- Check the status of the repo
- Use the touch command to create a file called myfile.bat
- Check the status of the repo



QA Attack of the Clones!

Cloning makes a physical copy of a Git repository

- It can be done locally or via a remote server, e.g. GitHub
- You can push and pull updates from the repository

The benefit of cloning repos is that the commit history is maintained

- Changes can be sent back between the original and the clone

Cloning is achieved with the clone command:

- Or through the GUI

```
$git clone source destination_url
```

The GUI branch visualiser gives us a very useful way to see the origins of branches

Q4 Cloning a repository

Cloning copies the entire repository to your hard drive

- The full commit history is maintained

To clone a remote repository:

```
$ git clone [repository]
```

```
$ git clone https://bitbucket.org/username/repositoryname
```

To clone a specific branch:

```
$ git clone -b branchName repositoryAddress
```



Web Fundamentals

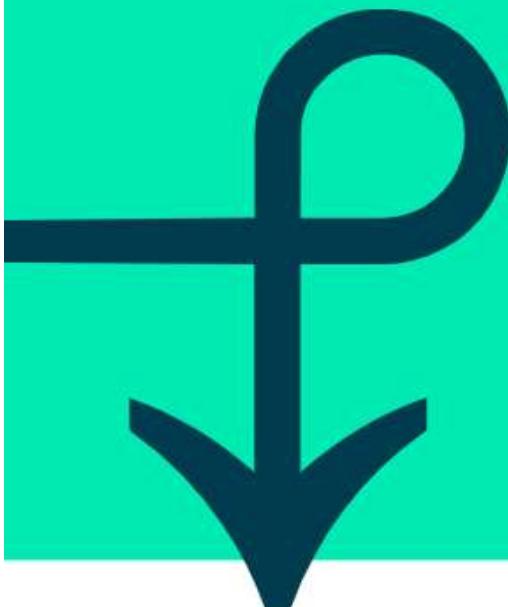
css



Q4

Learning Objectives

- Understand what CSS is
- Understand how CSS can be applied to web pages
- Understand the syntax of writing CSS rules
- Be able to select elements to apply CSS to
- Be able to work with text, colours, and images
- Be able to work with the box model and position elements
- Be able to style lists and tables
- Be able to add CSS animations, transforms, and transitions to elements



Q1

Cascading Style Sheets

- CSS application and syntax
- CSS and the DOM
- Inheritance and Selecting Elements
- Text and Colours
- Measurement Units
- Images and Backgrounds
- The Box Model
- Lists and Tables
- Animations, Transitions and Transformations





CSS Application and Syntax

CSS Fundamentals

CSS stands for Cascading Style Sheets

Cascading Style Sheets

It describes how HTML elements are to be displayed

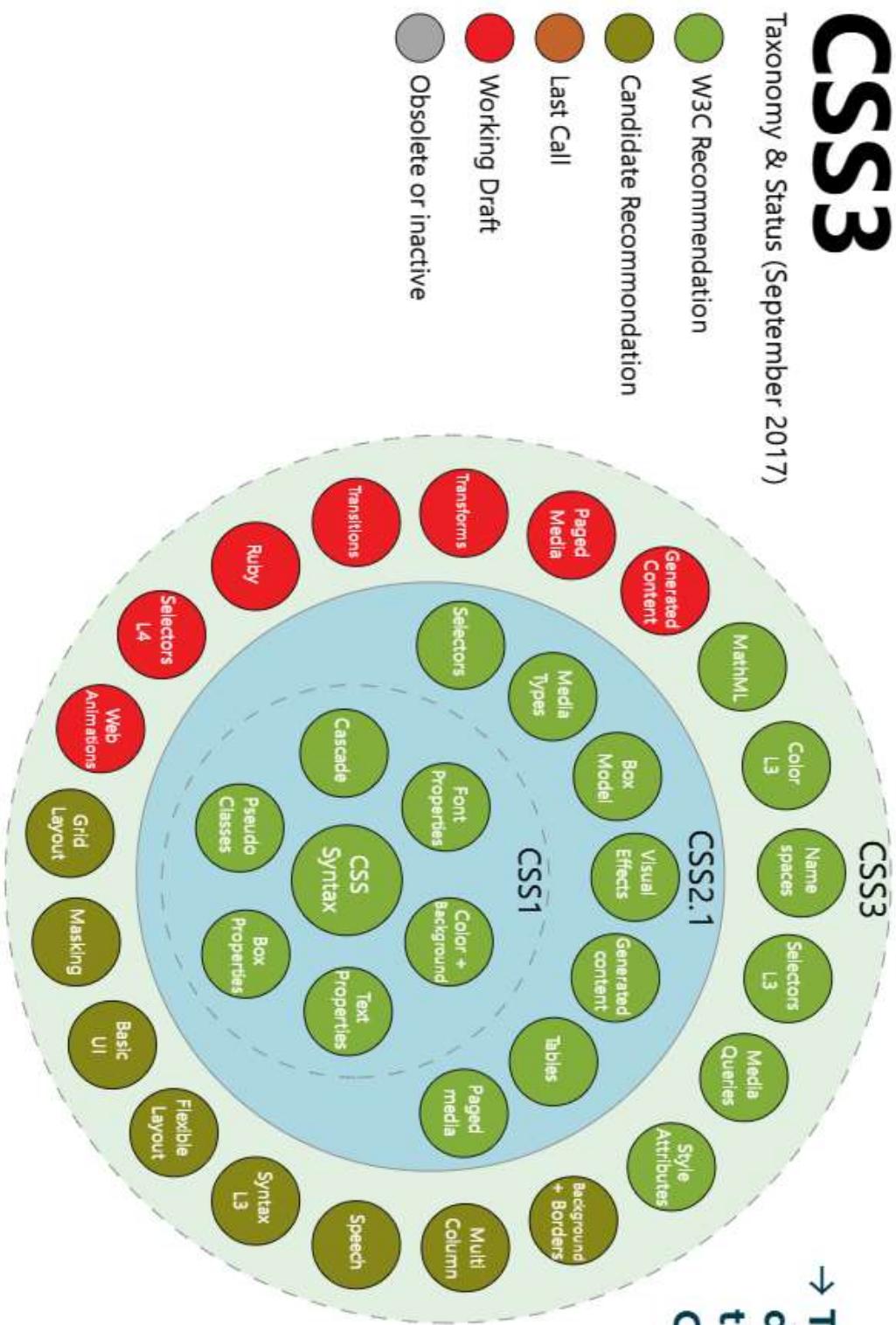
- This could be on the screen, on other media, or even how it should be printed on paper

Can control the layout of multiple web pages from a set of rules

Styling can be applied in one of four ways:

- Inline – defined in the actual element to style
- In an embedded stylesheet on the page – defined on a per-page basis
- In an external style sheet linked to the page – defined inside a separate .css file
- By linking in some existing CSS (almost never to be used)

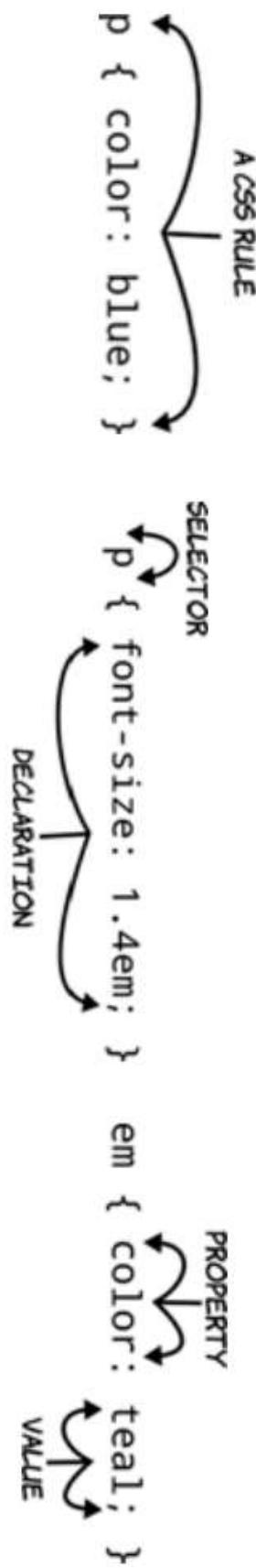
QA CSS Taxonomy And Status (2017)



→ This is a popular diagram to show the history of CSS progression

CSS Basic Syntax

- Rules, selectors, properties, and values
- A CSS style sheet is made up of rules
- Here are three examples CSS rules:



QA Inline Styles

- style attribute can be used on any HTML tag
- Affects that HTML tag only

```
<p style="margin-left: 1in; margin-right: 1in; line-height:200%">  
This text will be shown with one-inch left and right margins, and  
double-spaced.  
</p>  
<p>  
This text is formatted as normal for &lt;p&gt; tags.  
</p>
```

QA Embedded Style Sheets

Use <style> ... </style> inside the <head> tag

A style sheet definition contains a list of

- HTML tags, and
- Associated format information for that tag

```
...  
<style>  
  h1 { font-size: 15pt; font-weight:bold}  
  p { font: bold italic 12pt/20pt times, serif}  
</style>  
...
```

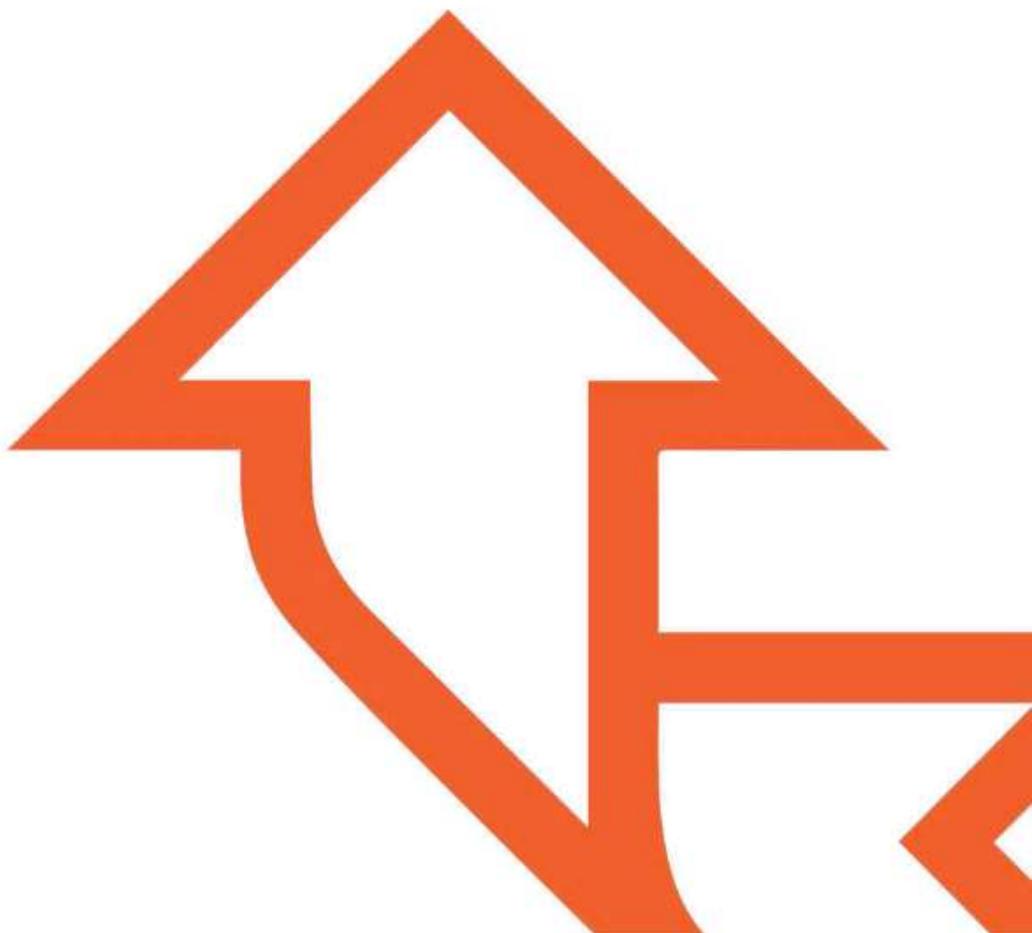
Q4 External Style Sheets

- Put all CSS in separate file and then link to it from each page
- In same format that it appeared in the Internal Style Sheets
- <link> element references an external style sheet
- Should appear in the head of the document

```
<link href="styles.css" rel="stylesheet">
```



CSS and the DOM

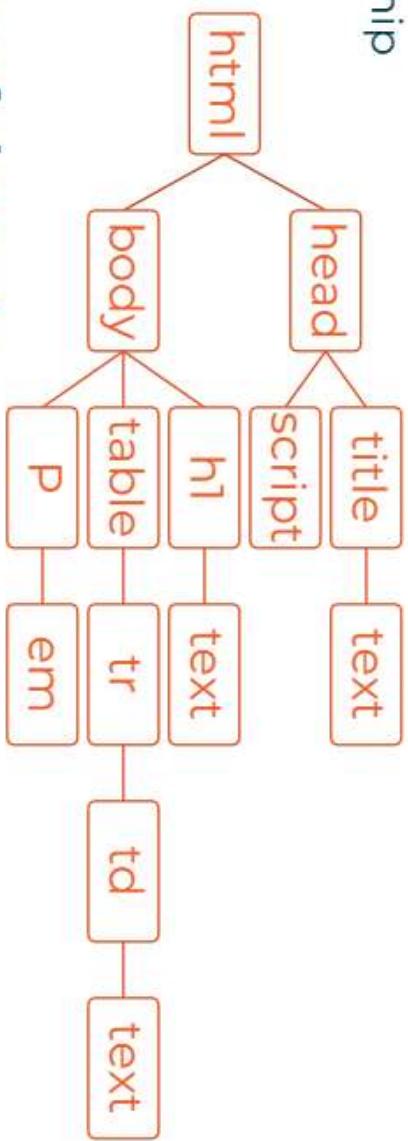


CSS Fundamentals

QA Recall: the Document Object Model

HTML documents have a hierarchical structure that form the DOM

- Every element, except <html> is contained within another
- Creating a parent/child relationship



A DOM tree contains two types of elements

- Nodes.
- Text.

Q4 HTML markup to DOM object (1)

Consider the following HTML

```

```

The tag has a type of `` and four attributes

- `id`
- `src`
- `alt`
- `title`

The element is read and interpreted by the browser into a DOM

- Each element becomes a NodeList object
- Assigned a property based on the html attribute

Q4 HTML markup to DOM object (2)

img element

- id: 'myImage'
- src: 'http://'
- alt: 'My image'
- class: 'someClass'
- title: 'This is my image'

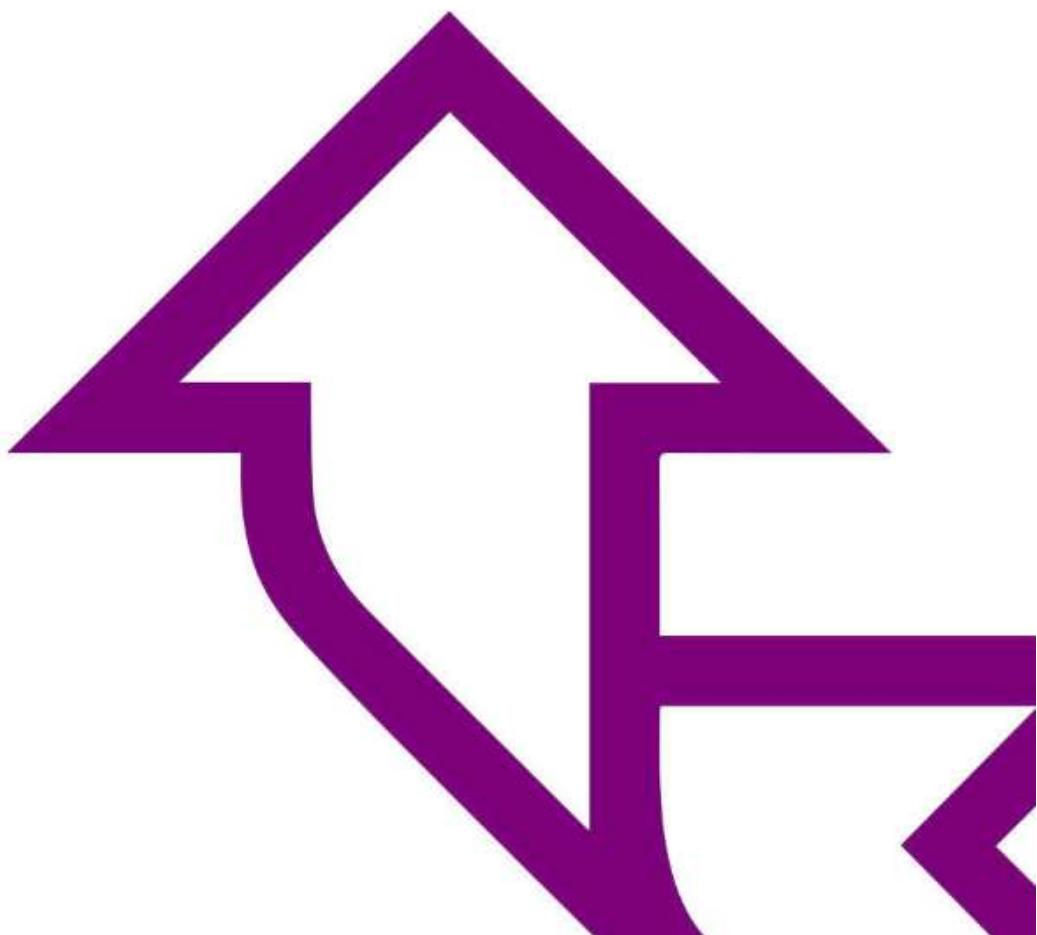
NodeList

- id = 'myImage'
- src = 'http://'
- alt = 'My image'
- className = 'someClass'
- title = 'This is my image'

-
- The diagram illustrates the translation process. On the left, a pink rounded rectangle contains the original HTML code. A green curved arrow points from this box to a light blue rounded rectangle on the right, which contains the resulting NodeList of DOM objects. The pink box is labeled 'img element' and the blue box is labeled 'NodeList'. The green arrow represents the conversion process.
- HTML is translated into DOM elements, including the attributes of the tag and the properties created from them.
 - These can be used to select and style elements



Inheritance and Selecting Elements



Selecting Elements To Style

- Select all tags of a particular type
- Select tags dependent on the relationship to others in the DOM
- Select tags based on their id or class attribute
- Select tags based on other attributes
- Select tags based on a combination of the above

Q1 Selecting all tags of a particular type

This is as simple as creating a rule for the tag name and nothing else

All elements with this tag will be affected, regardless of where they are in the DOM

• Assuming that this is the only CSS applied to the page (more on that later...)

```
p { color: green }
```

- Would make all text in any p element green

```
div { background-color: red }
```

- Would make the background colour of any div element red

• Multiple elements can be selected by putting a comma between them

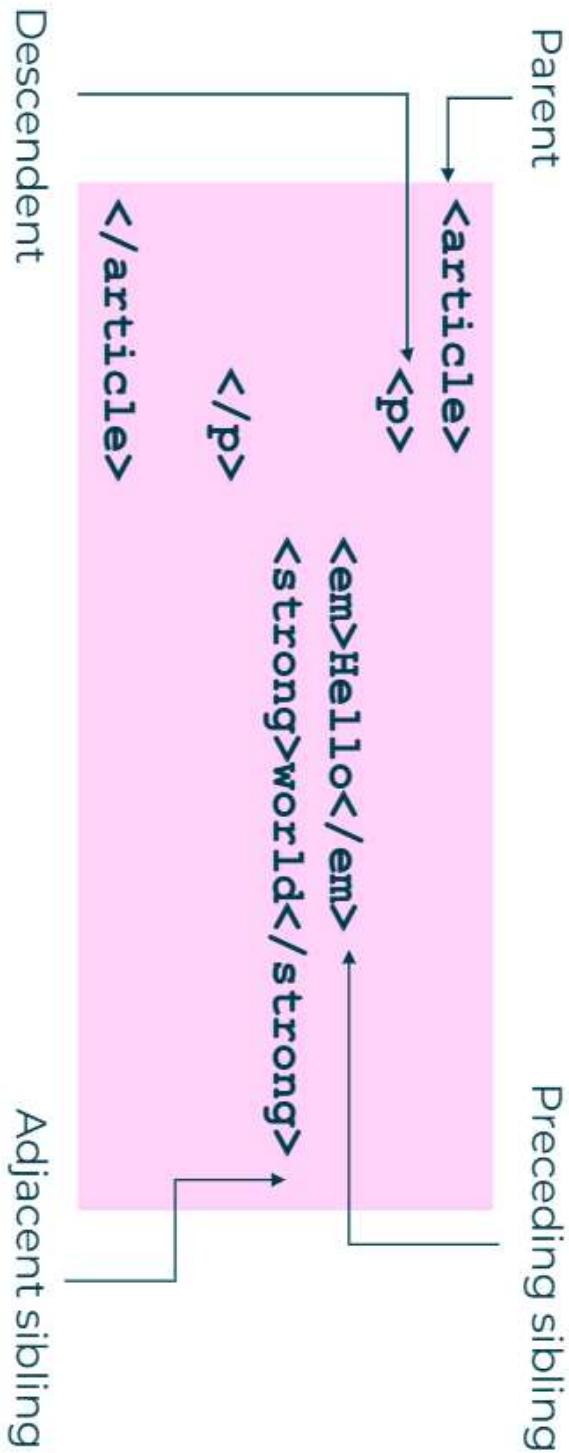
```
h1, a {color: pink}
```

- Would make the text of any h1 element and any a element pink

QA Understanding Inheritance

HTML tags exist in a hierarchical tree from <html> root to text nodes

- When a tag is surrounded by another tag the tags are nested.



QA Hierarchical Inheritance

Elements inherit from containing parents

- So we only need to define a style rule at the highest level
- We can then override rules at descendant levels

Complex hierarchies are difficult to manage

- Chrome's developer tools help greatly
- Showing which styles are applied
- Where they come from
- If rules are being overridden
- The order in which they're applied

```
▼ Styles + ↻ ⓘ
element.style {
}
Matched CSS Rules
.main article h1 { main.css:156
  font-size: 2em;
}
header h1 { main.css:22
  margin-bottom: -0.5em;
}
h1, h2, h3 { main.css:25
  font-family: 'ChunkFiveRegular', Arial,
  sans-serif;
}
h1 { normalize.min.css:2
  font-size: 2em;
  margin: 0;
}
```

Q A Select tags dependent on the relationship to others in the DOM

Descendant selector – put a space between the parent and child – all descendants will be styled

```
ul li { color: purple }
```

- Would make all text in any `li` that is a descendant of a `ul` **purple**

Child selector – put `a >` between the parent and child – any direct child will be styled

```
section > p { color: brown }
```

- Would make all text in any `p` that is a *direct descendant* of a `section` **brown**

Adjacent selector – put `a +` between the siblings – last sibling listed will be styled

```
h2 + p { color: black }
```

- Would make all text in any `p` that *immediately follows* a `h2` element will be **black**

Sibling selector – change the `+` for a `~` to select **any following element**

Q& Select tags based on their id or class attribute

id selector – put a # before the name of the id to be styled

```
#myChosenId { color: purple }
```

- Would make all text in any **element** that has an **id** attribute of **myChosenId** **purple**
- **Note:** an **id** should be *unique* within a page, if *more than one* is needed a **class** should be used

class selector – put a . Before the name of the class to be styled

```
.myChosenClass { color: brown }
```

- Would make all text in **any element** that has a **class** attribute of of a **myChosenClass** **brown**

```
<p id="myChosenId">A paragraph with myChosenId</p>
<p class="myChosenClass">A paragraph with myChosenClass</p>
```

QA Selecting sets of elements with pseudo-classes

Selecting first and last element

```
ul li:first-child { background-color: red; }  
ul li:last-child { background-color: red; }
```

Selecting an element by its ordering

```
li:nth-child(3), li:nth-child(5) { background-color: red; }  
li:nth-child(2n + 1) { background-color: red; }  
li:nth-child(odd) { background-color: blue; }  
li:nth-child(even) { background-color: green; }
```

QA Selecting sets of elements with pseudo classes

More selection patterns

```
ul:last-child { background-color: red; }
```

```
ul:first-child:last-child { background-color: red; }
```

Selecting by types of element

```
article:first-of-type { background-color: red; }
```

```
p:last-of-type { background-color: red; }
```

Choosing what isn't

```
input:not([type=checkbox]):not([type=radio]) {  
    display: block; width: 12em;  
}
```

QA Pseudo classes and elements

(Dynamic) pseudo classes, elements...

- Applying to user actions (pseudo classes)

```
:active { color: blue; }
:hover { color: blue; }
:focus { color: blue; }
:link { color: blue; }
:visited { color: blue; }
```

- Applying to placement (pseudo elements)

```
:after { color: blue; }
:before { color: blue; }
:first-letter { color: blue; }
:first-line { color: blue; }
```

- Selection pseudo element

```
::selection { color: blue; }
```

QA Pseudo classes specificity

Recall: in the cascade styles are sorted by specificity

- Latter rules are more specific than earlier rules

Hence for link pseudo classes to work use this order

```
a {color: black; }  
a:link {color: blue; }  
a:visited {color: red; }  
a:hover {color: green; }  
a:active {color: orange; }
```

Q\ before: and after:

Used to insert content before or after an element

- Can be specific content, counters or values of attributes

Specify style and content of inserted content

- content: normal | none | <string> | <uri> | <counter> | attr(<identifier>) | open-quote | close-quote | no-open-quote | no-close-quote | inherit

```
p.note:before { font-weight: bold; content: "Note: " }

h1:before {
  content: "Chapter " counter(chapter) ". ";
  counter-increment: chapter;
}
```

QA Choosing elements by their attribute

- `=` operator finds attributes whose value exactly matches
`a[href="http://www.qa.com"] { color: blue; }`
- `^=` operator finds attributes starting with a value
`a[href^="http:"] { color: blue; }`

- `$=` operator finds any element attributes ending with a value
`[src$=".png"] { color: red; }`

- `*=` operator finds attributes containing the value

```
[id*="stuff"] { color: red; }
```



Quick Lab 6 - CSS Selectors

Apply selectors to style rules to apply styling to particular elements



Text and colours

A large, bold, red 'A' character is centered at the bottom of the slide. It has a thick stroke and a slightly irregular, hand-drawn style.

Q& Working with fonts – setting the character type

As previously noted it is important to set the encoding type of a document.

- In HTML5

```
<meta charset="utf-8">
```

- In XHTML/HTML4

```
<meta http-equiv="Content-Type" content="text/html; charset="utf-8" />
```

A character set is a list of character codes your browser will accept

- If it does not understand a character, a glyph will appear in its place
- You are also leaving yourself exposed to dangerous JavaScript attacks



QA Font Families

CSS Defines five font families to which most fonts are categorised

- Serif - Times New Roman
- Sans-serif - Arial
- Monospace - Courier New
- Cursive - Brush Script
- Fantasy - Papyrus

font-family: Helvetica, Verdana, Arial, sans-serif

There are also dingbats and other symbol library fonts

Plus HTML character entities

- £ for a GBP symbol as an example

Fonts are set in a comma delimited list

- Browser checks if font is available, used if present moves on if not

Q4 Other Font Settings

- Additional typography properties can be set:

Property	Usage
font-size	Font size can use any of the units previously discussed or a value between xx-small and xx-large
font-weight	font-weight controls the normal weight of the font normal bold or a weight scale between 100 and 900
font-style	Normal, italic or oblique – if no oblique is present italic will be used.
line-height	The height of each line of text known as leading
vertical-align	Sets the alignment of the text in relation to the line box.

Q& Setting fonts as a compound rule

Fonts need to be set in a very specific way using CSS.

- Requiring a minimum set of keywords and a specific order
- The most basic rule requires:

```
font: <font-size> <font-family>;
```

When using a complex rule optional values precede the mandatory

```
font: italic small-caps 1.2em Georgia, serif;
```

With the exception of a sneakily inserted line-height

- Note the lack of measurement unit
- You can add them but it can cause issues

```
font: 100%/2.5 Helvetica,
```

QA Text Alignment and Other Properties

Property	Description	Common Values
<code>color</code>	Sets the text colour for this and child elements	Any valid colour
<code>text-align</code>	Sets the horizontal alignment of text	<code>left</code> , <code>right</code> , <code>center</code> , <code>justify</code>
<code>text-decoration</code>	Sets or removes decorations from text	<code>none</code>
<code>text-transform</code>	Specifies case for text	<code>uppercase</code> , <code>lowercase</code> , <code>capitalize</code>
<code>text-indent</code>	Specifies indentation of first line of text	Any valid measurement
<code>letter-spacing</code>	Specifies space between characters in text	Any valid measurement
<code>line-height</code>	Sets space between lines	Any valid measurement
<code>text-direction</code>	Changes the direction of text	<code>rtl</code> , <code>ltr</code>
<code>word-spacing</code>	Sets space between words	Any valid measurement
<code>text-shadow</code>	Adds shadow to text – gives horizontal, vertical and colour of shadow	<code>3px 3px green</code>

QA Adding a drop shadow

Drop shadow is back as if the 1990's never happened!

```
.shadow {  
    text-shadow: 10px 8px 20px rgb(56, 52, 153);  
}
```

text-shadow requires the following properties:

- X, Y offset
- Amount of blur
- Colour
- Corresponding box shadow rule

```
.shadow {  
    box-shadow: 3px 3px 3px 3px rgb(0, 0, 119);  
}
```

QA Color values and format

W3C specifies 4 numerical colour value methods:

- RGB, RGBA, HSL and HSLA

There are also 16 basic named colour values that can be used in CSS

- Additional 128 colours are named in the extended set

<https://www.w3.org/TR/2018/REC-css-color-3-20180619/#svg-color>

Color names and sRGB values				
Named	Numeric	Color name	Hex rgb	Decimal
		<i>black</i>	#000000	0,0,0
		<i>silver</i>	#C0C0C0	192,192,192
		<i>gray</i>	#808080	128,128,128
		<i>white</i>	#FFFFFF	255,255,255
		<i>maroon</i>	#800000	128,0,0
		<i>red</i>	#FF0000	255,0,0
		<i>purple</i>	#800080	128,0,128
		<i>fuchsia</i>	#FF00FF	255,0,255
		<i>green</i>	#008000	0,128,0
		<i>lime</i>	#00FF00	0,255,0
		<i>olive</i>	#808000	128,128,0
		<i>yellow</i>	#FFFF00	255,255,0
		<i>navy</i>	#000080	0,0,128
		<i>blue</i>	#0000FF	0,0,255
		<i>teal</i>	#008080	0,128,128
<i>aqua</i>			#00FFFF	0,255,255

Q1 Color values and format RGB/RGBA

Used to specify RED, GREEN and BLUE values

- Can be done with Hexadecimal or as a set of 3 numeric values (either integer or percentage)

```
em { color: blue; }          /* #rgb */  
em { color: #ff0000; }       /* #rrggbb */  
em { color: rgb(255,0,0); }  
em { color: rgb(100%, 0%, 0%); }
```

The A value can be used to represent ALPHA for opacity of the colour

- Cannot be used with HEX values

```
em { color: rgb(255,0,0); }      /* integer range 0 - 255 */  
em { color: rgba(255,0,0,1); }    /* the same, with explicit opacity of 1 */  
em { color: rgb(100%,0%,0%); }    /* float range 0.0% - 100.0% */  
em { color: rgba(100%,0%,0%,1); } /* the same, with explicit opacity of 1 */
```

QA Color values and format - HSL/HSLA

- RGB is hardware oriented and harps back to the days when CRT were used in monitors

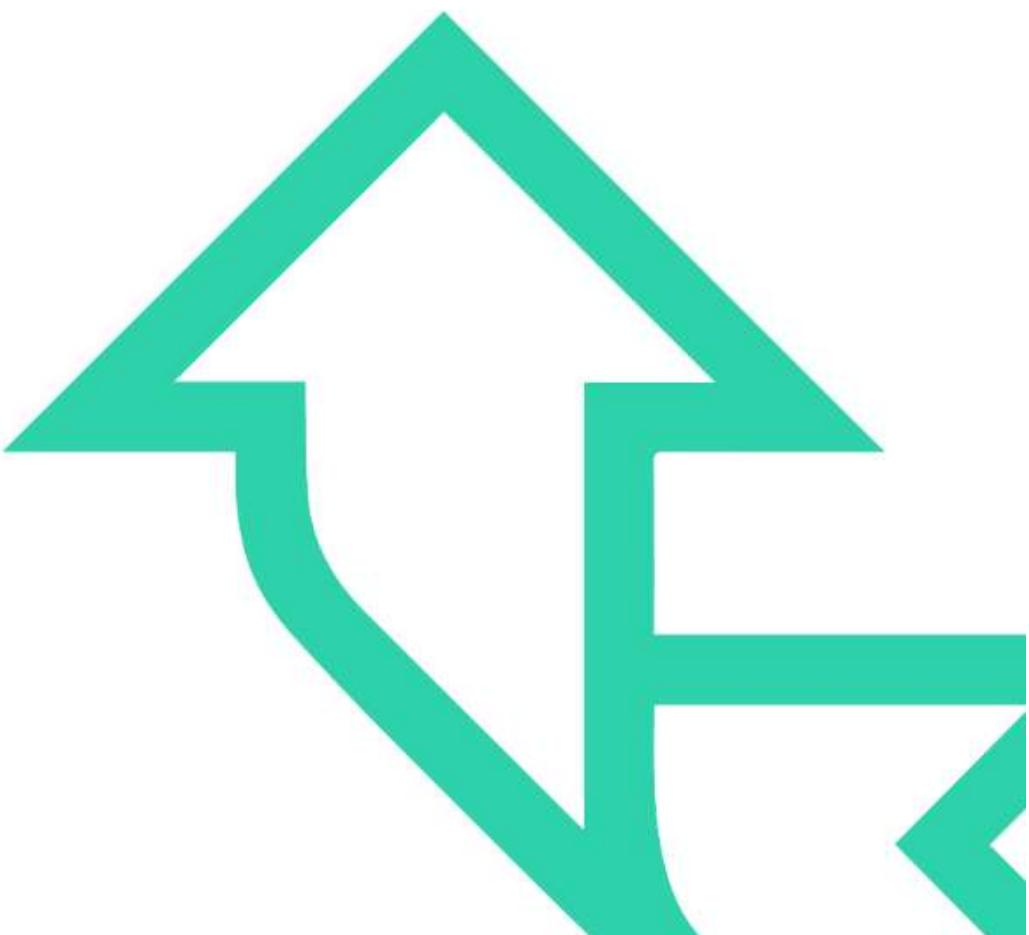
- **HSL are encoded as Hue, Saturation and Lightness**

- Hue is represented as an angle of the colour circle – measured in degrees and value is used in CSS
 - Saturation and Lightness are represented as percentages
 - 100% is full saturation and 0% is a shade of grey
 - 0% lightness is black, 100% is white, and 50% is 'normal'

```
* { color: hsl(0, 100%, 50%); }          /* red */
* { color: hsl(120, 100%, 50%); }        /* lime */
* { color: hsl(120, 100%, 25%); }         /* dark green */
* { color: hsl(120, 100%, 75%); }         /* light green */
* { color: hsl(120, 75%, 75%); }          /* pastel green, and so on */
```



Measurement Units



CSS Fundamentals

QA Element Sizing

Sizing elements can be achieved in a number of different ways:

Pixels (px) - a fixed measurement based on the size of a pixel

```
img { width: 150px; }
```

Em (em) - a relative unit that equates to the font size of the element.

- An em unit is relative to the parent element's font size.

```
article{ width: 3em; }
```

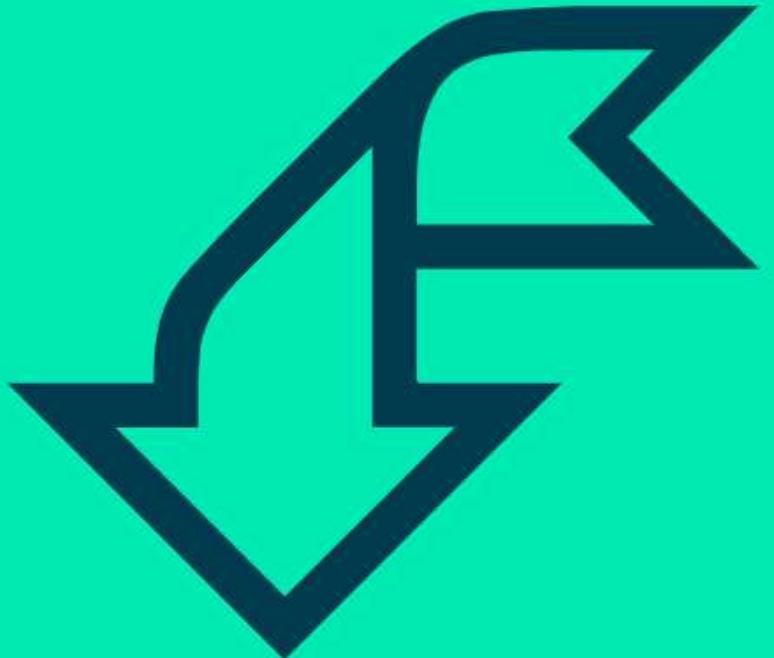
Points (pts) - Points are an absolute unit equal to 1/72 of an inch

- Points can be useful when setting type sizes for print

```
body{ font-size: 12pt; }
```

% - Size is relative to the containing element

```
p{ width: 50%; }
```



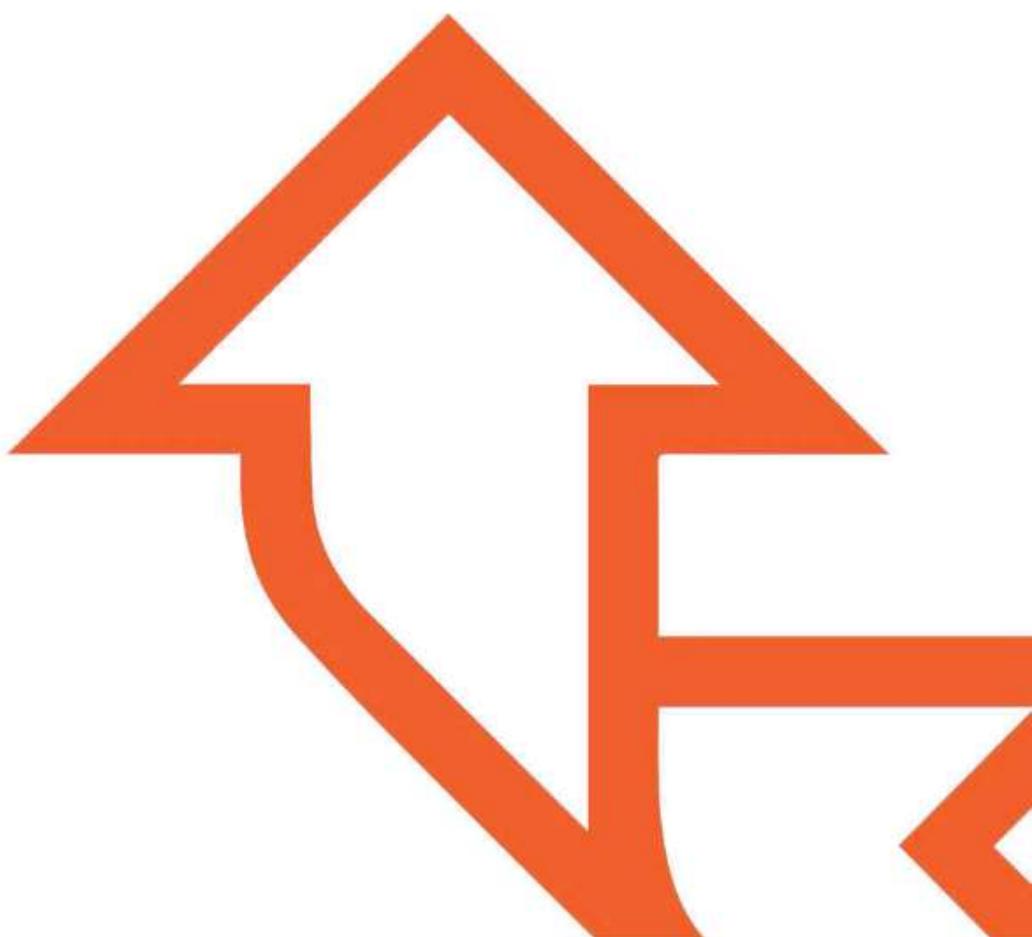
Quick Lab 7 – Text, Colours and Sizing

Experiment with adding colours and sizing to text and elements



Images and Backgrounds

CSS Fundamentals



QA Setting Image Properties

The markup for an image often contains the height and width

```

```

Working towards multi device display we should avoid this

- Creates a hard coded appearance rule for a graphic

Attributes such as width and height can be set in the CSS

```
  
  
#electric {  
    width: 450px;  
    height: 372px;  
}
```

Positioning, borders and spacing must be done with the box model

- Never use inline attributes

QA Page Background Colour and Images

The background can be a colour fill

- Use background-color CSS property of the <body> tag

```
<body style="background-color: #FFFFFF;">  
<!-- White background -->
```

Alternatively, you can tile an image

- Use background-image CSS property of the <body> tag

```
<body style="background-image: url('paper.jpg');">  
<!-- Background of paper texture -->
```

- Can also set whether to repeat and position

QA Element Background Images

The background of an element can be set using the url property

• The CSS requests an image asset using the url property

```
background-image: url ('./img/thumb/mountain.jpg');
```

The following properties can also be set:

- **repeat**
 - Sets whether the image tiles appear only once
 - Or repeat only horizontally or vertically
- **attachment**
 - Sets whether the image scrolls with the rest of the page or stays in one place
- **position**
 - Moves the image to the left and down (positive values) or to the right and up (negative values)
 - Calculated from the top-left corner of the parent element.

Background Images



size

- Sets the width and height of the image within the element
 - As an absolute length or percentage

clip

- Sets if the background fits to the border or within the content area

origin

- Sets the position of the background relative to the border, padding, or content

Multiple background images

- CSS3 allows you to layer multiple background images
 - Uses a comma-separated list

So Or Background- image?



Pros for

- Use plus alt attribute if the image is part of the content
- Use when the image has an important semantic meaning, such as a warning icon.
- **This ensures that the meaning of the image can be communicated in all user-agents**
- **Including screen readers.**
 - Use if you rely on browser scaling to render an image in proportion to text size.
 - Use with a z-index in order to stretch a background image to fill its entire window.
 - Using instead of background-image can dramatically improve performance of animations

So Or Background- image? - Pt2

Pros for CSS Background Images

- Use CSS background images if the image is not part of the content.
- Use CSS background images when doing image-replacement of text
- Use background-image if you need to improve download times, as with CSS sprites.
- Use background-image if you need for only a portion of the image to be visible
- Use background-image if you need different images for different screen resolutions



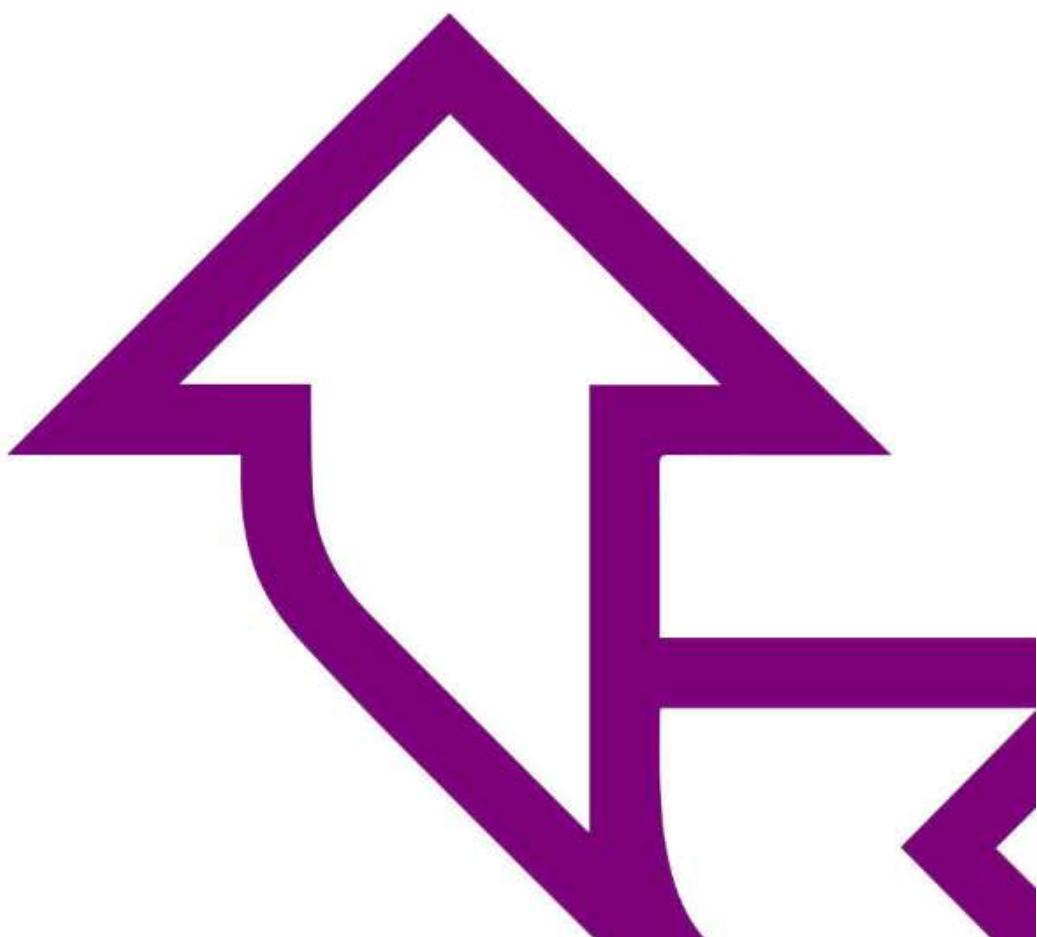


Quick Lab 8 - Images and Backgrounds

- Experiment with the format of images on pages
- Add background images to elements and format how they are displayed



The Box Model



CSS Fundamentals

QA The Box Model

All HTML elements can be considered as boxes

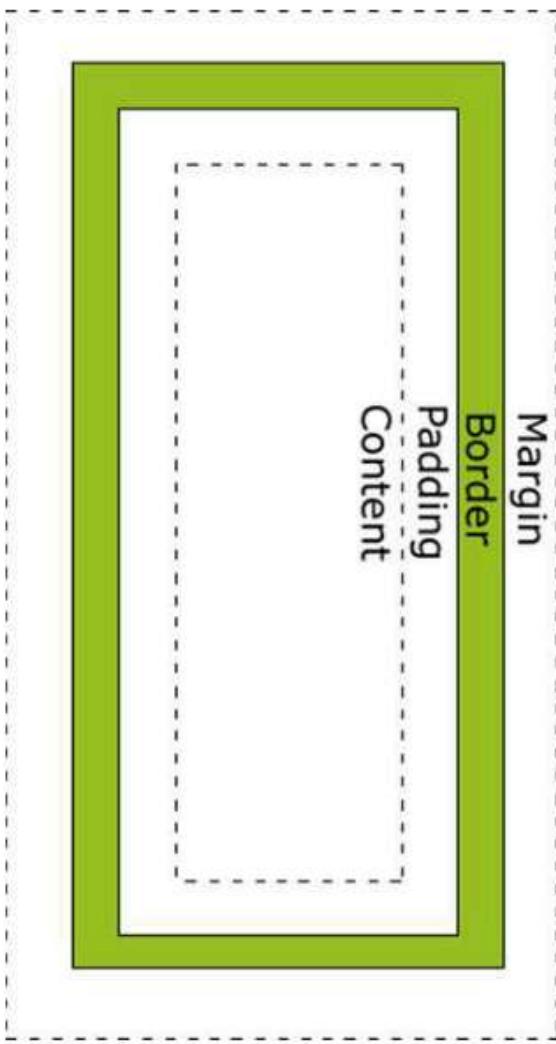
Box Model is used when talking about design and layout

Essentially a box that wraps around HTML elements

Consists of margins, borders, padding and the actual content

Box model allows:

- Placing a border around elements
- Space elements in relation to other elements



QA The Box Model

Margin:

- Clears an area around the border
- Is transparent (no background colour)

Margin

Border:

- Goes around the padding and the content
- Affected by the background colour of the box

Border

Padding

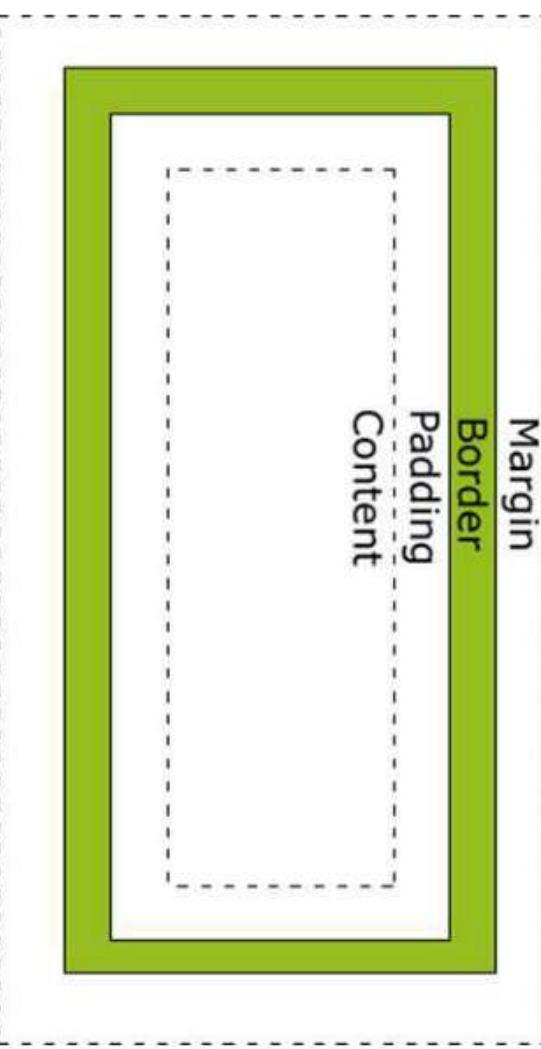
Content

Padding:

- Clears an area around the content
- Affected by the background colour of the box

Content:

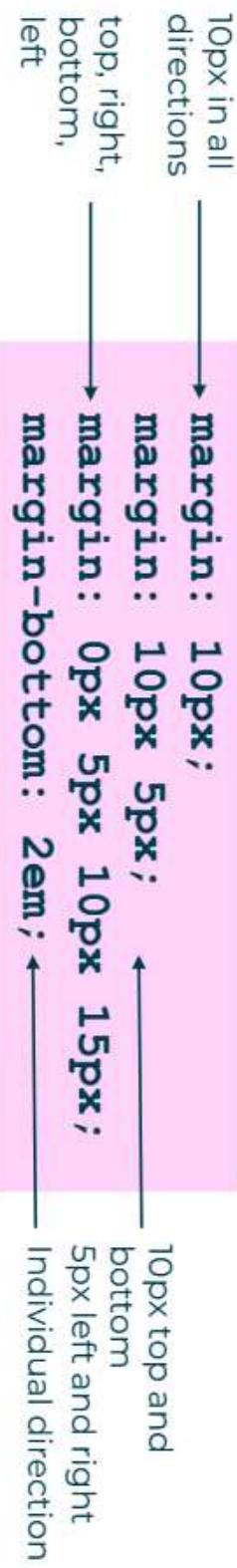
- The content of the box, where text and images appear



QA The Box Model – Settings Properties

All HTML elements have four sides – top, bottom, left and right

- Properties can be set for each dimension or in a compound rule:



Child elements typically have their own block properties

- Can be set independent of the parent
- The inner width of an element (content) available is a remainder of reserved space by parent elements
- Background colours and images can also be set

Q&A Element Width and Height

- When you set the width and height properties of an element with CSS you only set them for the CONTENT area
- To calculate the full size of an element you must add the padding, borders and margin to the width of the content
- What is the TOTAL width of the space the element takes here?

```
width: 250px;  
padding: 10px;  
border: 5px solid gray  
margin: 10px
```

300px

It is 300px

Let's do the maths

- + 250px (content width)
 - + 20px (left and right padding)
 - + 10px (left and right border)
 - + 20px (left and right margin)
-

QA The border-box model

The **broken box model is a familiar tale of woe to most**

- CSS3 includes an attribute called box-sizing
- Set to content-box to get the traditional W3C box model.

```
article { box-sizing: content-box; }
```

- The total width of the element will be:
 - the width set on the element
 - plus the width of the borders and padding.
- If border-box borders and paddings include in the width.

```
article { box-sizing: border-box; }
```

QA Borders

Borders can have the following attributes set:

- border-width: all [top, right, bottom left]
- border-style: all [top and bottom, left and right]
- border-color: top [left, bottom, right]
- Properties can be set individually for all by using shorthand border property

```
div { border: 2px dashed blue; }
```

Can specify border for each side by inserting top, left, bottom or right between border and property to set or use the shorthand:

```
div { border-top-style: double; }  
div { border-left: 5px inset purple; }
```

QA Rounded borders

- Pre-CSS3 had to be achieved through JavaScript or images:

```
border-radius: 30px
```

- Different radius can be added to different corners

```
border-top-left-radius: 50px;  
border-top-right-radius: 30px;  
border-bottom-right-radius: 50px;  
border-bottom-left-radius: 30px;
```

- Shorthand

```
border-radius: 50px 30px 50px 30px;
```

QA Outline

Renders a uniform line for viewers attention

- Rendered on top of an elements rendering box
- Does not influence a box's position or size

```
outline: 3px dashed #3a5c7a;
```

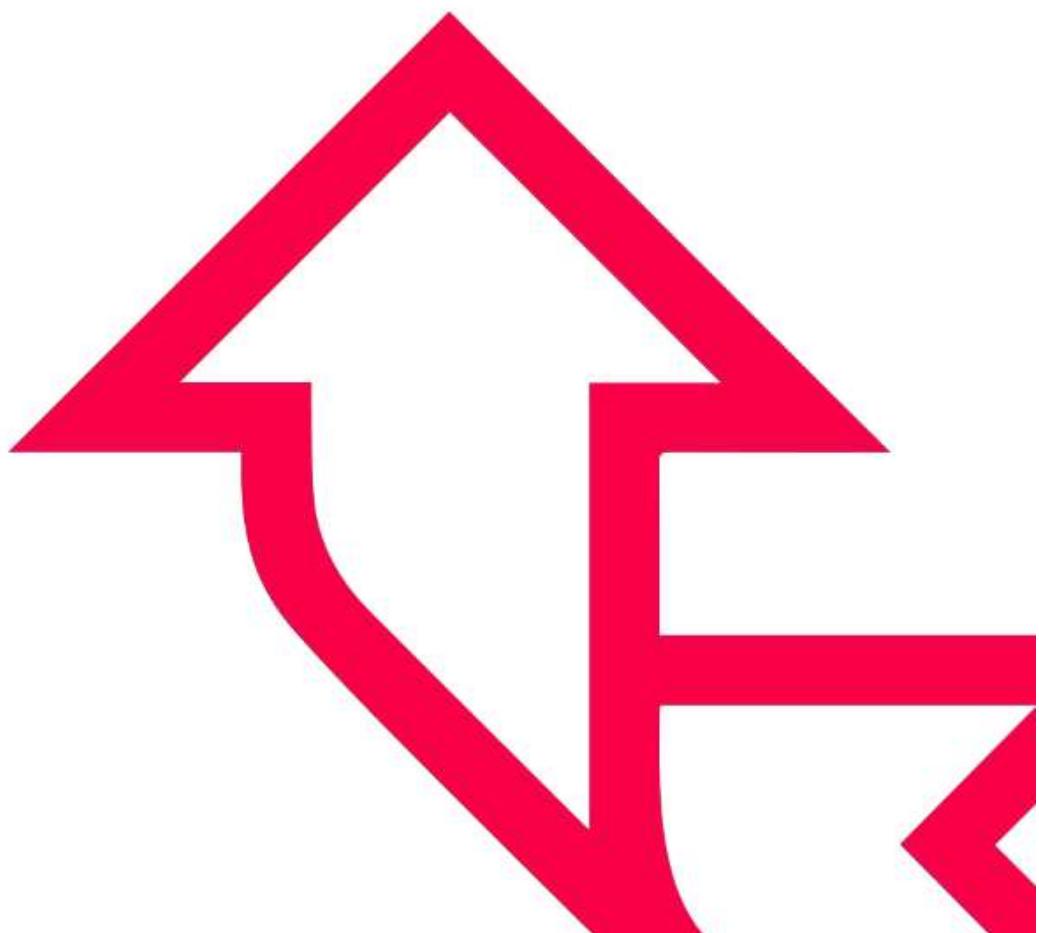
Optional outline-offset property

- Offsets an outline
- Then draws it beyond the border edge.

```
outline-offset: 10px
```



Positioning Elements



CSS Fundamentals

Positioning Elements

Position: relative | static

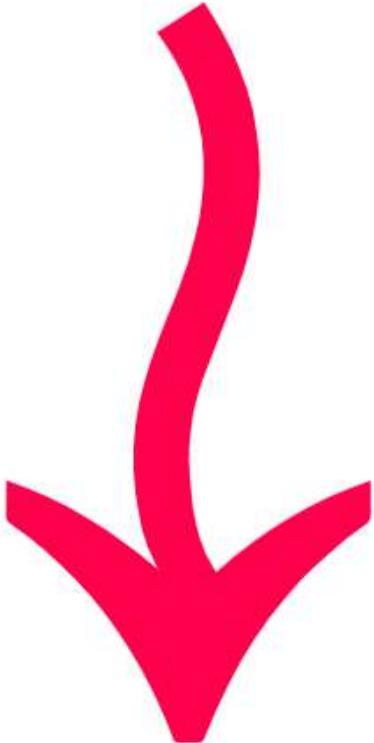
- The content edge of the nearest block-level ancestor

Position: absolute

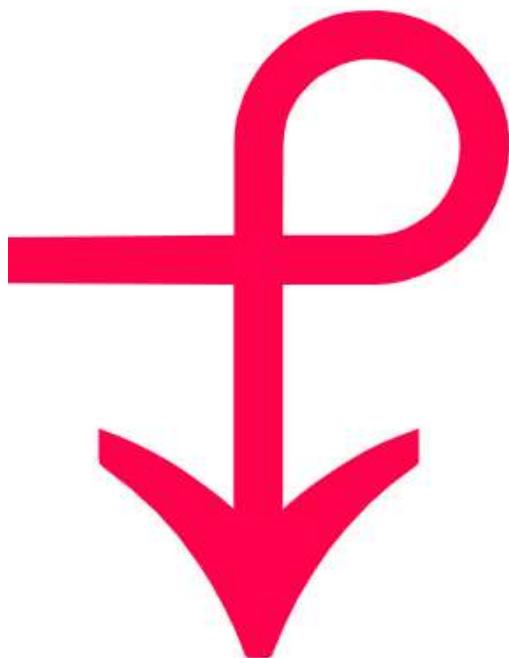
- The nearest positioned ancestor according to
 - The padding edge of the if the ancestor is block-level
 - The content edge of the first/last box if the ancestor is inline

Position: fixed

- The window / printed page



Relative positioning



Relative positioning: offset from default position

- I.e., moved from where it would have been
- Offset not measured from containing block

Next element flows as if the box hadn't been moved

- Relative boxes take up space where they would have been

Moved element has same size as if it hadn't been moved

- Hence specify only one of left/right and top/bottom
 - E.g., if you specify left and right, this could change the width of the element, which is not allowed, hence one of left/right will be ignored

See

- <http://www.w3.org/TR/CSS21/visuren.html#relative-positioning>

Absolute Positioning

Absolute positioning: offset from container's position

- I.e., relative to container not page

Offset measured from

- Block level ancestor: the top, left of ancestor's padding box
 - I.e., outside of padding, inside of border
- Inline ancestor: the top, left of the ancestor's content box
 - I.e., outside of content

See

- <http://www.w3.org/TR/CSS21/visuren.html#position-props>
- <http://www.w3.org/TR/CSS21/visuren.html#absolutely-positioned>

QA Margin - Positive and Negative Values

Giving CSS positive values for padding or margin puts space between element and its reference

```
margin-left: 20px;
```

- Puts 20 pixels between the left margin of the element and its reference - effectively moves the element 20 pixels to the right

Giving CSS negative values for padding or margin moves the element towards its reference

```
margin-left: -20px;
```

- Effectively moves the element 20 pixels to the left

QA

Float and clear

Float will move an element and flow text around it

- Treats the element as a block element and moves it left / right
- Rest of the page flows around the floated element
- The available box is shrunk by the amount the floats take up

Clear will move an element to after the float

- Adds clearance to the top margin to move it clear of the float
- Moves top border edge below the bottom outer edge of the float
- Unless the cleared element is also a float (line up outer edges)

See

<http://www.w3.org/TR/CSS21/visuren.html#propdef-float>
<http://www.w3.org/TR/CSS21/visuren.html#propdef-clear>

Q4

OVERFLOW, MIN & MAX DIMENSIONS

The width and height of an object can be constrained

- With **min-height/min-width** and **max-height/max-width**
- Once set, an element will never grow/shrink beyond these values

The element is now smaller than the content it displays

- What happens to this content can be controlled with the **overflow**
- Can be set to:
 - auto
 - visible
 - hidden
- CSS3 allows overflow control on a specific axis **overflow-x/y**
- In CSS3 we also have the **hidden** property

QA

Controlling how an element displays



Elements are primarily set to be block or inline as their display type

- This behaviour can be changed in CSS
- By modifying the display attribute
- By setting an element property `display:none` it is hidden

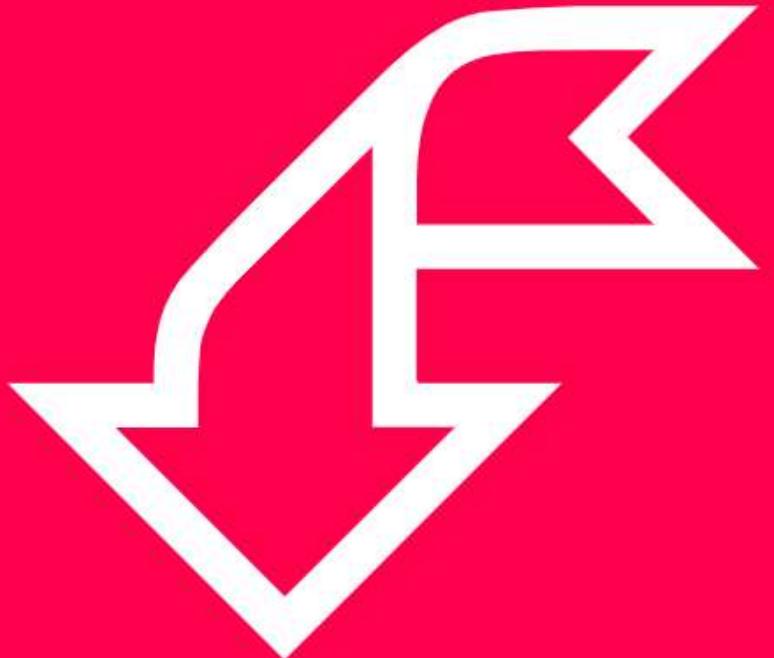
• The element is then removed from the flow

- Can be accomplished with a hidden attribute in HTML5
- Alternatively there is the visible property

• Does not remove the element from the document flow

Elements can also be switched between inline and block display

- Useful for advanced layout

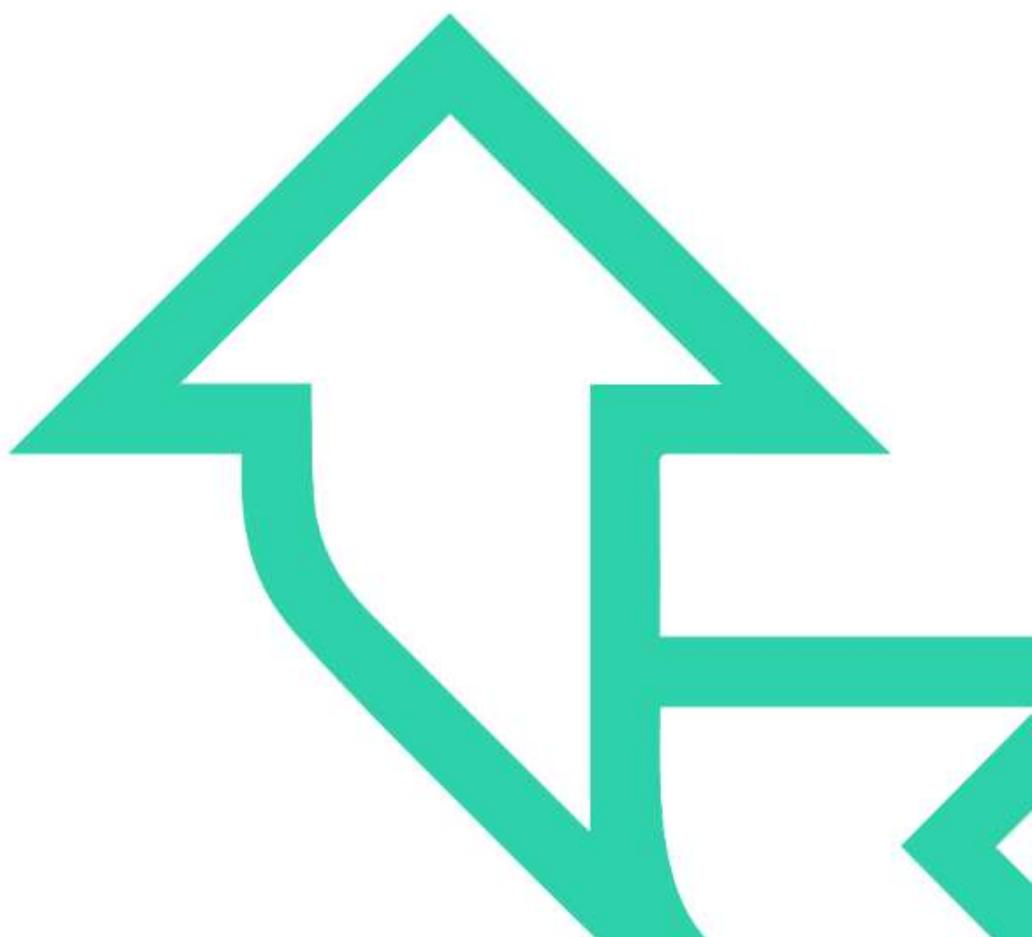


Quick Lab 9 – Positioning Elements

Use positioning and styling techniques to layout a page to a given design



Lists



CSS Fundamentals

QA List Styles

- Set on the enclosing list tag - either or
- Can be:

Property	Description	Examples of Possible Values
list-style-image	Sets an image as the list-item marker	url("images/bullet.svg"), none
list-style-position	Sets the position of the list-item markers	inside, outside
list-style-type	Sets the type of the list-item marker	disc, circle, square, decimal, georgian, none, inherit, initial
list-style	Shorthand that sets all properties in one declaration	lower-roman url("images/bullet.svg") outside

QA Lists with Custom counters

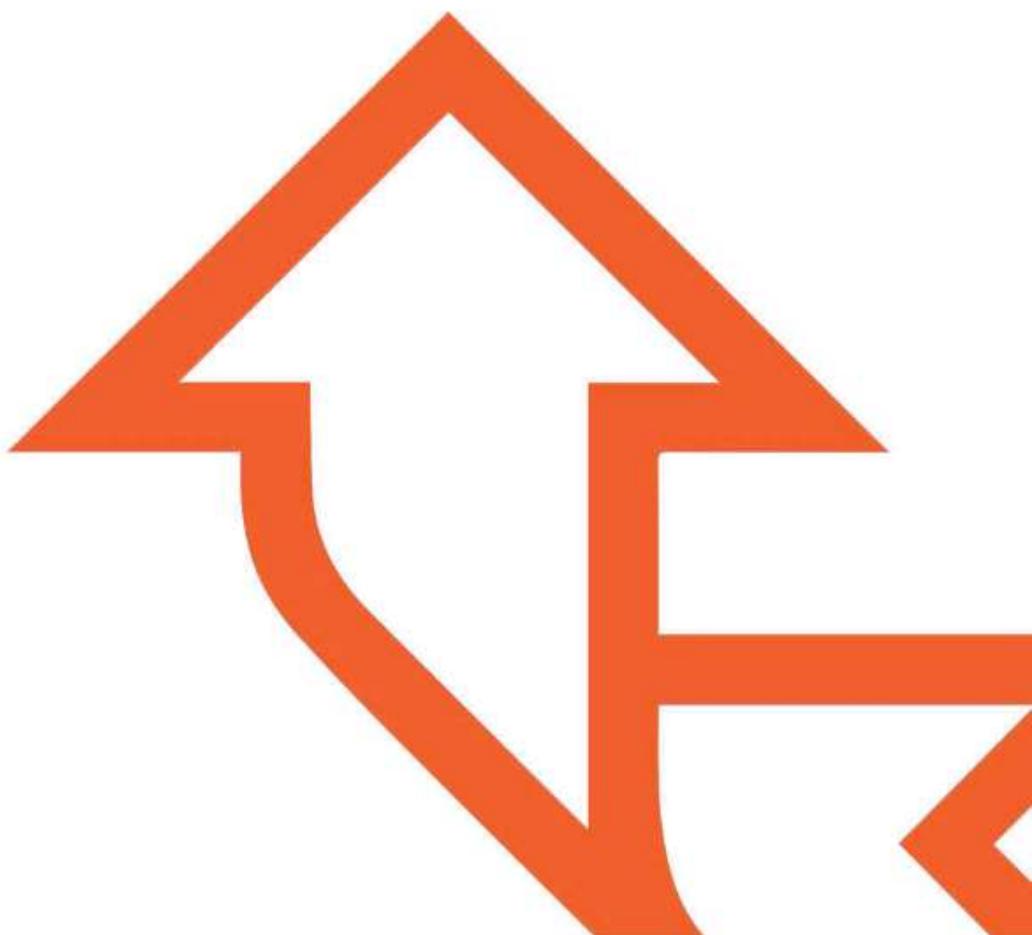
Useful for making outline lists

- New instance of counter automatically created in child elements
- Uses CSS function **counters()** – can insert separating text in between different levels

```
ol {  
    counter-reset: section; /* Creates new instance of section  
    counter for each new ol element */  
    list-style-type: none;  
}  
  
li::before {  
    counter-increment: section; /* Increments only this instance */  
    content: counters(section, ". ") " "; /* Combines values of all  
    instances of section  
    counter, separated by a . */  
}
```



Tables



CSS Fundamentals

TABLES

Tables can be controlled with CSS with a series of properties

- The first is the **table-layout** which has two options that describe how to precisely divide up column widths

- **auto**
- **fixed**

Inter-cell padding is set with the **border-spacing** attribute - Equal in all directions

Every table cell defined by a <td> or <th> tag has four borders

- These butt up against each other so setting a 1px border with no **border-spacing** the gap is doubled
- This can be controlled with the **border-collapse** property
 - **separate** – default borderers butt
 - **collapse** – borders overlap

QA Table Properties

- Set on the enclosing <table> tag
- Can be:

Property	Description	Examples of Possible Values
caption-side	Puts content of table's <caption> on specified side	top, bottom
empty-cells	Sets how browser should render borders and backgrounds around table cells that have no visible content	show, hide
vertical-align	Sets vertical alignment of an inline or table-cell box	baseline, sub, super, text-top, text-bottom, middle, top, bottom

QA Table Formatting and Interactivity

The pseudo-class :hover can be applied to <tr>

- Will change the style of the row dependent on the format set

```
tr:hover { background-color: hotpink; }
```

Striped tables can be created by using the nth-child pseudo-selector and odd or even

```
tr:nth-child(odd) { background-color: palevioletred; }
```

Responsive tables can be created to display a horizontal scroll bar if the screen size is too small to display the whole content of the table

- Add a container around the table and use overflow-x: auto

```
<div style=overflow-x : auto>
<table>...table content</table>
</div>
```

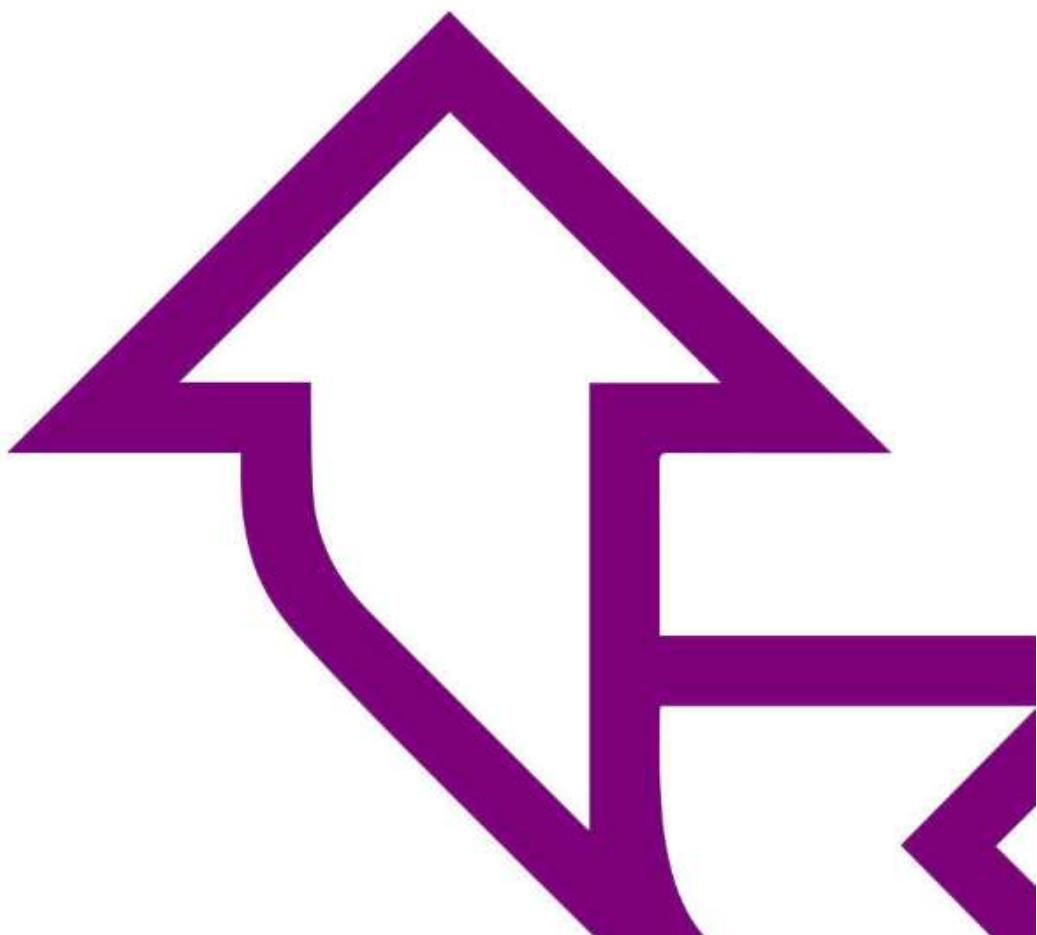


Quick Lab 10 – Tables with CSS

Add styling to a table to make it more readable and interactive with hoverable rows



Animations, Transitions, and Transformations



Q A @keyframe at-rule

Defines and controls the immediate steps in a CSS animation sequence

- Defines styles for keyframes along animation sequence – name used in **animation-name**
- Gives more control over immediate steps than transitions

```
@keyframes slidein {  
  from {  
    margin-left: 100%;  
    width: 300%;  
  }  
  to {  
    margin-left: 0%;  
    width: 100%;  
}
```

QA Animation Properties (1)

Allows animation of css properties over time using keyframes and properties below:

Property	Description	Examples of Possible Values
animation-name	Specifies one or more animations that should be applied to an element (defined by @keyframes)	none, slide, bounce
animation-duration	Sets length of time that animation takes to complete one cycle	0s, 750ms
animation-timing-function	Sets how animation should progress over duration of cycle	Linear, ease-in-out, steps(5, end)
animation-delay	Sets when animation should start – immediately, in the future or partway through the animation cycle	250ms, -2s

QA Animation Properties (2)

Allows animation of css properties over time using keyframes and properties below:

Property	Description	Examples of Possible Values
animation-iteration-count	Specifies number of times animation should play before stopping	0, 2, 3.2
animation-direction	States whether animation should play forwards, backwards, or alternate	normal, reverse, alternate, alternate-reverse
animation-fill-mode	Sets how animation should apply styles to target before and after	none, forwards, backwards, both
animation-play-state	Sets whether animation is playing or paused	paused, running

QA Animation Properties (3)

Shorthand animation can be used to specify all properties:

Order:

- duration | timing-function | delay | iteration-count | direction | fill-mode | play-state | name

animation: 3s ease-in 1s 2 reverse both paused slidein

- Would run an animation that lasted for 3 seconds after a delay of 1 second, easing in, running twice in reverse starting paused, and using the **slidein** definition

PLAY

An Overview of Transitions

CSS3 allows you to define transitions for property changes

- E.g., when a user hovers over an element, change its size to XXX over a period of YYY
- The transition kicks in automatically on the property value changes

To define a simple transition in a CSS rule:

- Set the **transition** property
- Specify the property to vary and the duration of the transition

```
someCssRule {  
    ...  
    transition: aProperty duration;  
}
```

Note:

- You must use vendor-specific extensions for some browser versions

QA Transition Properties (I)

Enables definition of transition between 2 states of an element

- States may be defined using pseudo-classes or dynamically set using JavaScript

Property	Description	Examples of Possible Values
transition-property	Defines which CSS property (or properties) for transition	margin-right, width, height
transition-duration	Defines number of seconds or milliseconds a transition should take	500ms, 2s
transition-timing-function	Sets timing function to set intermediate values during transition	linear, ease-in, steps(6, end), cubic-Bezier(1, 1, 1, 1)
transition-delay	Sets amount of time to wait before starting the transition	250ms, 1s

QA Transition Properties (2)

Shorthand transition can be used to specify all properties:

Order:

- property | duration | timing-function | delay

transition: margin-right 2s ease-in-out .5s

- Would run a transition that lasted 2 seconds after a delay of 0.5 seconds, easing in then out on the margin-right property of the element it has been applied to

QA 2D Transformations

CSS3 supports 2D and 3D transforms

- Enables elements rendered by CSS to be transformed in space

To define a transformation in a CSS rule:

- Set the **transform** property
- Optionally set the **transform-origin** property

```
someCssRule {  
    ...  
    transform: transformation-function(s);  
    transform-origin: horizPosition vertPosition;  
}
```

QA Transform Functions (I)

Different Transform functions – for Rotation:

Function	Description
<code>rotate()</code>	Rotates element around fixed point on 2D plane
<code>rotate3d()</code>	Rotates element around fixed axis in 3D space
<code>rotateX()</code>	Rotates element around horizontal axis
<code>rotateY()</code>	Rotates element around vertical axis
<code>rotateZ()</code>	Rotates element around z-axis

Different Transform functions – for Skewing

Function	Description
<code>skew()</code>	Skews element on 2D plane
<code>skewX()</code>	Skews element in horizontal direction
<code>skewY()</code>	Skews element in vertical direction

QA Transform Functions (2)

Different Transform functions – for Scaling:

Function	Description
scale()	Scales element up or down 2D plane
scale3d()	Scales element up or down in 3D space
scaleX()	Scales element up or down horizontally
scaleY()	Scales element up or down vertically
scaleZ()	Scales element up or down along z-axis

Different Transform functions – for Matrix Transformations

Function	Description
matrix()	Describes a homogeneous 2D transformation matrix
matrix3d()	Describes a 3D transformation as a 4x4 homogeneous matrix

QA Transform Functions (3)

Different Transform functions – for Translation:

Function	Description
<code>translate()</code>	Translates element on 2D plane
<code>translate3d()</code>	Translates element in 3D space
<code>translateX()</code>	Translates element horizontally
<code>translateY()</code>	Translates element vertically
<code>translateZ()</code>	Translates element along z-axis

Different Transform functions – for Perspective

Function	Description
<code>perspective()</code>	Sets distance between the user and the z=0 plane

QA Translations

To translate an element, use one of these CSS functions:

`translate(tx, [ty])`

`translateX(tx)`

`translateY(ty)`

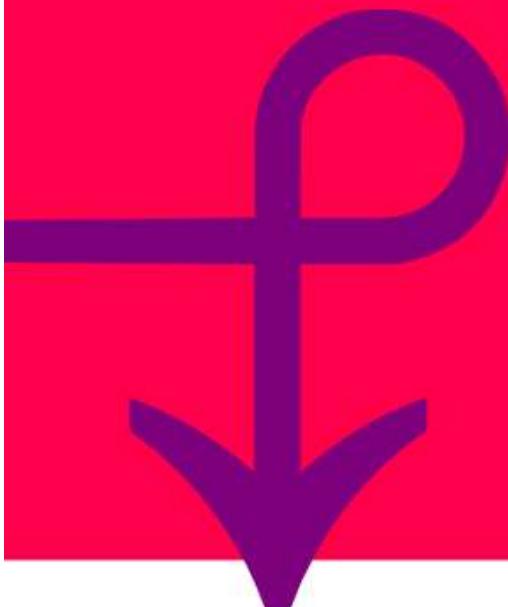
Example:

```
someCssRule {  
    transform: translate(400px, 20px);  
}
```

QA

Learning Objectives

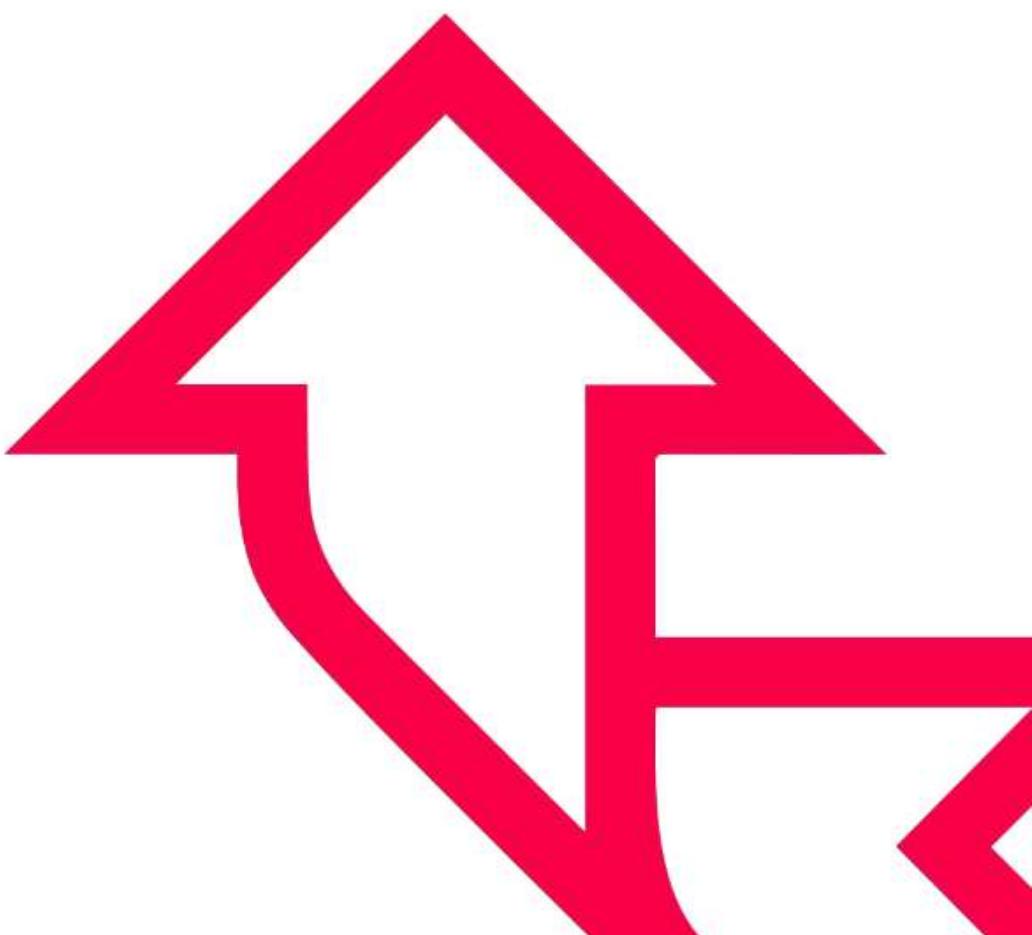
- Understand what CSS is
- Understand how CSS can be applied to web pages
- Understand the syntax of writing CSS rules
- Be able to select elements to apply CSS to
- Be able to work with text, colours, and images
- Be able to work with the box model and position elements
- Be able to style lists and tables
- Be able to add CSS animations, transforms and transitions to elements





Web Fundamentals

RESPONSIVE WEB DESIGN (RWD)



Learning Objectives

- Understand why 'Mobile First' and 'Responsive Web Design'
- Be able to apply RWD principles
- Be able to use media queries
- Understand and implement grid systems
- Understand flexbox
- Be able to create and use responsive images

QA

Responsive Web Design

- Mobile First and Responsive Design
- Viewports
- Media Queries
- Grids
- Flex Box
- Responsive Images



HTML

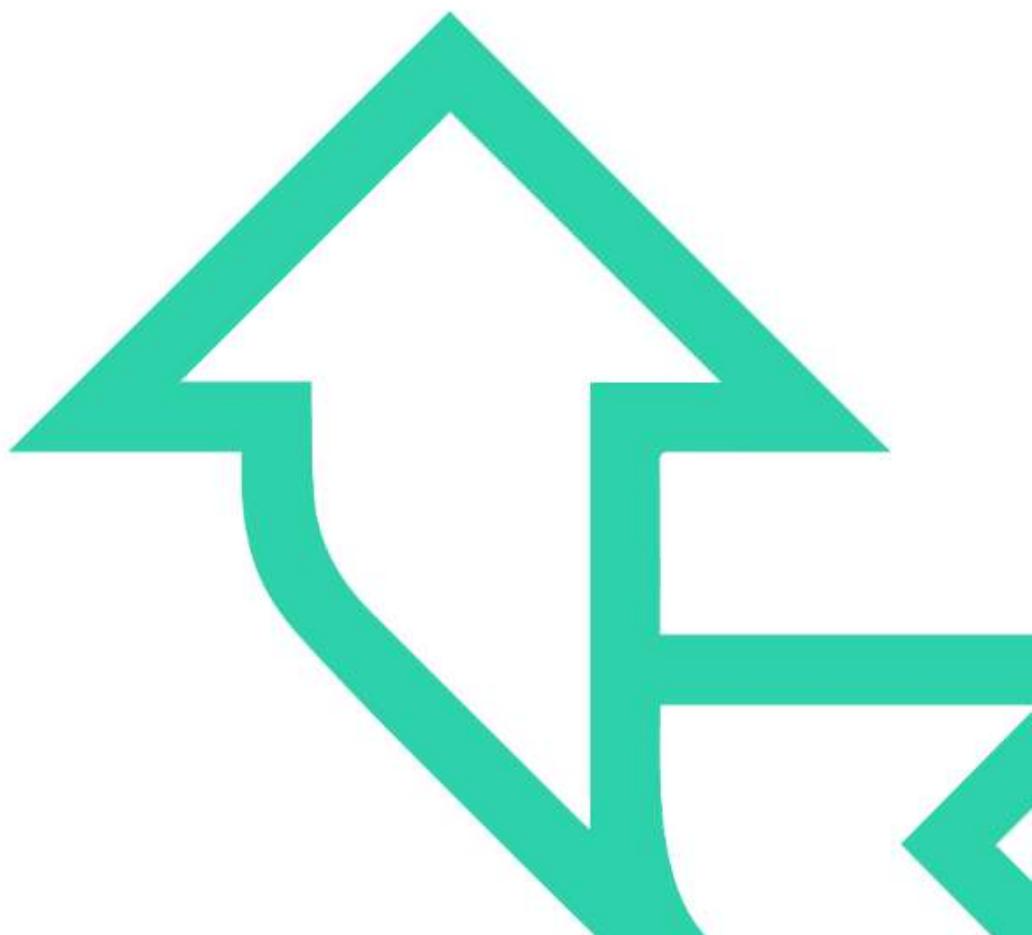


CSS



Mobile First and Responsive Web Design

Responsive Web Design



QA Mobile First

UI design idea first coined by Luke Wroblewski in 2011

- Encourages UI to be designed for a mobile device before a large screen
- Helps to identify what most important content is
- Ensures that this content is displayed prominently on a mobile device

Why 'Mobile First'?

- Prepares you for the explosive growth and new opportunities emerging on mobile today
- Forces you to focus and prioritise your products by embracing the constraints inherent in mobile design
- Allows you to deliver innovative experiences by building on new capabilities native to mobile devices and modes of use

Taken from **Mobile First**, L Wroblewski, P1

Responsive Web Design

Pioneered by Ethan Marcotte in 2010
Ensures that web pages render well regardless
of device/screen/window size

- Adapts the layout by using fluid, proportion-based grids, flexible images and media queries
- Fluid grid concept requires sizing to be in relative units (percentages rather than pixels or points)
- Flexible images sized in relative units
- Media queries allow different rules to be applied usually dependent on width of screen available

Only uses HTML and CSS



QA Content First - Manipulate

RWD is more than changing layouts

Media queries do not just change layouts

- Allow designers to manipulate content dependent on viewport size and device type

Make sure main content is visible on phone when user views homepage

- Hamburger navigation, move ads down, etc

General Rules for Manipulating Content

- Reorder
- Reposition
- Replace
- Remove (last resort)

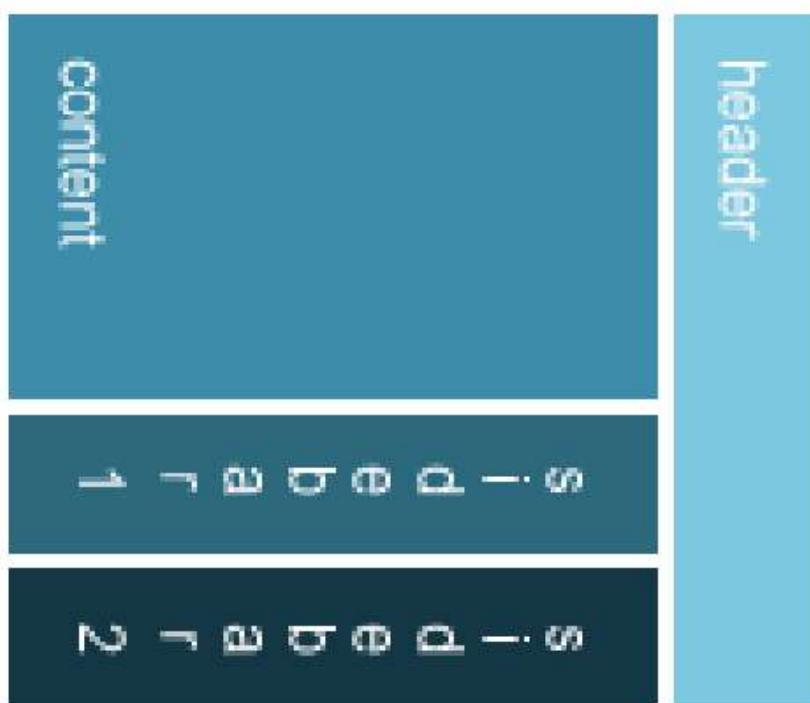
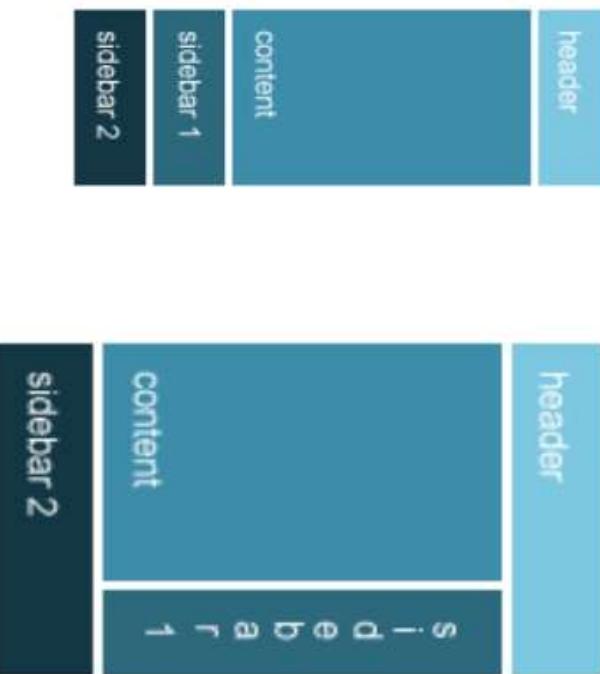
Changing content is common

- Lower resolution videos for small screens, etc

Removing content completely is last resort

QA Responsive is not just smaller text

- Responsive development is about re-envisioning content



- Relative sizing can make content too big on mobile devices and too small on desktop sized screens and above

QA The Viewport

User's visible area of the page

Varies from device to device

- Small on a mobile phone, larger on a full size screen

Setting the viewport

- Done in a meta tag with the head of the HTML page
- Has to be included for responsiveness to work

```
<meta name="viewport" content="width=device-width, initial-scale=1.0">
```

Size content to the viewport

- Users should only scroll vertically – never horizontally
- Content should not rely on a particular viewport size
- Media queries should be used to apply different styles on different sized screens

QA max-width

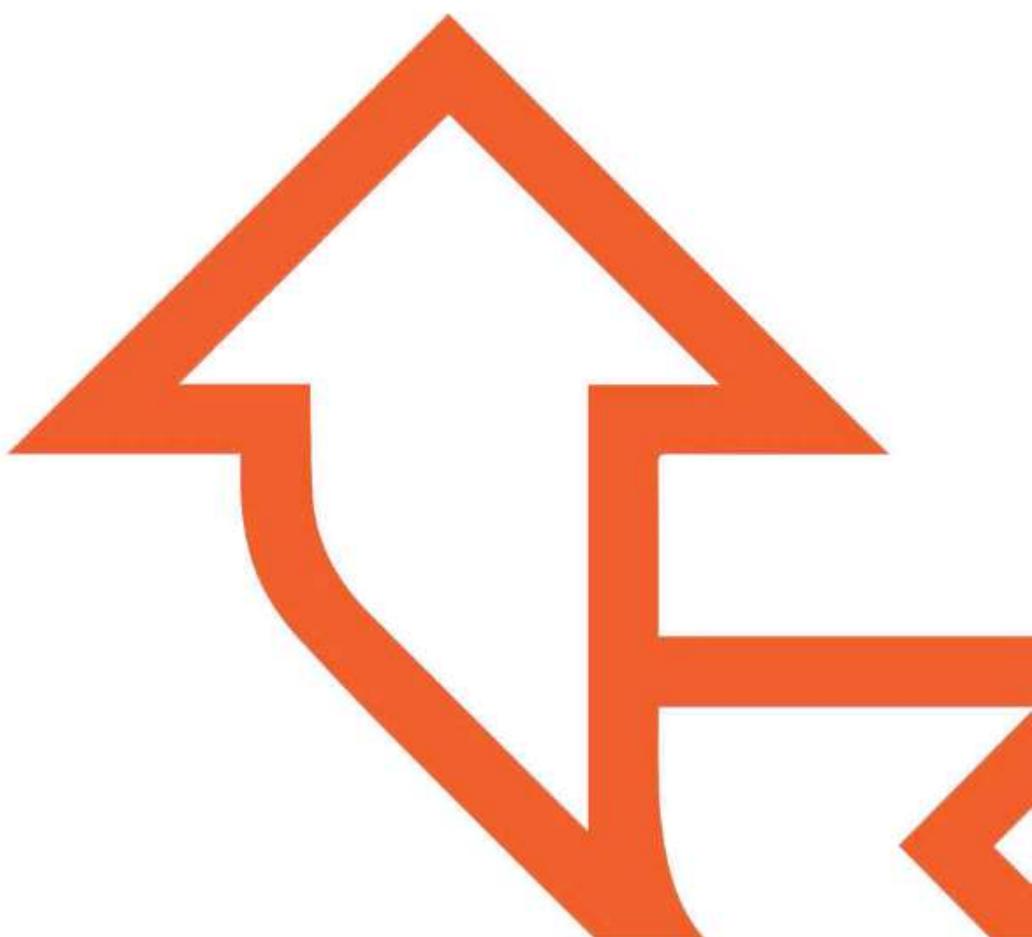
- Will solve most problems with images
- Sets the maximum width of a given element (if % it is that of the parent element)
- Elements will not appear wider than the maximum specified for the element

```
img {  
  max-width: 100%  
}  
  
img.biggest {  
  max-width: 300px  
}
```



Media Queries

Responsive Web Design



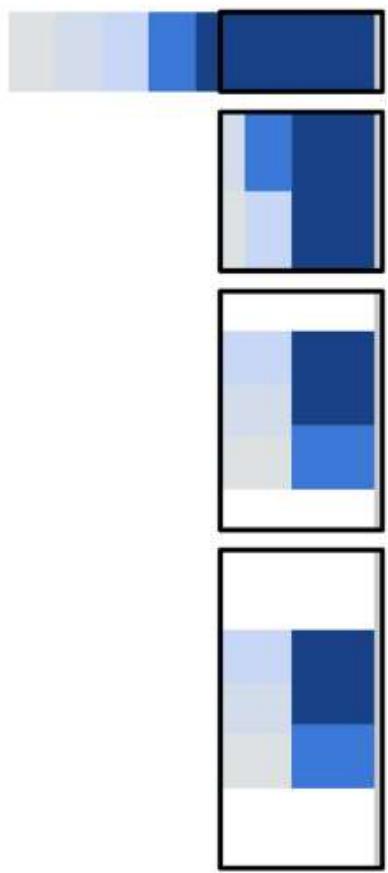
QA Media Queries

Modify the layout of your site based on different criteria

Same layout not necessarily appropriate for every device

Might have

- Single column on phone
- Two columns on tablet
- Three columns on desktop
- etc



```
@media (max-width: 480px) {  
  /* layout for phones */  
}  
  
@media (max-width: 720px) {  
  /* layout for tablets */  
}
```

QA Match Breakpoints to Content

- Devices are constantly changing
 - Viewports getting bigger and smaller
 - Pixel density, pixel shape display quality
- Designers should not be forced to make change every time new viewport appears
- Follow rules opposite

General Rules for Creating Breakpoints in Content

- Start small
- Add major breakpoints
- Add minor breakpoints if necessary
- Optimise for reading: 70-80 characters per line

QA Mobile First Media Queries

- Look at the minimum width of a device to display content in particular way instead
- No fixed rule about whether to include media queries inline or use a separate file
- Might want to consider using ems or rem

```
/* small by default */  
@media (min-width: 480px) {  
    /* medium */  
}  
  
@media (min-width: 720px) {  
    /* large */  
}
```



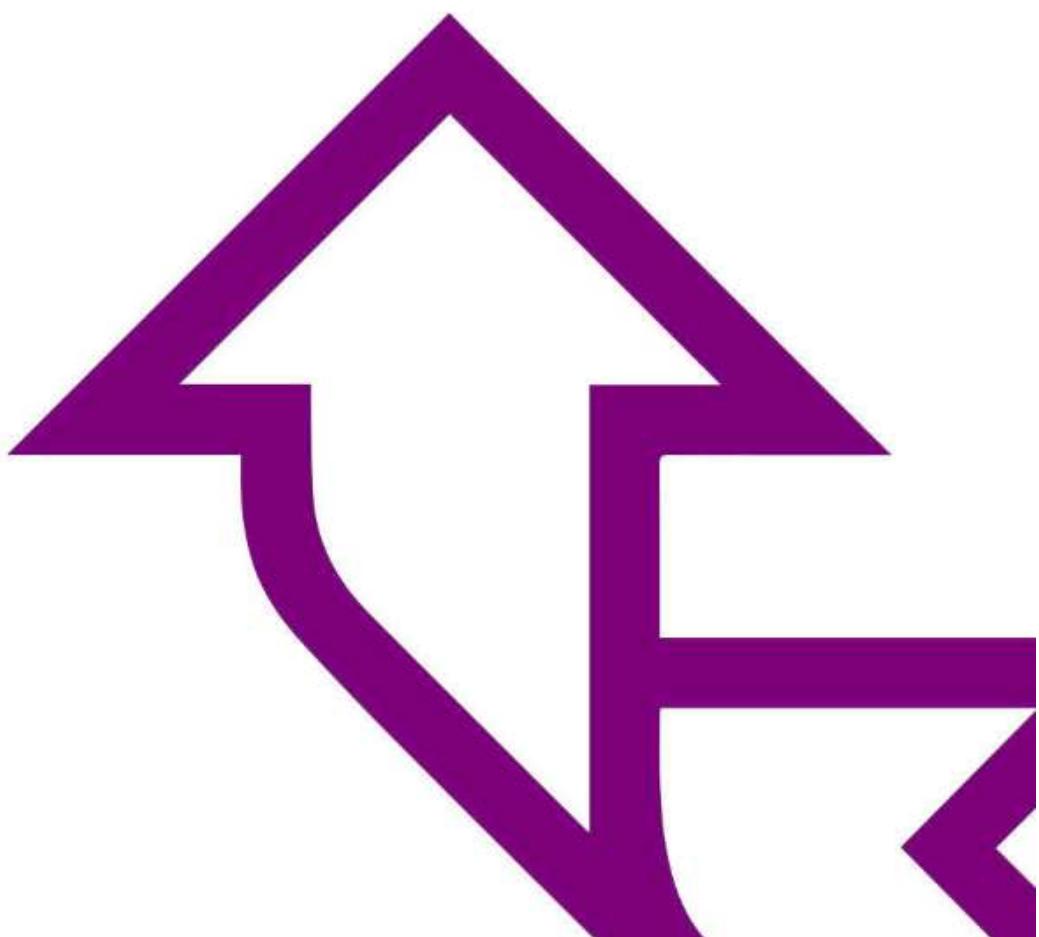
Quick Lab 11 – Media Queries

Create some media queries to make a page adapt to the width available.



Grids

Responsive Web Design



QA Changing Layouts – Grid View

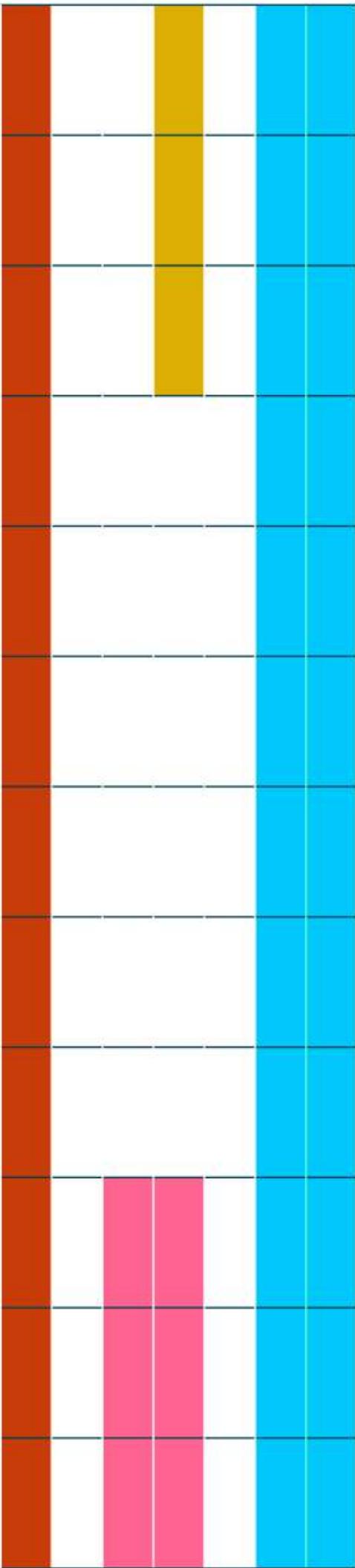
Modern websites often based on a grid-view

Pages are divided into (usually 12) equal columns

1	2	3	4	5	6	7	8	9	10	11	12
---	---	---	---	---	---	---	---	---	----	----	----

Helpful when designing pages for different devices as easier to place elements on page

- Total width is 100% and will shrink and expand as the browser window is resized



QA Responsive Grids

All HTML elements have to have box-sizing set to border-box

- Ensures padding and border are included in total width and height of elements

```
* { box-sizing: border-box; }
```

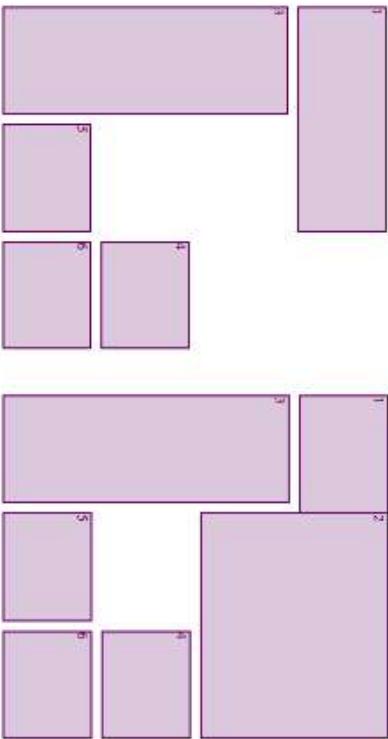
Can create simple responsive page with 2 columns

```
.left { width: 25%; float: left; }  
.right { width: 75%; float: left; }
```



QA CSS Grid Layout Module

- Offers a grid-based layout system, with rows and columns
- **HTML**
- **CSS**



```
<div class="grid-container">
<div class="one">1</div>
<div class="two">2</div>
<div class="three">3</div>
<div class="four">4</div>
<div class="five">5</div>
<div class="six">6</div>
</div>

.grid-container {
  display: grid;
  grid-template-columns: repeat(3, 1fr);
  grid-gap: 10px;
  grid-auto-rows: minmax(100px, auto);
}
.one { grid-column: 1/3; grid-row: 1 }
.two { grid-column: 2/4; grid-row: 1/3 }
.three { grid-column: 1; grid-row: 2/5 }
.four { grid-column: 3; grid-row: 3 }
.five { grid-column: 2; grid-row: 4 }
.six { grid-column: 3; grid-row: 4 }
```

QA CSS Grid Layout Module

There are 18 different properties that can be set for the CSS Grid Layout module:

column-gap	grid-auto-rows	grid-row-start
gap	grid-column	grid-template
grid	grid-column-start	grid-template-areas
grid-area	grid-column-end	grid-template-columns
grid-auto-columns	grid-row	grid-template-rows
grid-auto-flow	grid-row-end	row-gap

There are 3 functions that can be used:

fit-content()	minmax()	repeat()
---------------	----------	----------

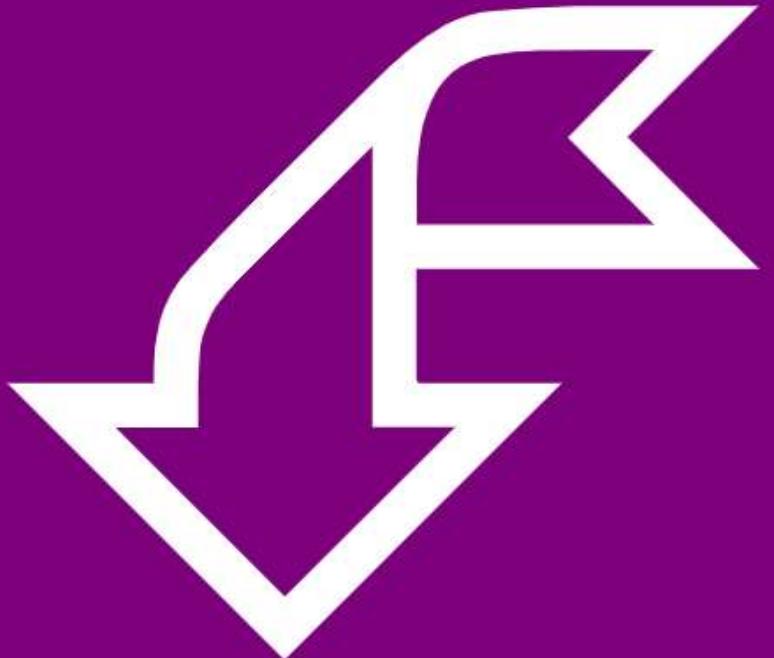
QA

Grid Frameworks

- No need to define all of the classes by hand
- Several Responsive CSS Frameworks exist that already have the CSS classes defined along with many other useful features
 - All developer has to do is add correct classes to the HTML

Examples are Bootstrap and Foundation





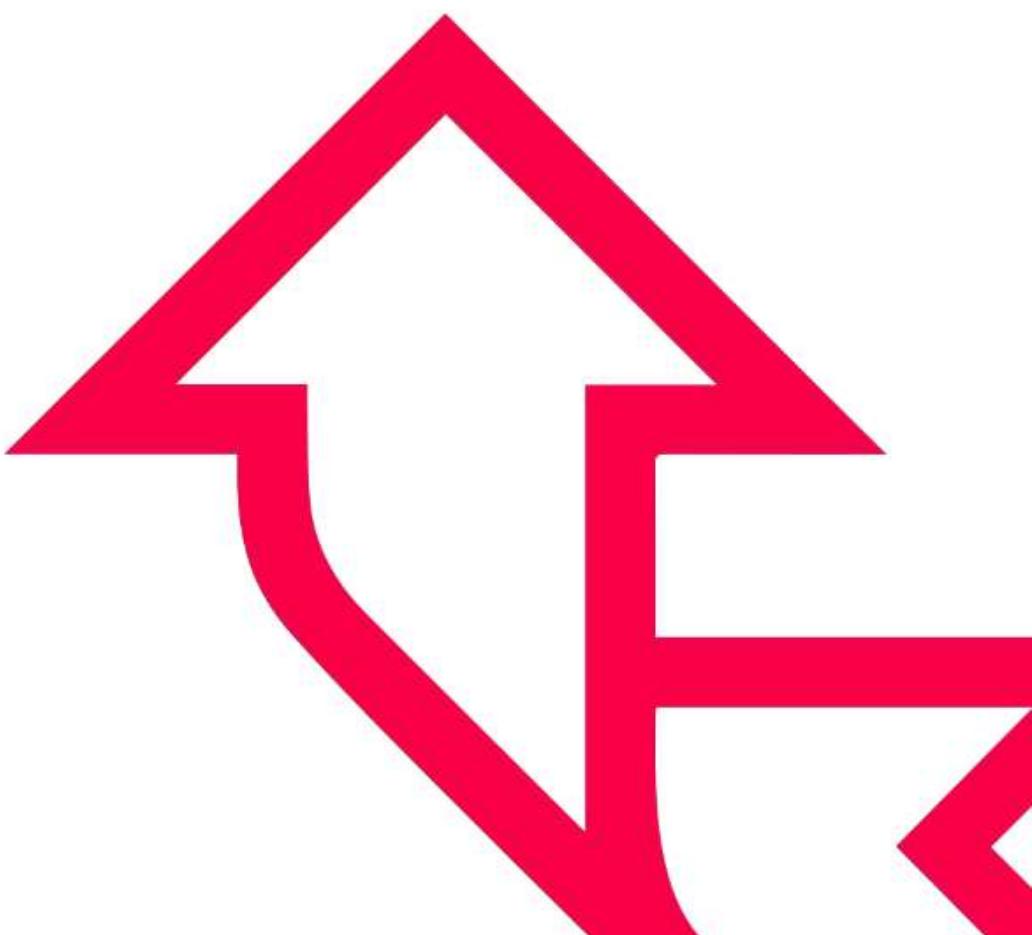
Quick Lab 12 - Grids

Add CSS classes to HTML elements to utilise a provided CSS grid system.



Flex Box

Responsive Web Design



QA Flex Box

Enables easy design of responsive layout structure without using float or positioning

Requires a flex container

- Element becomes flex container by setting display property to flex

```
.flex-container { display: flex; }
```

QA Flex Container Properties

- Flex containers can have the following properties:

Property	Description	Common Values
flex	Specifies how flex item will grow or shrink to fit the container – shorthand to set flex-grow , flex-shrink and flex-basis	auto, initial, none, <+ve num>
flex-grow	Specifies how much of available space should be assigned to the item	<+ve num>
flex-shrink	Specifies how items will shrink to fit container when default size is larger than flex container	<+ve num>
flex-basis	Specifies the initial main size of a flex item (determines size of content-box unless box-sizing is defined)	<width> (% or units), auto, fill, max-content, min-content, fit-content, content, unset

QA Flex Container Properties

- Flex containers can have the following properties:

Property	Description	Common Values
<code>flex-direction</code>	Defines direction the container stacks the flex items	<code>row</code> , <code>row-reverse</code> , <code>column</code> , <code>column-reverse</code>
<code>flex-wrap</code>	Specifies if the flex items will wrap if necessary	<code>nowrap</code> , <code>wrap</code> , <code>wrap-reverse</code>
<code>flex-flow</code>	Shorthand property to set <code>flex-direction</code> and <code>flex-wrap</code>	As above
<code>justify-content</code>	Defines how browser distributes space between and around content along main axis of container	<code>center</code> , <code>end</code> , <code>flex-end</code> , <code>flex-start</code> , <code>left</code> , <code>right</code> , <code>start</code> , <code>norm</code> , <code>space-between</code> , <code>space-around</code> , <code>space-evenly</code> , <code>stretch</code> , <code>safe center</code> , <code>unsafe center</code> , <code>unset</code>

QA Flex Container Properties

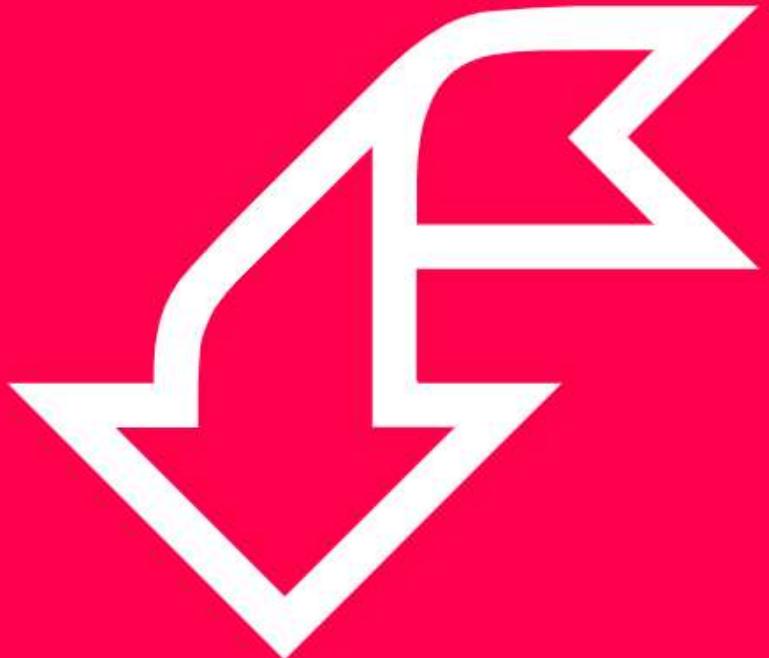
→ Flex containers can have the following properties:

Property	Description	Common Values
<code>align-items</code>	Sets <code>align-self</code> value on all direct children as a group	<code>normal, stretch, baseline, self-start, self-end, flex-start, flex-end, end, center, start</code>
<code>align-content</code>	Specifies how browser distributes space between and around content items along cross axis of their container	<code>As above</code>
<code>align-self</code>	Aligns flex items of current flex line overriding <code>align-items</code> – ignored if any item's cross-axis margin is <code>auto</code>	<code>auto, as above</code>
<code>order</code>	Specifies order used to lay out flex or grid item in flex or grid container	<code><integer></code>

QA Flex Container Properties

→ Flex containers can have the following properties:

Property	Description	Common Values
place-content	Shorthand property to set align-content and justify-content Note: If second value is not provided, first value is used for both Note: If invalid value is used in declaration, whole declaration is invalid	As for align-content and justify-content



Quick Lab 13 – Flex Box

Create a flex-container and experiment with flexbox attributes and settings.



Responsive Images

Responsive Web Design



Q A What are Responsive Images

A method for providing the browser with multiple image sources depending on display density, size of the image element in the page, or any number of other factors.

-- Jason Grigsby

Responsive images refer to one or more methods for providing browsers with the correct image based on attributes of the user's device

- Such as display density, image and page size, or any number of factors.

General principles for performance

- Avoid images wherever possible
- Use vector formats where possible: SVG or icon fonts
- Use the lowest possible resolution and quality
- Use the right format for the image type: WebP, PNG, JPEG

QA The <picture> and <source> elements

<picture> and <source> elements enable us to provide alternative sources for the same resource

- Holds two different tags: one or more <source> tags and one tag
- <source> element has the following attributes
 - **srcset** (required) - defines the URL of the image to show
 - **media** - accepts any valid media query that would normally be defined in a CSS
 - **sizes** - defines a single width descriptor
 - **type** - defines the MIME type

```
<picture>
  <source media="(min-width: 1024px)" srcset="kitten-large.png">
  <source media="(min-width: 667px)" srcset="kitten-medium.png">
  
</picture>
```

QA The width descriptor

Sizes attribute

- Tells the browser the size or sizes of the element the **srcset** is attached to so that the browser can use the appropriate image
- **sizes="50vw"**: telling the browser that the image will be displayed at 50% of the viewport width
- **w** unit : width of each image in pixels, enabling the browser to choose the right image to retrieve, depending on the screen pixel density and the viewport size

```

```

Q& Combining Media Queries and `srcset`

Combining media queries and `srcset` to specify images for different viewports

- Also providing different images for different pixel densities
- Tools like responsivebreakpoints.com will generate the images and the corresponding code for you so you don't have to do

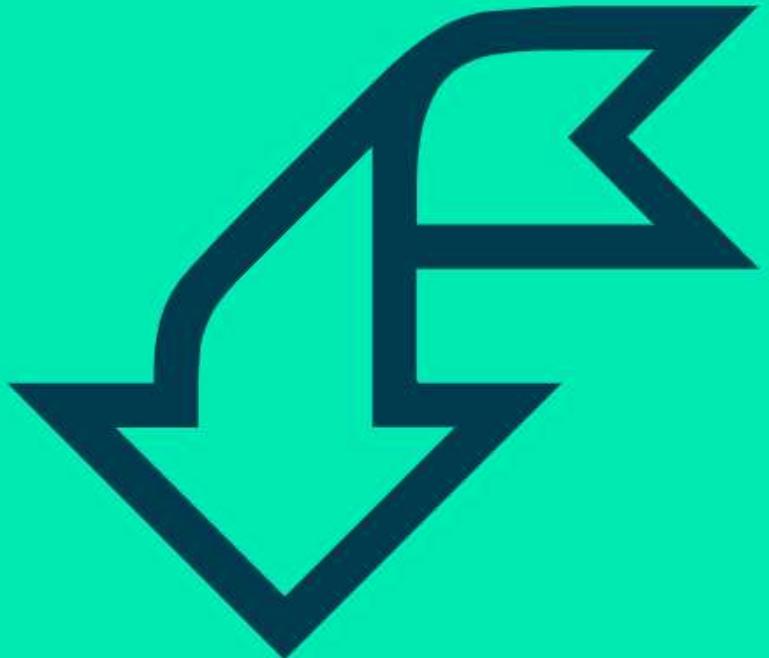
```
<picture>

  <source media="(min-width: 1024px)"
        srcset="kitten-large.png 1x, kitten-large_2x.png 2x">

  <source media="(min-width: 667px)"
        srcset="kitten-medium.png 1x, kitten-medium_2x.png 2x">

</picture>
```



Quick Lab 14 - Responsive Images

Add images to HTML that respond to the device/screen that is being used to view them.

Learning Objectives

- Understand why mobile first and responsive web design
- Be able to apply responsive design principles
- Be able to use media queries
- Understand and implement grid systems
- Understand flexbox
- Be able to create and use responsive images

Hackathon – QA Cinemas

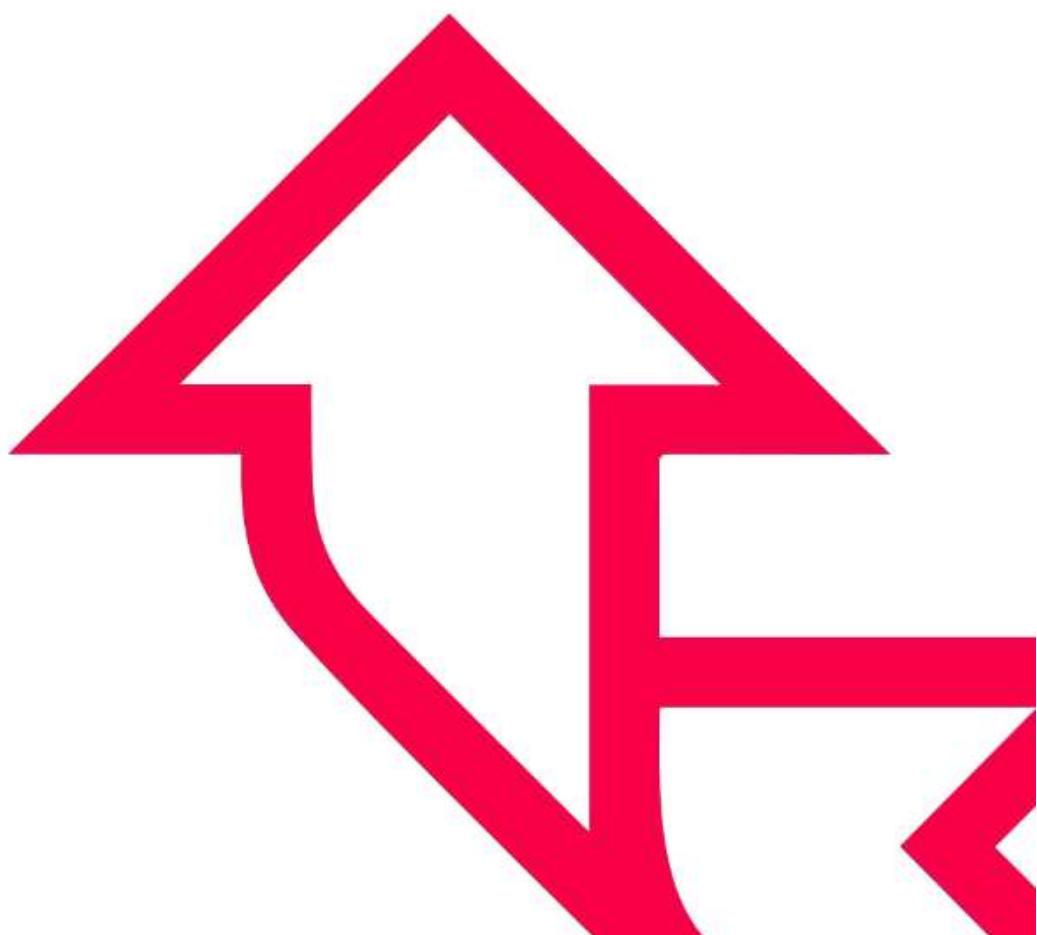
You have a brief for a website to be set up using the skills, knowledge and understanding of HTML, CSS and RWD gained on the course

- The brief has a number of user stories to tackle
- You are expected to:
 - Have a planning meeting to decide how the work will be distributed
 - Create the website
 - Have a review meeting to demonstrate the website to your trainer
 - Hold a retrospective to review what has been learnt as part of the process



Web Fundamentals

Bootstrap Framework



QA

Learning Objectives

- Understand how frameworks make web design easier
- Be able install and use Bootstrap
- Be able to import Icons for use in websites
- Use pre-built bootstrap features in websites

Bootstrap

- Most popular framework in the world – Of all websites queried, Bootstrap is used on 80% of them
- Very simple to install and use via CDN (Content Delivery Network)
- Extensive documentation and example collection
- An externally linked CSS Library
- Controls the appearance of a website with classes



QA Importing

Added inside a link tag in your <head> section like any other CSS document

- This is the most common way, removes the overhead of delivery from your own webserver (if you have one)
- Downside is that it relies on the Bootstrap repository to be available at all times
- Can be downloaded and customised
- getbootstrap.com is the place to start
- One is for the CSS Library
- The script tag allows for animations and the such



Include via CDN

When you only need to include Bootstrap's compiled CSS or JS, you can use [jsDelivr](#). See it in action with our simple [quick start](#), or [browse the examples](#) to jumpstart your next project. You can also choose to include Popper and our JS [separately](#).

```
<link href="https://cdn.jsdelivr.net/npm/bootstrap@5.3.0-alpha1/dist/css/bootstrap.min.css" rel="stylesheet" type="text/css">
<script src="https://cdn.jsdelivr.net/npm/bootstrap@5.3.0-alpha1/dist/js/bootstrap.bundle.min.js" type="text/javascript">
```

Q Importing - Icons

Bootstrap and Font Awesome have impressive libraries

- Saves time on the creation of your own imagery
- Imported in the same way
 - Pulled into the page with a simple <i> tag

Bootstrap Icons

Bootstrap Icons

<https://cdn.jsdelivr.net/npm/bootstrap-icons@1.10.2/font/bootstrap-icons.css>

Click to copy

HTML

```
<link rel="stylesheet" href="https://cdn.jsdelivr.net/npm/bootstrap-
```

Click to copy

Version

6.4.0

Asset Type

All

Some files are hidden, click to show all files

<https://cdnjs.cloudflare.com/ajax/libs/font-awesome/6.4.0/css/all.min.css>

840

QA Grid layout

Bootstrap has the default 12 column grid layout

- Very easy to define how many columns wide an element should be e.g.,

```
<div class="col-md-6">  
    Some Stuff  
</div>
```

- Simply means on a medium screen size, this div should span 6 columns (from a left floated position)
- Allows for the webpage to be responsive without the need for media queries

QA Colours

Bootstrap has a standard set of colours available for text, button colour, background colour, and anything else you might want coloured.

```
<p class="text-primary">.text-primary</p>
```

```
.text-primary  
.text-secondary
```

```
.text-success
```

```
.text-danger
```

```
.text-warning
```

```
.text-info
```

```
.text-light
```

```
.text-dark
```

```
.text-muted
```

```
.text-white
```

QA Margins and padding

All spacing is abbreviated for ease

Where `property` is one of:

- `m` - for classes that set `margin`
- `p` - for classes that set `padding`

Where `sides` is one of:

- `t` - for classes that set `margin-top` or `padding-top`
- `b` - for classes that set `margin-bottom` or `padding-bottom`
- `l` - for classes that set `margin-left` or `padding-left`
- `r` - for classes that set `margin-right` or `padding-right`
- `x` - for classes that set both `*-left` and `*-right`
- `y` - for classes that set both `*-top` and `*-bottom`
- blank - for classes that set a `margin` or `padding` on all 4 sides of the element

Where `size` is one of:

- `0` - for classes that eliminate the `margin` or `padding` by setting it to `0`
- `1` - (by default) for classes that set the `margin` or `padding` to `$spacer * .25`
- `2` - (by default) for classes that set the `margin` or `padding` to `$spacer * .5`
- `3` - (by default) for classes that set the `margin` or `padding` to `$spacer`
- `4` - (by default) for classes that set the `margin` or `padding` to `$spacer * 1.5`
- `5` - (by default) for classes that set the `margin` or `padding` to `$spacer * 3`
- `auto` - for classes that set the `margin` to `auto`

```
<div class="mb-2">
```

• Simply means this div has a margin on the bottom edge of size 2

QA Basic layout tricks

Bootstrap makes it easy to separate parts of your website by building containers

```
<div class="container">
  <div class="grid">
    <div class="row">
      <div class="col-md-3">
        CONTENT HERE
      </div>
    </div>
  </div>
</div>
```

- Section goes into a container
- Grid set up inside the container (same 12 columns)
- Row added to separate content
- First 3 columns (on a medium Screen) are used
- This can be repeated to have 4 items side by side in the same row

Bootstrap Components



QA Navbars

Including automatic collapse with screen resize

Navbar Home Features Pricing Disabled

HTML



```
<nav class="navbar navbar-expand-lg bg-body-tertiary">
<div class="container-fluid">
  <a class="navbar-brand" href="#">Navbar</a>
  <button class="navbar-toggler" type="button" data-bs-toggle="collapse" data-bs-target="#navBarNav"
    <span class="navbar-toggler-icon"></span>
  </button>
<div class="collapse navbar-collapse" id="navBarNav">
  <nav class="navbar navbar-expand-lg bg-body-tertiary">
    <div class="container-fluid">
      <a class="navbar-brand" href="#">Navbar</a>
      <button class="navbar-toggler" type="button" data-bs-toggle="collapse" data-bs-target="#navBarNav"
        <span class="navbar-toggler-icon"></span>
      </button>
    <div class="collapse navbar-collapse" id="navBarNav">
      <ul class="navbar-nav">
        <li class="nav-item">Home</li>
        <li class="nav-item">Features</li>
        <li class="nav-item">Pricing</li>
        <li class="nav-item">Disabled</li>
      </ul>
    </div>
  </div>
</nav>
</div>
</div>
```

Navbar



```
<nav class="navbar navbar-expand-lg bg-body-tertiary">
<div class="container-fluid">
  <a class="navbar-brand" href="#">Navbar</a>
  <button class="navbar-toggler" type="button" data-bs-toggle="collapse" data-bs-target="#navBarNav"
    <span class="navbar-toggler-icon"></span>
  </button>
<div class="collapse navbar-collapse" id="navBarNav">
  <ul class="navbar-nav">
    <li class="nav-item">Home</li>
    <li class="nav-item">Features</li>
    <li class="nav-item">Pricing</li>
    <li class="nav-item">Disabled</li>
  </ul>
</div>
</div>
</div>
```

QA Carousel

Just drop the images into the code and let it do the work



HTML

```
<div id="carouselExampleIndicators" class="carousel slide">  
  <div class="carousel-indicators">  
    <button type="button" data-bs-target="#carouselExampleIndicators" data-bs-slide-to="0" class="active" aria-current="page">1</button>  
    <button type="button" data-bs-target="#carouselExampleIndicators" data-bs-slide-to="1" aria-current="page">2</button>  
    <button type="button" data-bs-target="#carouselExampleIndicators" data-bs-slide-to="2" aria-current="page">3</button>  
  </div>  
  <div class="carousel-inner">
```

QA Cards

Repeated sections for similar content

Image cap

Card title

Some quick example text to build on the card title and make up the bulk of the card's content.

[Go somewhere](#)

HTML

```
<div class="card" style="width: 18rem;">
  
  <div class="card-body">
    <h5 class="card-title">Card title</h5>
    <p class="card-text">Some quick example text to build on the card title and make up the bul
      <a href="#" class="btn btn-primary">Go somewhere</a>
    </p>
  </div>
```



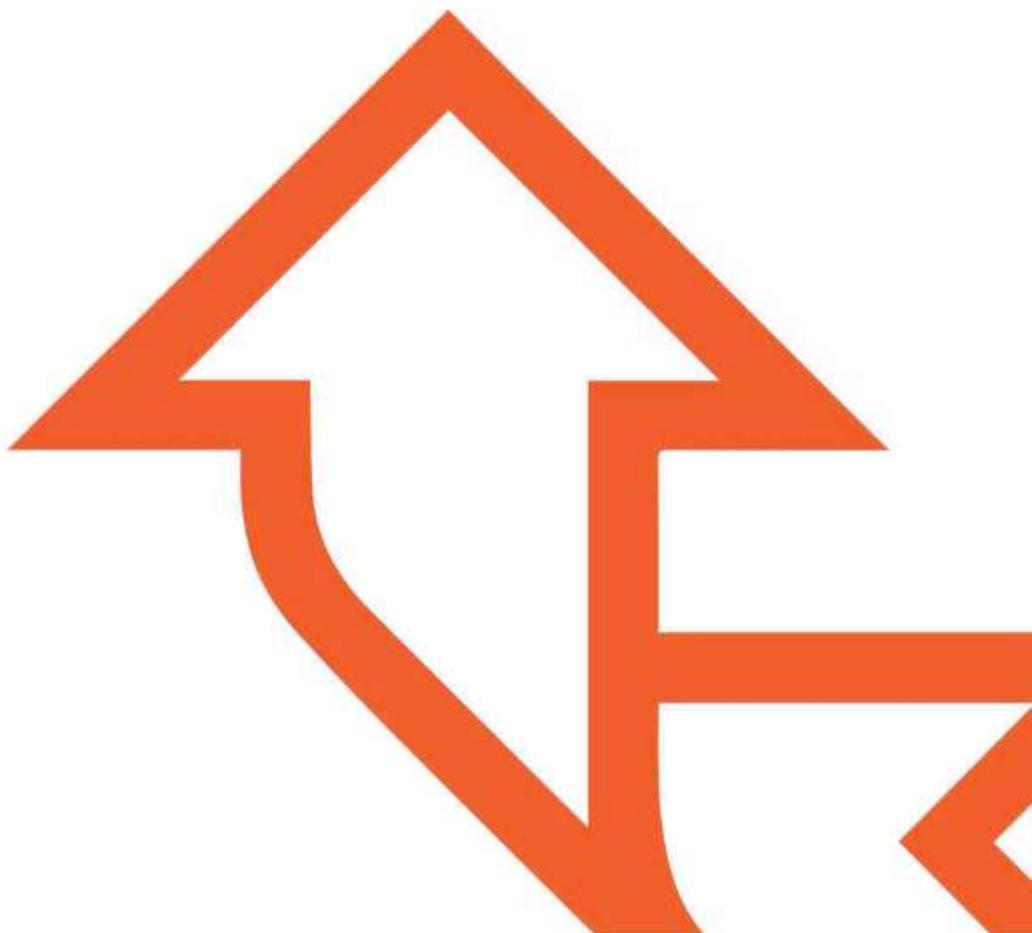
Demonstration

Hackathon – QA Cinemas

You have a brief for a website to be set up using the skills, knowledge and understanding of HTML, CSS and RWD gained on the course

- The brief has a number of user stories to tackle
- You are expected to:
 - Have a planning meeting to decide how the work will be distributed
 - Create the website
 - Have a review meeting to demonstrate the website to your trainer
 - Hold a retrospective to review what has been learnt as part of the process

Conclusion





Overview

- Checking objectives
- References
- What next?

Review Aims And Objectives

By the end of the course, you will be able to:

- Describe how web pages are delivered over the Internet
- Build structured HTML pages with text, links, images, tables and forms
- Use style sheets (CSS) for colours, background, formatting text, page layout and simple transformation, and animation effects
- Use Responsive Web Design techniques to make pages display well on all devices they may be viewed on.

QA

What's Next?



Further reading



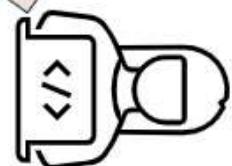
Course material



Consolidate learning



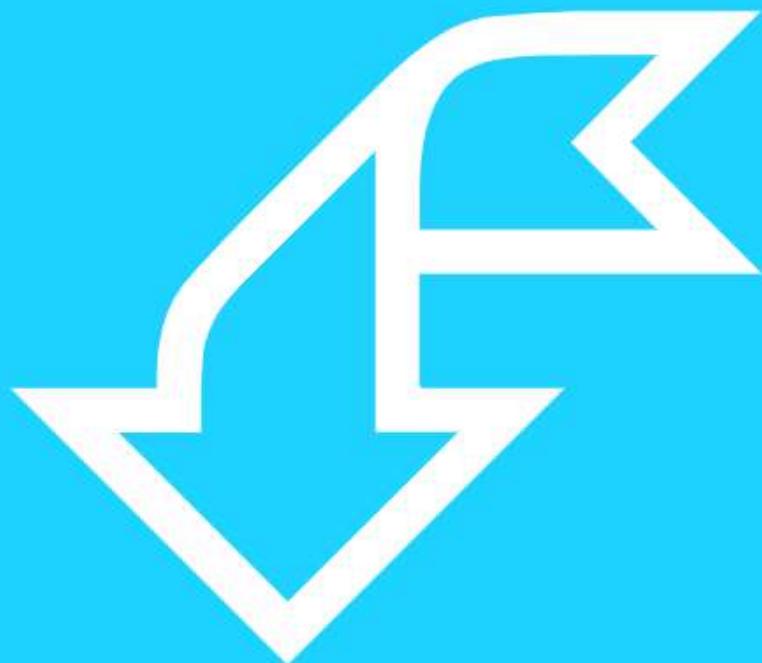
support
from
QA



Doing it
for real

VQ

Hope you enjoyed
this training
course.



**Please complete your
evaluation**

*Code: QAHMLCSS
PIN:*