Intro to HTML & CSS

Presented by Toni Gardiner

In case of emergency

Slides & files

Overview

- ## HTML
  How markup works, basic elements, links, images, lists

- ## CSS
  Anatomy of a CSS rule, specificity, box model

  Types of styling we can apply, display and positioning

  Workflow and debugging

- ## Build a web page

My background

- Front end developer (HTML/CSS)
- Web design, graphic design
- Art and design

Getting started

- Who has done some HTML before?
- CSS?
- Existing issue you'd like to resolve?

Questions

Please ask!

HTML

HyperText Markup Language (HTML)

The standard markup language for creating web pages and web applications.

HyperText

Text and links.

Markup Language

The good part!

Markup Language

A set of elements (aka tags) which can be applied to pieces of text to give them different meaning in a document.

It's how the computer interprets your content.

Imagine you are a computer.

You are supposed to use the following sentence on a webpage:

```
This is a very dangerous liquid.
```

How does the computer know to give the words 'very dangerous' emphasis?

We have to tell it — mark it up!


```
This is a [start strong importance]very
dangerous[end here please] liquid.
```

This is where the language bit comes in:

This is a very dangerous liquid.

gives us:

This is a **very dangerous** liquid.

Basic HTML page

```html
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="utf-8">
  <title>Title of the page (browser tab)</title>
</head>

<body>
  <!-- Content start -->
  <p>This is a <strong>very dangerous</strong> liquid.</p>
  <!-- Content end -->
</body>

</html>
```

Let's add the paragraph above

Code speak

- **Indentation** — tabs, spaces
- **Comments** — `<!-- This is a comment (in HTML) -->`
- **Tab complete** — yay!
- Read the **docs** — these are your friends

Any others you've heard of?

Semantic

Relating to **meaning** in language or logic.

HTML elements:
Text

- **Headings** — <h1>, <h2>, <h3>, <h4>, <h5>, <h6>

- **Paragraph** — <p>

- **Semantics** — <strong>, <em> (different to <b>, <i>)

- **Line break** —
 (self-closing)

- **Span** — <span> (non-semantic)

HTML elements:
Ordered list

```
<ol>
  <li>First place</li>
  <li>Second place</li>
  <li>Third place</li>
</ol>
```

1. First place
2. Second place
3. Third place

HTML elements:
Unordered list

```
<ul>
  <li>Oranges</li>
  <li>Lemons</li>
  <li>Plums</li>
</ul>
```

- Oranges
- Lemons
- Plums

HTML elements:
Links

```
<a href="[path-to-file]">Link text</a>
```

[Link text](#)

In a sentence:

```
<p>Learn more about <a href="team.html">our team</a></p>
```

Learn more about [our team](#)

HTML elements: Links
Your site – Email link – External site

```
<a href="team.html">Page about your team</a>

<a href="mailto:toni@catalyst.net.nz">email Toni</a>

<a href="https://www.wikipedia.org">Wikipedia website</a>
```

```
<a href="https://www.wikipedia.org"
target="_blank" rel="noopener noreferrer">Wikipedia website</a>
```

HTML elements:
Images

```
<img src="cute-cat.png" alt="cute cat plays in grass" />
```

- Formats / .jpeg (.jpg) / .png / .gif / .svg
- Prepare image with design software
- Resolution/optimisation
- Change size/presentation with CSS

HTML elements:
Images

```
<img src="cute-cat.png" alt="cute cat plays in grass" />
```

- Formats / .jpeg (.jpg) / .png / .gif / .svg
- Prepare image with design software
- Resolution/optimisation
- Change size/presentation with CSS

HTML elements:
Parts of a web page

- `<header>`
- `<nav>`
- `<section>`
- `<aside>`
- `<footer>`
- `<div>` (non-semantic)

The docs

developer.mozilla.org — HTML elements

CSS

Cascading Style Sheets (CSS)

A language used for describing the presentation of a document written in a markup language.

CSS

We're using it to **Style** the page

It's separate from the HTML (is its own **Sheet**)

It's **Cascading***

*We'll get to this part soon

HTML

Content

CSS

Presentation

Left version

Museum of New Zealand Te Papa Tongarewa

Open every day 10am—6pm
(except Christmas Day)

Free entry
Charges apply to some short-term exhibitions and activities

55 Cable Street, Wellington

- Visit Toro mai
  - Plan your visit Whakaritea tō toronga
  - Exhibitions Ngā whakaaturanga
  - Events Ngā kaupapa motuhake
  - Guided tours He haerenga ārahi
  - Book tickets Whakarite tīketi
  - Venues Wāhi
- Discover the collections Tūhuratia ngā kohinga
  - Collections Online
  - Blog
  - Read, watch, play Kōrero, mātaki, purei
- Learn Ākona
  - For educators Mā te pouako
  - For museums and galleries Mō ngā muhiama me ngā whare toi
  - Research Rangahau
  - Guides to caring for objects Tiaki Kohinga, Tiaki Taonga
- About Mō Te Papa
  - Contact us Whakapā mai
  - News He pānui
  - What we do Ā mātou mahi
  - The collections Ngā kohinga taonga
  - Repatriation Karanga Aotearoa
  - Touring exhibitions Ngā whakaaturanga poi haere
  - Past exhibitions Ngā whakaaturanga o mua
  - Jobs Tūranga mahi
  - Te Papa Press
  - Press and media Papāho
  - Media sales and licensing Te hohoko papāho me te manatā
  - Our building Tō mātou whare
- Shop Wharehoko
- Support & join Tautokotia, kuhu mai
  - Our membership programme – Friends of Te Papa Te hōtaka mema – Ngā Hoa o Te Papa
  - Te Papa Foundation
  - Corporate partnerships Hononga kaipakihi
  - Support from trusts and foundations Ngā hoa whakawhirinaki ā-pūtea

Search [Search]

- Facebook⧉
- Twitter⧉
- Instagram⧉
- YouTube⧉

Right version

**MUSEUM OF
NEW ZEALAND
TE PAPA
TONGAREWA**

Open every day 10am—6pm
(except Christmas Day)

Free entry
Charges apply to some short-term exhibitions and activities

55 Cable Street, Wellington

VISIT
Toro mai

ABOUT
Mō Te Papa

Search [🔍]

DISCOVER THE COLLECTIONS
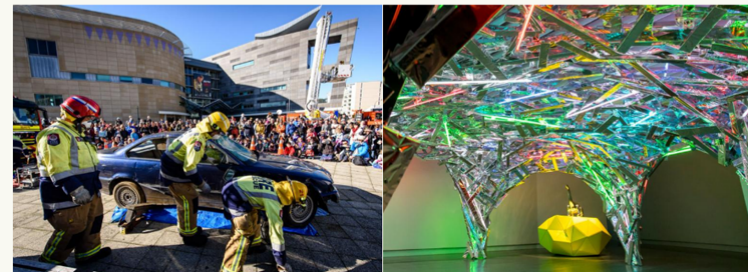Tūhuratia ngā kohinga

SHOP
Wharehoko

LEARN
Ākona

SUPPORT & JOIN
Tautokotia, kuhu mai

Terracotta Warriors: Guardians of Immortality – Tickets now on sale
Get up-close to some of China's ancient imperial icons this summer. Opening Sat 15 Dec.

Link to your CSS file

In the <head> section of your HTML document:

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

Note: not in the <header>!

Working with HTML

HTML

```
<p>Text</p>
```

CSS

```
p { ... }
```

A simple CSS rule

```
p {
  color: green;
  font-weight: bold;
}
```

A simple CSS rule

```
p {                        /* selector */
  color: green;            /* rule */
  font-weight: bold;       /* rule */
}
```

Let's get more specific

There are different ways of referring to a specific element or elements on the page.

In the HTML, we give elements either a **class** or an **ID** (or both!).

```
<p>Paragraph which needs different styling</p>
<p>Paragraph which is standard</p>
```

```
<p class="intro">Paragraph which needs different styling</p>
<p>Paragraph which is standard</p>
```

Now we can refer to the class name in the CSS

For a class, we use a period:

```css
.intro {
  font-size: 20px; /* note: default is 16px */
}
```

A class can be used more than once

Using a class

```
<p class="intro">Paragraph which needs different styling</p>
```

To refer to a class, we use a period:

```
.intro {
  font-size: 20px;
}
```

A class can be used more than once

Using an ID

```
<p id="elephant-info">Elephants are big</p>
```

To refer to an ID, we use a hash:

```
#elephant-info {
  color: green;
}
```

An ID is unique — it can only be used once.

HTML

```
<p>Text</p>
```

```
<input id="vote">
```

```
<p class="time">Text</p>
```

```
<p class="time new">Text</p>
```

CSS

```
p { ... }
```

```
#vote { ... } /* unique */
```

```
.time { ... } /* resuable */
```

```
.time.new { ... }
```

Inheritance and specificity

The great thing about CSS is that we can stack the selectors.

Let's say we have this HTML:

```
<p>Blah blah <a href="#">first link</a> blah</p>

<p class="fact">Blah <a href="#">second link</a> blah</p>
```

And we want to make only the second link green

Let's make a rule for that...

```
.fact a { color: green; }
```

This will only affect 'a' tags that are contained within elements that have the 'fact' class.

What happens if we also have a rule to make all the links blue?

```
a { color: blue; }
```

Most specific wins

The most specific rule will always be applied where possible

But the ordering of CSS rules **is** important

If two (or more) rules are equally specific, the lowest (bottommost) rule wins

Note:

- **Class selectors**

  are more specific than element selectors

- **ID selectors**

  are more specific than class selectors

What can we do with CSS?

All sorts! Here are a few things we can adjust:

- Borders
- Padding
- Margins
- Widths

- Rounded corners
- Opacity
- Font size, weight, color*
- Background color*

*color!

Box model

MARGIN

BORDER (5px)

PADDING

Paragraph text

CONTENT

40px   110px   40px
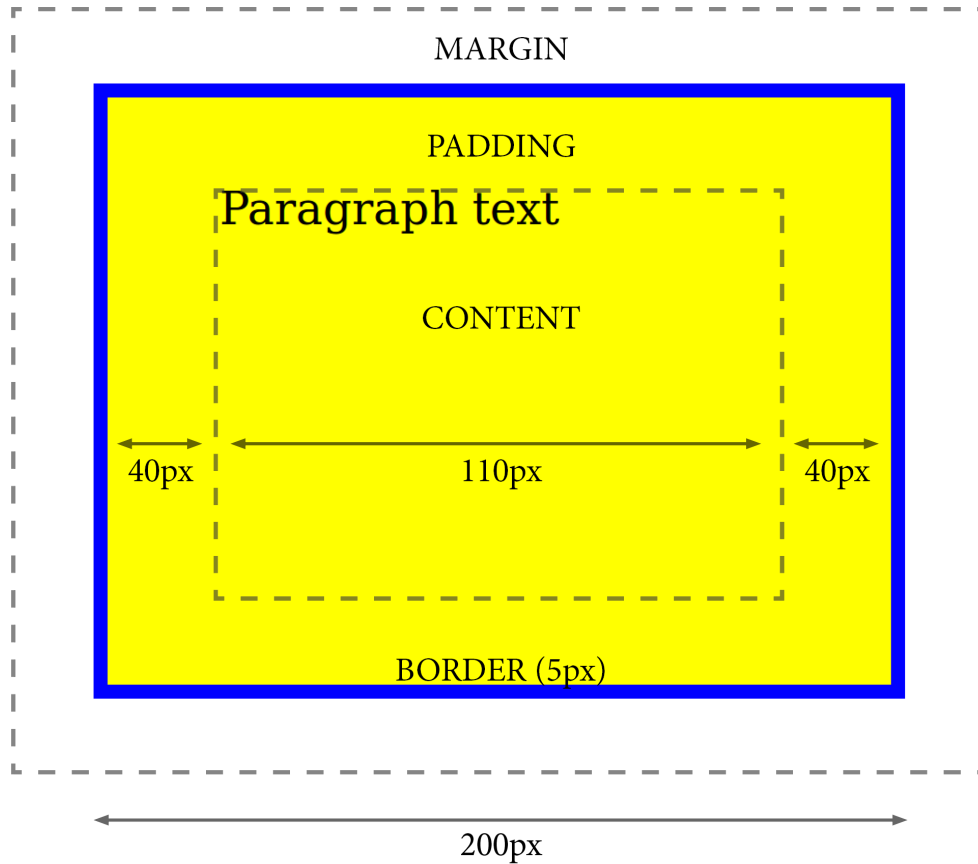
200px

Let's make the example

```
div {
  background-color: yellow;
  padding: 30px 40px;
  width: 200px;
  height: 150px;
  border: 5px solid blue;
  margin: 30px;
  box-sizing: border-box; /* see note* */
}
```

* Tells browser to account for any border and padding in the values you specify for an element's width and height. This typically makes it much easier to size elements.

Shorthand options

When declaring sizes, we can:

```
margin-top: 10px;
margin-right: 20px;
margin-bottom: 10px;
margin-left: 20px;

margin: 10px 20px 5px 15px; /* [top] [right] [bottom] [left] */
margin: 10px 20px 5px;      /* [top] [right + left] [bottom] */
margin: 10px 20px;          /* [top + bottom] [right + left] */
margin: 20px;               /* [all values are the same] */
```

Inline, block and inline-block

Note: examples use the same HTML

Display: inline

```
<p>Text with <a href="#">link</a> <a href="#">link</a> inside</p>
<p>Text without</p>
```

```css
a {
  display: inline; /* note: default */
  padding-bottom: 20px;
}
```

Text with link link inside

Text without a link

Display: block

```
a {
  display: block; /* change */
  padding-bottom: 20px;
}
```

Text with

link

link

inside

Text without a link

Display: inline-block

```
a {
  display: inline-block; /* change */
  padding-bottom: 20px;
}
```

Text with link link inside

Text without a link

Floats

We want to 'float' this image so the text reflows around it. We can float things either to the left or the right. We need to be careful though, floated elements don't sit in the page properly any more. Read about 'clearfix' to learn about this.

```
img { float: left; margin: 0 20px 10px 0; }
```

Pseudo selectors

```
a:hover          /* link is interacted with, but not clicked */
a:active         /* link is being activated */
a:visited        /* link has been visited */

p:first-child    /* select the first paragraph (child) */
p:last-child     /* select the last paragraph (child) */
```

Positioning

```
static  /* default */

relative

absolute

fixed
```

The docs

developer.mozilla.org — CSS syntax/properties/selectors

Debugging & inspecting

Modern browsers give us great tools for checking how our CSS is being interpreted.

Right click on any element, choose **'inspect element'** option, or press **F12** to bring up the developer tools.

Chrome, Firefox, IE (Edge)

Build a web page

WLG Sites and Activities

- Header and logo

- Navigation (placeholder)

- Main content:

  - Heading and introductory text

  - Activities × 2: heading, image, text and link

- Footer and footer content

WLG SITES & ACTIVITIES

Colour values

■	#181d23	■	#ffcd03
■	#2478bd	■	#e9e9e9

Wellington sites and activities

Your guide to the best things to see and do in the coolest little capital.

Wellington is a city where having fun is easy. You can meet meerkats, kiwis and cave trolls on the same day. Paddle on the harbour in the morning, mountain bike in the afternoon. Galleries for breakfast, museums for lunch and theatre for dinner.

Wellington Cable Car

One of Wellington's most popular tourist attractions, the Wellington Cable Car runs from downtown Wellington to the picturesque suburb of Kelburn and Wellington Botanic Garden.

Wellington Cable Car website

The Roxy Cinema

Breathing life back into one of Wellington's original suburban cinemas, Roxy Cinema is a labour of love for a group of the city's most successful film and hospitality personalities.

The Roxy Cinema website

Follow us on Twitter

Content sourced from WellingtonNZ.com

61

Extra

- Make the header and footer content line up with the main content.
- Include other HTML pages — link using nav area.
- Add Twitter logo.
- Make the web page responsive (hint: start by removing width).

Going forward

- Practise
- Content comes first
- Use docs
- Use browser developer tools (**F12**)

Resources

- **HTML elements**
  https://developer.mozilla.org/en-US/docs/Web/HTML/Element

- **CSS syntax/properties/selectors**
  https://developer.mozilla.org/en-US/docs/Web/CSS/Reference

- **Browser support**
  http://caniuse.com/

- **Colour contrast checker**
  https://webaim.org/resources/contrastchecker/

freedom to innovate

catalyst

open source technologists