HTML & Forms

Programming with Web Technologies

THE UNIVERSITY OF AUCKLAND
NEW ZEALAND
THE UNIVERSITY OF WAIKATO
Te Whare Wānanga o Waikato
Auckland ICT Graduate School

Today

More useful HTML tags to build on what we already know

- Anchors
- Multimedia
- Forms
- IFrames

Less obviously useful tags

- More on these in future lectures

Anchors

Also known as links, or hyperlinks, anchors are what allow us to navigate easily from page to page

Generally look something like this, but can be styled to look different

Can link to external pages, or to sections within the same page

Anchor Tags

<a> - The anchor tag. Marks the content surrounded by the tags as a link. The content can be text or even an image.

Anchors have a required attribute `href` which indicates the address of the resource they link to.

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

Google

href Attribute - External

Absolute Address

http://www.fakewebsite.com/res/2017.html

Equivalent to specifying a street address in long-form

130 Victoria St

Hamilton

New Zealand

Indicates exactly where to find the page and how to get there

href Attribute - External

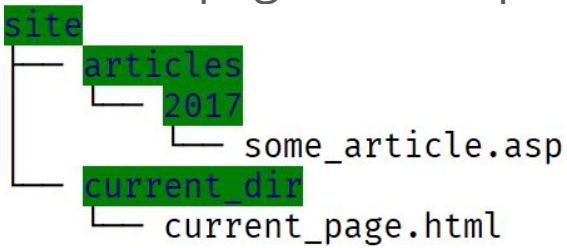
Relative Address

../articles/2017/some_article.asp

Equivalent to specifying a nearby street address

> Go back up the road and take the next right, you will find it at number 17.

Indicates exactly where to find the page with respect to the current page



href Attribute - External

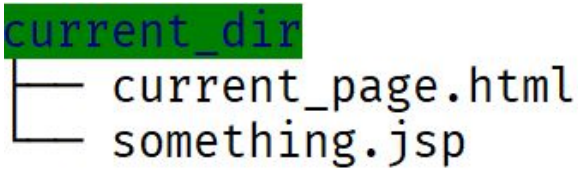
Local Link

something.jsp

Equivalent to sending someone to another house on the same street

> You want number 20, right next door

Indicates exactly where to find the page with respect to the current page



href Attribute - Bookmark

A bookmark is an anchor that jumps to a specific part of a web page

You will have encountered one of these if you have used the contents of a Wikipedia page

Bookmarks are created by setting the `id` attribute of an element

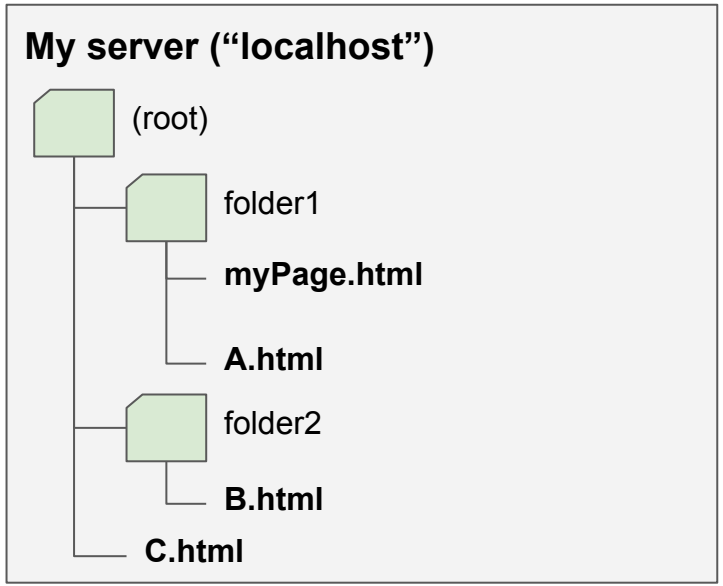
```
<h2 id="example_bm">A heading</h2>
```

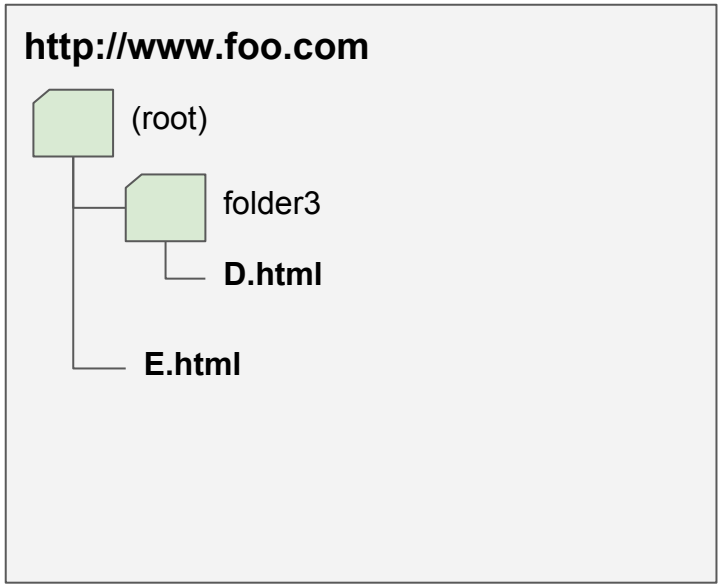
Anchors can then link to the bookmark

```
<a href="#example_bm">Link to heading</a>
```

Quiz

- Within myPage.html below, what would be the value of the `href` attribute required to link to each of the other html pages in the diagram?





Images

`<img>` - The image element. This tag does not need to be closed

Images have a required attribute `src` which, like the `href` attribute used with anchors, indicates where the image can be found.

```
<img src="http://waikato.ac.nz/coa.gif">
```

THE UNIVERSITY OF
WAIKATO
Te Whare Wānanga o Waikato

Images - Attributes

```
<img src="http://waikato.ac.nz/coa.gif" height="30">
```


```
<img src="http://waikato.ac.nz/coa.gif" width="80">
```


```
<img src="http://waikato.ac.nz/coa.gif" height="100" width="100">
```

Images - Attributes

What if the image file isn't available, or the user is visually impaired (using a screen reader)?

The `alt` attribute provides a textual description

```
<img src="http://waikato.ac.nz/coa.gif" alt="U of W logo">
```

U of W logo

Images - File Types

Browsers usually support: GIFs, PNGs, JPEGs/JPGs, BMP and ICO; but not TIFF, JPEG2000, DjVu, etc

JPEGS are a **Lossy** format, with the quality settings set in software

> Choose JPEGs for photos and photo-like images

GIFs & PNGs are **Loss-less** formats, PNG usually produces smaller file sizes

> Choose GIFs or PNGs for flat colour, graphic text, simple logos and screenshots. Choose PNG if you want transparency

GIFs can also be animated, PNG animation (APNG, MNG) is not supported by browsers, so simple image animation on the Web has to be GIF

Images - More points

You can resize a large image to fit in a smaller area using the width and height attributes, however this is not always wise

- You are still downloading a large photo, which can be slow

Sometimes you are better off creating a smaller "thumbnail" version of the image and linking to the larger version

```
<a href="fullsize_image.png">
    <img src="thumbnail_image.png">
</a>
```

Multimedia - Audio and Video

<audio> - A container for audio files. The audio tag represents the controls, or player for audio-only files

<video> - A container for video files. The video tag represents the controls, or player for audio/visual files

<source> - Represents a possible source location for audio or video files. Source elements should be placed inside the appropriate <audio> or <video> tags

Multimedia - Using the tags

```
<audio controls autoplay>
    <source src="myaudio.mp3" type="audio/mpeg">
    <source src="myaudio.ogg" type="audio/ogg">
    Your browser does not support the audio element.
</audio>

<video controls width="640" height="480">
    <source src="myvideo.mp4" type="video/mp4">
    Your browser does not support the video element.
</video>
```

Why multiple <source>'s

Not all browsers across the operating systems support the same audio and video formats

The browser will try to play each source until it finds one that it can handle.

By providing multiple options, we can hopefully get the media to play for everyone

If all else fails, we can display an error message

Multimedia - Patents

Variability of codec support in browsers result of patents (and "politics" to some extent)

Audio

MP3 Support fairly widespread in browsers now

Video

Video codec support more variable than audio

High-Level View of the Web

- A typical web page consists of many separate resources
  - HTML markup
  - Images
  - JavaScript code
  - Presentation details using CSS
  - Video
  - Audio
- Resources that make up a page can come from different web servers
- The web browser needs to request each resource, manage the responses, and construct the final result into the final layout to display to the user

High-Level View of the Web

Computer running a web browser

Internet

Computers running web server software

High-Level View of the Web

`http://google.com`

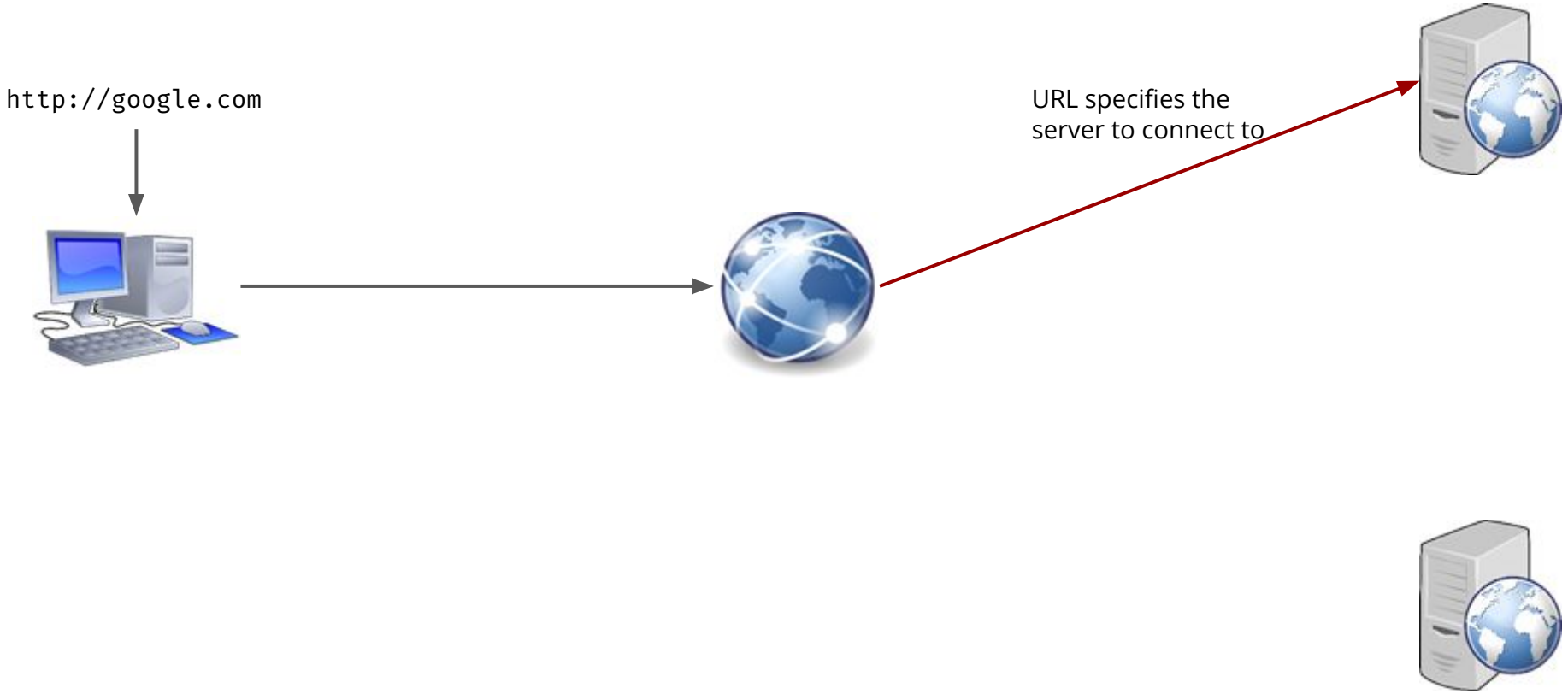
The web browser is given a URL to load a webpage

High-Level View of the Web

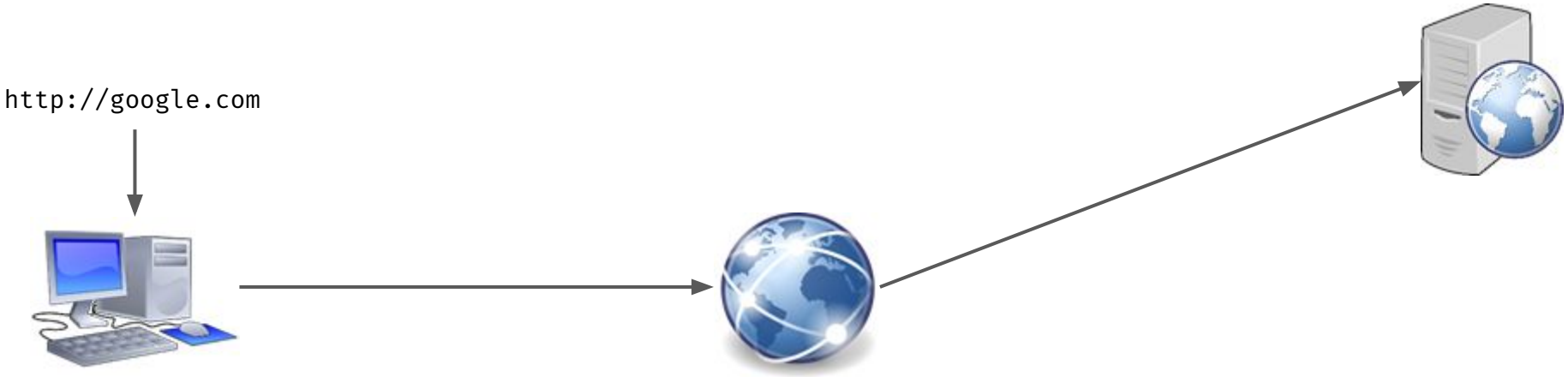
`http://google.com`

HTTP request for
the HTML page

High-Level View of the Web



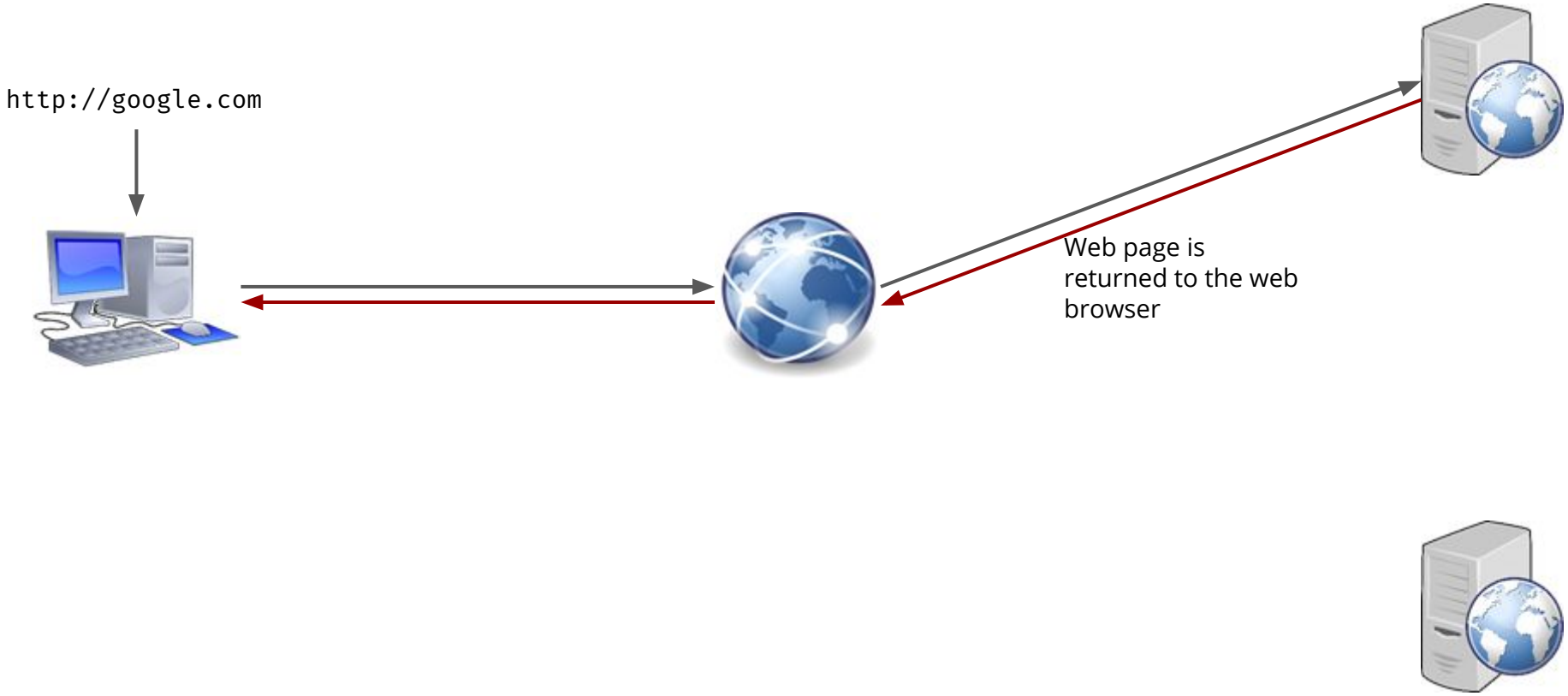
High-Level View of the Web



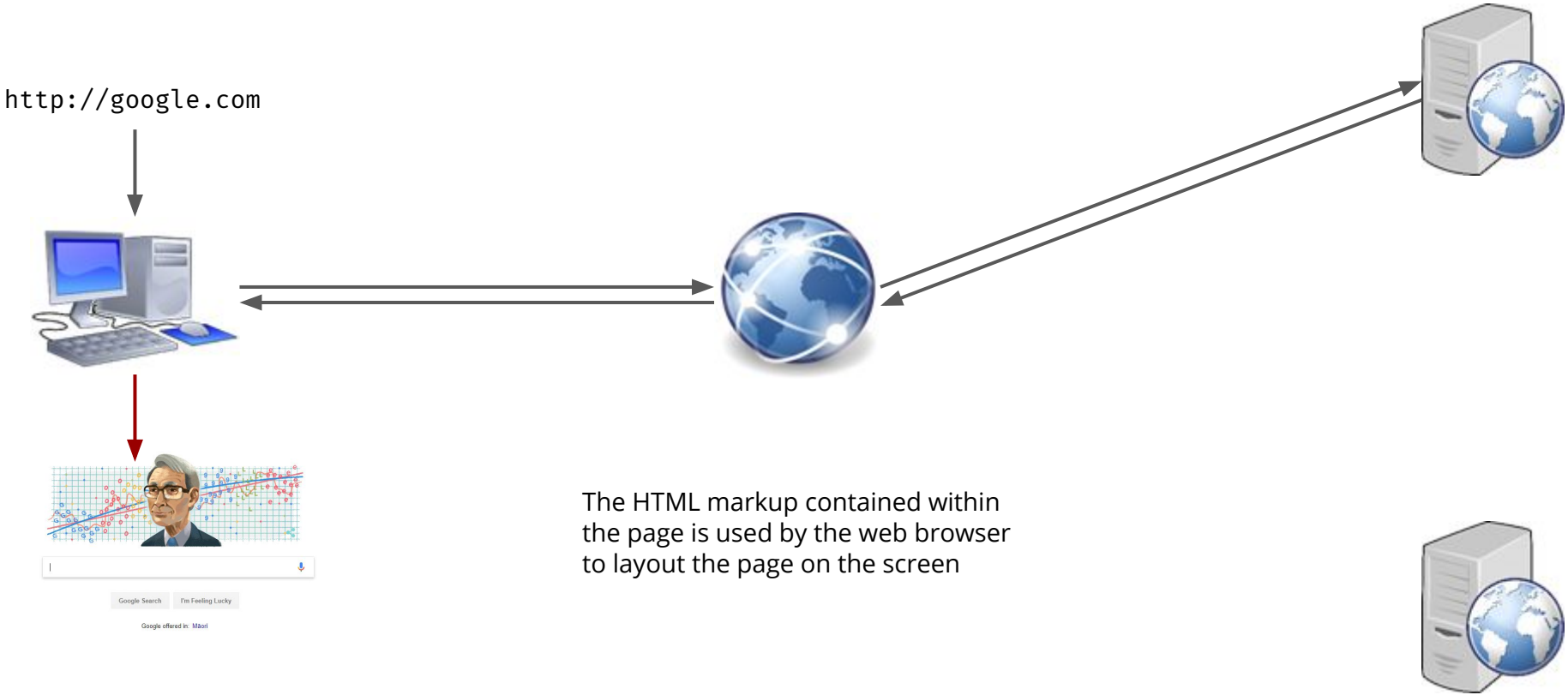
The URL also contains details about the page that is being requested.

Connected server retrieves the page from its hard drive

High-Level View of the Web

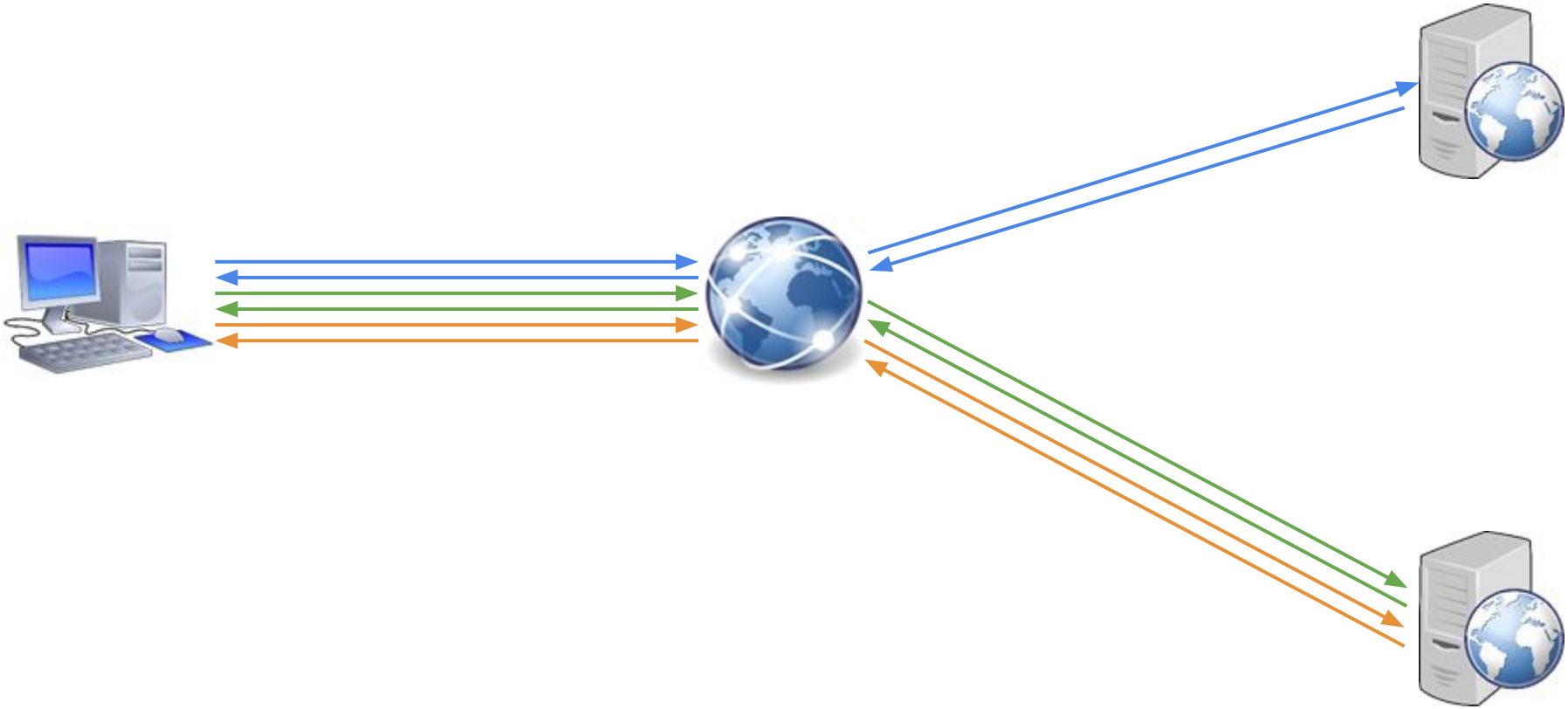


High-Level View of the Web



The HTML markup contained within the page is used by the web browser to layout the page on the screen

High-Level View of the Web



HTML Forms

In short, HTML Forms allow input rather than just reading Web content

An HTML ***form*** is a section of the HTML document containing special elements — called **controls** — which can be used to obtain data from the user. This data can then be processed either in the browser or on the server

- *http://www.w3.org/TR/html401/interact/forms.html*

Forms

We can use forms to allow users to input data which can be processed in the client, or sent to another site or application to process

- Login details (username + password)
- Registration details (name, contact info etc)
- Preferences for using the site
- Entering search queries
- Payment info for online purchases

There are a wide range of controls available to suit your needs

Forms - Tags

Like with tables, forms have a number of tags and attributes associated with them. The first one we need to know about is:

<form> - Defines the start and end point of the form. Groups the form components

All controls and labels that make up the form need to be contained within the <form> tags

The <form> tag has two important, but optional attributes: action & method

More on these shortly

Forms - Tags

`<input>` - An input element in a form, does not need a closing tag. Almost all form controls are inputs

All of the controls shown here are made using `<input>` tags

Made possible through gratuitous use of attributes, most importantly, the `type` attribute

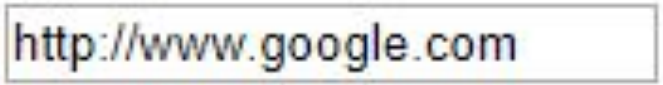
Forms - Text Input

```
<input type="text">
```
A single-line text box


```
<input type="password">
```
A specialized text-box that masks input

```
<input type="tel">
```
A text-box for entering phone numbers


```
<input type="url">
```
A text-box for entering URLs



Many more specialized options exist. Useful to use these, as they give context to browsers and other clients

Forms - Identifying and Naming Elements

If we are processing form information, then we need to know what values from each control means

Where did they come from, what do they represent

Use the `name=""` attribute to give a name to an input's value

```
<input type="text" name="firstname">
<input type="text" name="lastname">
```

Useful to software, but no so useful for the end-user. Would be useful to have some sort of `<label>` for these

Forms - Labels

`<label>` - Provides a non-editable textual label for other controls

Labels are associated with other controls through use of the `for` attribute

The value of a `label`'s `for` attribute should match the value of another element's `id` attribute

All html elements can have an `id` attribute, they must be unique

```
<label for="fnameID">First Name</label>
<input type="text" id="fnameID" name="firstname">
```

Forms - A Basic Form

```
<form>
  <label for="fnameID">First Name</label>
  <input type="text" id="fnameID" name="firstname">

  <label for="lnameID">Last Name</label>
  <input type="text" id="lnameID" name="lastname">
</form>
```

Forms - Grouping Controls

Often a form will be made up of a number of sections, that contain related controls. We can group our controls into sections with a fieldset

<fieldset> - Groups form controls with a visual border

<legend> - A short textual label for a fieldset. Should be the first element inside a fieldset

```
<fieldset>
  <legend>An example</legend>
  Controls here
</fieldset>
```

Forms - Checkboxes

Checkboxes normally appear in a logical grouping in which 0 or more items can be selected. They are created using `<input>` controls with the `type` attribute set to `checkbox`

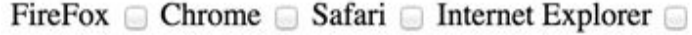
0 or more items can be selected by default, indicated by the presence of the `checked` attribute

The value sent to the processing application for a checked box can be controlled with the `value` attribute

Checkboxes can be grouped together by setting the `name` attributes of related inputs to the same value

Forms - Checkboxes

```
<label for="ffCheckbox"> FireFox</label>
<input type="checkbox" name="browser_choice" value="firefox" id="ffCheckbox">
<label for="cCheckbox"> Chrome</label>
<input type="checkbox" name="browser_choice" value="chrome" id="cCheckbox">
<label for="sCheckbox"> Safari</label>
<input type="checkbox" name="browser_choice" value="safari" id="sCheckbox">
<label for="ieCheckbox"> Internet Explorer</label>
<input type="checkbox" name="browser_choice" value="ie" id="ieCheckbox">
```



Forms - Radiobuttons

Radiobuttons normally appear in a logical grouping in which only 1 item can be selected. They are created using <input> controls with the type attribute set to radio

Normally 1 item is selected by default, indicated by the presence of the checked attribute

The value sent to the processing application for a checked box can be controlled with the value attribute

Related radiobuttons should be grouped together by setting the name attributes of related radiobuttons to the same value

Forms - Radiobuttons

```
<label for="maleRadioID">Male</label>
<input type="radio" checked id="maleRadioID" value="male" name="Gender">
<label for="femaleRadioID">Female</label>
<input type="radio" id="femaleRadioID" value="female" name="Gender">
```

Form - Select

A select, also known as a drop-down list, provides a list of items for the user to choose from

<select> - Groups options together to form a drop-down list. Should have a name attribute set to identify the selected option for the processing application

<option> - An option that exists within a <select>. The value sent to the processing application for a selected item can be controlled with the value attribute. The default option can be set with the selected attribute

Form - Select

```
<label for="countryID">Country</label>
<select id="countryID" name="countryName">
    <option value="nz" selected>New Zealand</option>
    <option value="aus">Australia</option>
    <option value="in">India</option>
    <option value="us">United States</option>
    <option value="other">other</option>
</select>
```

Forms - More Text Input

The `textarea` control allows for multiline textual input by the user. By default, the textarea is shown in a monospaced (fixed width) font, and needs to be told via the `rows` and `cols` attributes how many rows and columns worth of text to show without scrollbars

If `rows` or `cols` limits are exceeded, the browser is responsible for wrapping text or adding scroll bars

Can optionally provide initial text as the content of the control

Forms - More Text Input

```
<label for="profile_text"> About Text</label>
<textarea id="profile_text"
          name="profile_text_area"
          rows="4"
          cols="40">
    enter text here
</textarea>
```

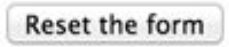
Forms - Reset

The `reset` control is an input element with the `type` attribute set to `reset`, and is presented by the browser as a button

When clicked, all form controls are reset to their initial values. The browser handles this, we do not need to do this ourselves

By default, the button will appear with the text "reset", but if an alternative label is desired, it can be provided via the `value` attribute

```
<input type="reset" value="Reset the form">
```



Forms - Submit

The `submit` control is an input element with the `type` attribute set to `submit`, and is presented by the browser as a button

When clicked, the values selected/provided in form controls are sent to the processing application

By default, the button will appear with the text "submit", but if an alternative label is desired, it can be provided via the `value` attribute

```
<input type="submit" value="Submit for processing">
```

Forms - Submitting

When the user clicks the 'Submit' button we want to have the form data processed in some way. This is normally done by a piece of program code e.g., JavaScript code executed in the browser, or alternatively by server-side code (e.g. a Servlet or PHP script) executed on a server

We might call this piece of code a form processing agent – and we need to identify the agent that the form data should be sent to

We identify the agent in the `action` attribute of the <`form`> element

Forms - Submitting

The value of the action attribute is usually an absolute or relative URL of a server side script

```
<form action="https://sporadic.nz/web_lab_echo_service/">
   <!– form content here -->
</form>
```

```
<form action="../processing_script.jsp">
   <!– form content here -->
</form>
```

Form - Methods

Along with the `action` attribute a `form` element has a `method` attribute which defines the HTTP method (verb) to use when communicating with the server

> For forms, the verb is normally **GET** or **POST**

The method chosen affects the HTTP message header and determines the way in which the data will be transported to the server

> The HTTP message header is a bit out of scope for now, but the way in which the data is transported is relevant to us

GET vs POST

The browser is responsible for sending data from a form to the the destination specified in the action attribute

If the **GET** method is used, `name=value` pairs are appended to the action attribute address, starting with a "?", separated by "&"s

```
https://ex.com/form.jsp?fname=Peter&lname=Jackson&gend=male
```

If the **POST** method is used the data is sent as the body of a message

Not visible in the browser/history. Useful for passwords!

IFrames

Sometimes it is useful to be able to display one web page inside another. Enter the inline frame, or `<iframe>`

`<iframe>`s can be used to embed one page in another

- Preview a webpage from within yours
- Display a map so users don't leave your page
- Include a "common" piece of HTML in multiple pages

IFrames – The tag

`<iframe>` - Represents the frame in which a linked page is displayed

Iframes, like images, have a required attribute `src` which indicates where the linked page can be found.

```
<iframe src="common.html"></iframe>
```

Divs and Spans

To wrap up the day, we have two mystery tags which we will use in future labs

 - A block-level element that has no default appearance. It can be used to group block-level elements

 - An inline element that has no default appearance. It can be used to group inline elements

If you inspect any modern web page, you will see a huge number of 's and 's. Any guesses as to why?