CODING FOR BEGINNERS: HTML AND CSS

Leonie Dunnett

INTRODUCTIONS

WHO AM I?

- Have been developing websites for around 10 years
- Lonely Planet (LP.com, Client Solutions), Intrepid Travel, not-for-profits and freelance.
- I love learning and meeting new amazing people
- GA Profile, LinkedIn

WHO ARE YOU?

- Name
- What you do
- Why you're here
- Any web development experience?

BEFORE WE START

Have you installed:

`Sublime Text (Text Editor)`

`Google Chrome (Browser)`

TAKEAWAYS

- How the Internet and web pages work
- Introducing HTML / CSS `Code along`
- Getting your site on the Internet
- How to modify the code of existing websites
- Techniques to use at-home when stuck on code

HOW THE INTERNET AND WEB PAGES WORK

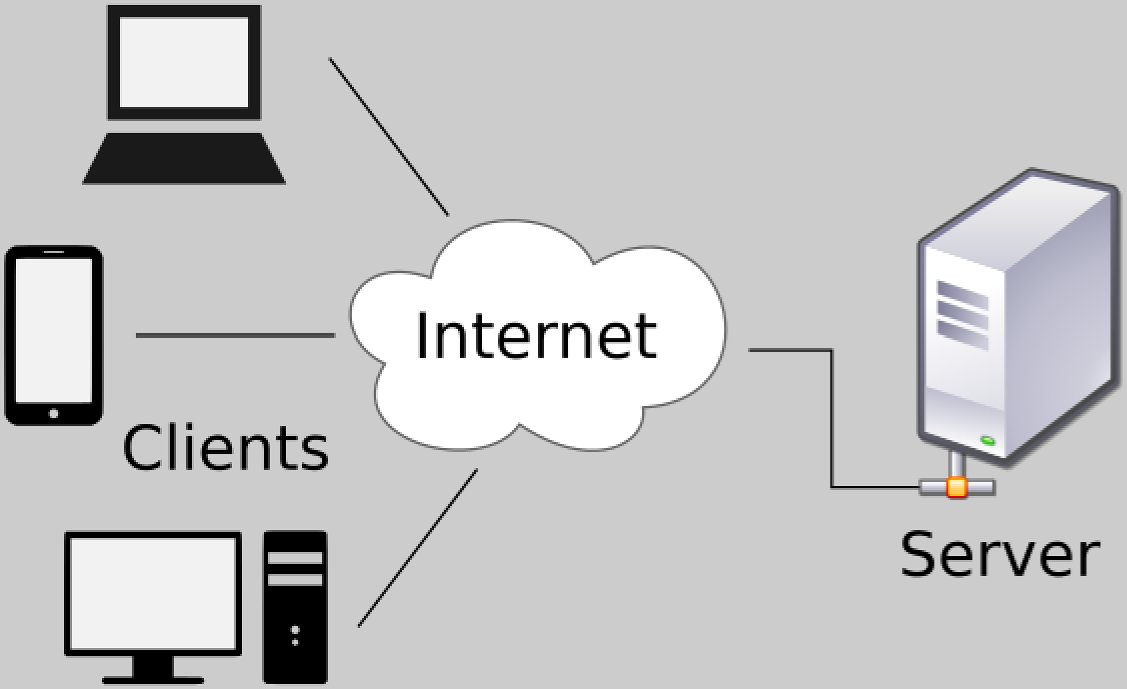
WHAT IS THE INTERNET?

- A worldwide telecommunications system
- A network that transports electronic messages across distance and platform

WHAT ARE SERVERS AND CLIENTS?

- A **server** is a fast computer with a huge amount of memory and space
- Internet networks are connected to servers
- Servers host web pages
- Our devices are considered the **client**

CLIENT-SERVER MODEL



Clients = service requesters

Servers = providers of a resource or service

HOW DO WE ACCESS THE INTERNET?

- An **Internet Service Provider (ISP)** enables us (the client) to access the Internet (via their server)
- A **browser** (such as Chrome, Firefox, or Safari) enables your device to read and display web pages

HOW DO WEB PAGES WORK?

- Web pages are writen in HTML (Hypertext Markup Language)
- A browser translates the HTML into the content you see on the screen.
- CSS (Cascading Style Sheet) controls how the page looks, making the HTML content pretty

HUMAN BODY ANALOGY

`HTML` = structure (bones)

`CSS` = style (skin/makeup)

`JavaScript` = behaviour
(nervous system - action, reaction, computation, etc)

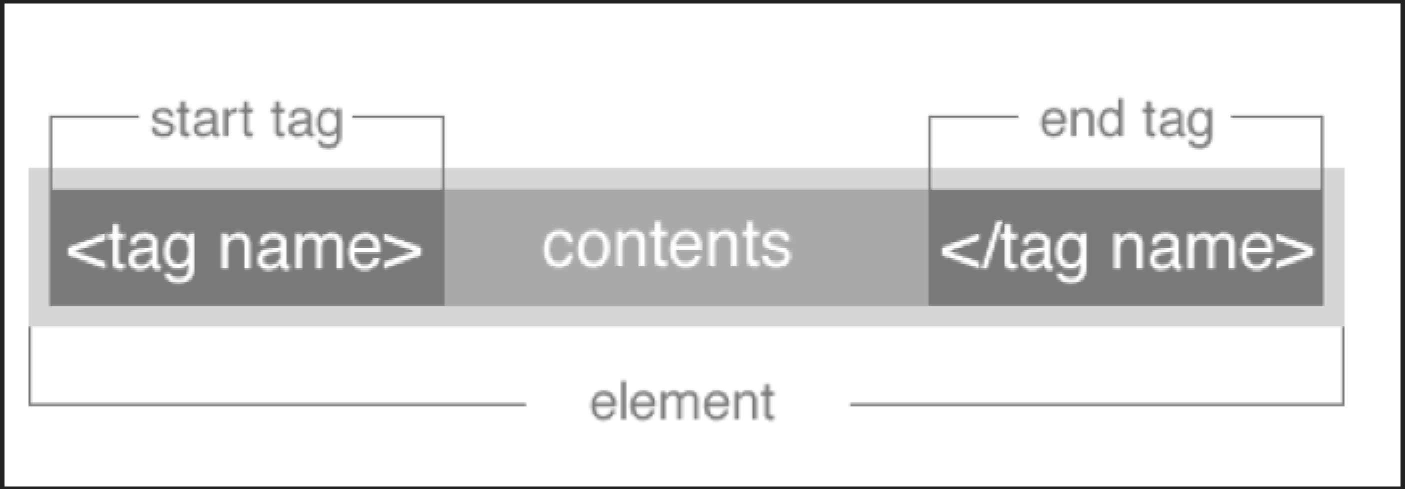
INTRODUCING HTML

WHAT IS HTML?

HTML = THE PAGE STRUCTURE

- Describes the kind of things on a page.
- Different elements have different somantic (structural) meaning.
- A page is a document.

HTML TAG SYNTAX

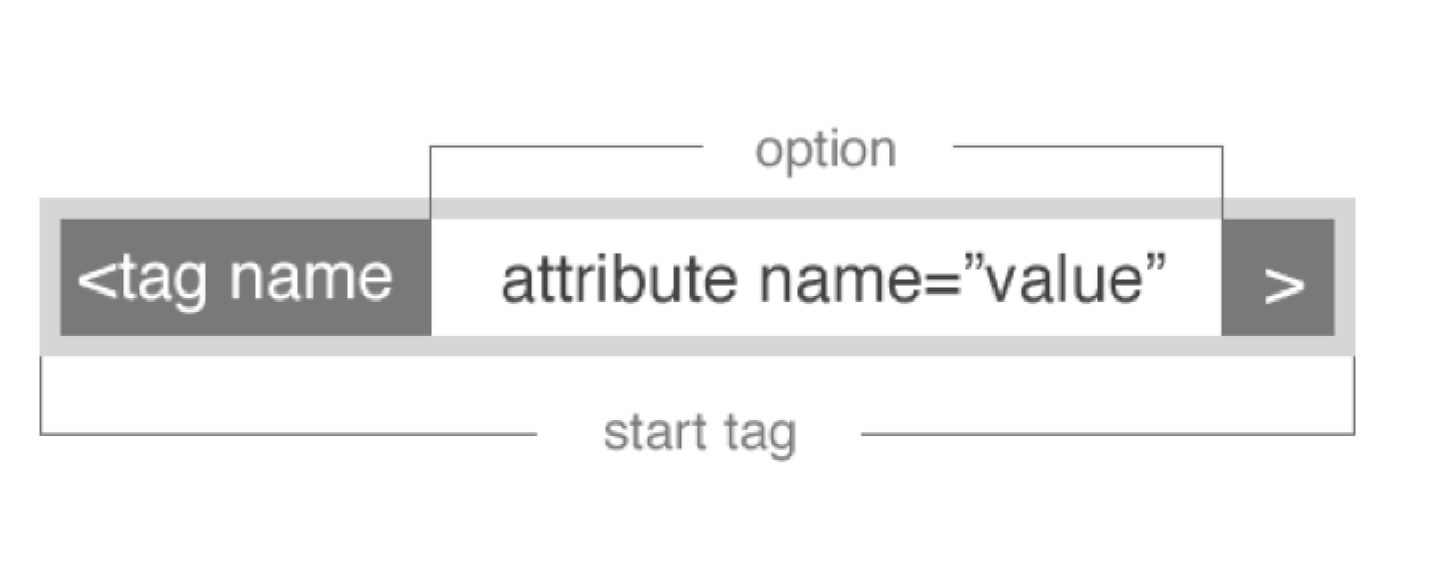


Not all elements need a closing tag:

```
<img src="/img/kittens.jpg">

<br>
```

TAG ATTRIBUTES



Common attributes include:

`id class style href src data`

BASIC DOCUMENT STRUCTURE

```
<!DOCTYPE html> - tells browser what type of document

<html>

    <head> - document metadata container
    </head>

    <body> - on screen container
    </body>

</html>
```

DOCUMENT LEVEL TAGS

```
<head>
    <meta charset="utf-8">
    <title>Some web page</title>
    <link rel="stylesheet" href="css/style.css">
</head>
```

- The tags inside the head
- Not visible on the page
- Contain information about the document
- eg. `meta`, `title`, `link`

CONTENT TAGS

- Headings
- Text
- Lists
- Links

HEADINGS

```
<h1>Largest Heading</h1>
<h2>...</h2>
<h3>...</h3>
<h4>...</h4>
<h5>...</h5>
<h6>Smallest Heading</h6>
```

TEXT

```
<p>This is a paragraph</p>
<code>This is some computer code</code>
<span>This is a some text</span>
```

LISTS

```
<ul>
    <li>First list item</li>
    <li>Second</li>
    <li>Third</li>
</ul>
```

- This is an `unordered list` example.
- Note the indentation for child elements.

LINKS

```
<a href="Link">First item</a>
```

- A hyperlink is actually called an anchor tag.
- It can link to:
    - another section within the same page
    - another page on your site
    - a page on another site

```
<a href="#section2">Jump to section 2</a>

<a href="/some/other/page">See more</a>

<a href="http://website.com">Great site</a>
```

SEMANTIC VS NON-SEMANTIC

- Semantic elements give structural meaning to the page

  (eg. `nav`, `header`, `footer`, `section`, `h1`, etc)
- Non-semantic elements don't

  (eg. `div`, `span`)
- Give your page meaning where possible

HOW DO WE ADD STYLE TO HTML?

WE LINK A STYLE SHEET!

- A style sheet is a set of rules that tell the browser how to decorate elements and element content.
- Style sheets can be embedded in the html page (bad) or linked as an external file (good).
- External Style Sheets are linked in the <head> as:

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

PUTTING IT ALL TOGETHER

```
<!DOCTYPE html>
<html>
    <head>
        <meta charset="utf-8">
        <title>Some web page</title>
        <link rel="stylesheet" href="css/style.css">
    </head>
    <body>
        <h1>Heading text here</h1>
        <p>Paragraph text here with a
            <a href="LinkUrl">Link</a></p>
    </body>
</html>
```

LET'S BUILD A HTML WEB PAGE!

IN 10 SIMPLE STEPS

INTRODUCING CSS

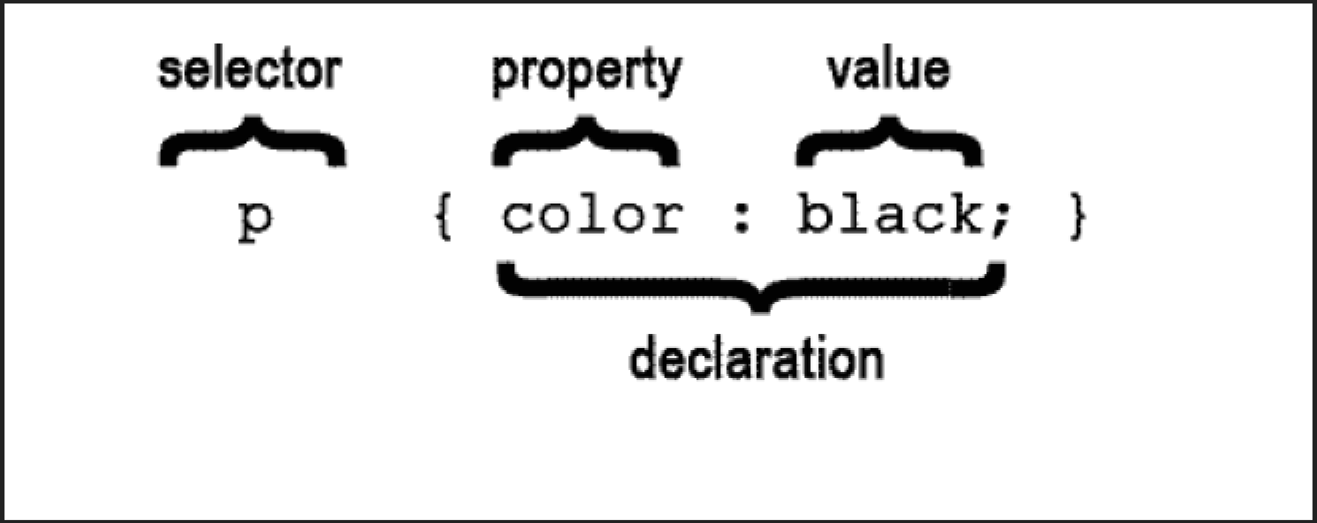
WHAT IS CSS?

CSS = style (skin/makeup)

CASCADING STYLE SHEETS

Styles have an affect on the look, the layout, and the positioning of elements.

CSS RULE(S)!



```
p   { color : black; }
```

This whole thing is called a **rule**.

The browser reads the rules, finds the elements that match the rule, and renders the element using the rules properties.

CSS SELECTOR

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

The **selector**, p in this case, specifies what parts of the HTML document should be styled by the declaration.

This selector will style all p elements on the page.

THE DECLARATION BLOCK

```
{
    color: red;
    font-weight: bold;
}
```

Declarations go inside curly braces.

Every declaration is a **property** followed by a **value**, separated by a colon, ending in a semicolon.

This example has two declarations.

THE CSS CASCADE

CSS reads top to bottom.

If conflicting then last overrides all previous.

```
p {
    background: blue;
    font-size: 16px;
}
p {
    background: orange;
}
```

SPECIFICITY

More specific rules override less specific rules

```
body {background: orange;}
p {background: blue;}

<body>
    <p>Paragraph text here.</p>
</body>
```

INHERITANCE

Child elements inherit rules from parent elements

```
body {font-family: sans-serif;}
p {font-family: Georgia, serif;}

<body>
    <p>Paragraph text here.</p>
</body>
```

p will inherit body font-family if we don't set it

IMPORTANCE

Order of Selector Importance (from least to most):

- Type (eg. `p`, `div`)
- Class (eg. `.example`)
- ID (eg. `#example`)

CSS COLORS

Colors can be specified in CSS in a variety of ways:

- keywords `white`, `black`
- hex codes `#6756A7`, `#FFF`, `#000`
- rgb `rgb(0,0,0)`
- rgba `rgba(12, 78, 200, 0.7)`
- hsl `hsl(0, 100%, 50%)`
- hsla `hsla(0, 100%, 50%, 0.5)`

BUILDING RESPONSIVELY

Hello `media` queries!

```
/* Phones  ----------- */
@media screen and (max-width: 767px) {

}

/* Tablet/Desktop  ----------- */
@media screen and (min-width: 768px) {

}
```

LET'S ADD SOME STYLES TO OUR STYLE SHEET

IN 10 SIMPLE STEPS

GETTING YOUR SITE ON THE INTERNET

1. Find a web host (eg. Crazy Domains, Dreamhost, etc)
2. Buy a domain name

   Many hosting companies offer a free domain when you sign up for hosting

3. Download an FTP program (eg. FileZilla)

   Upload your files to your server

HOW TO MODIFY THE CODE OF EXISTING WEBSITES

- Get to know the basics
- Practice, practice, code, code
- Keep building on what you know
- Use debugging tools to edit and learn

CHROME DEBUGGER

Useful to see/edit:

- the styles applied to the elements
- the elements location in the page document structure

Let's check it out!

(Install Firebug if using Firefox)

TECHNIQUES TO USE AT-HOME WHEN STUCK ON CODE

- Make a cup of tea/coffee
- Google the problem (check the date!)
- Visit helpful coding websites (see Resources)
- Don't get intimidated
- Determine to be a problem solver
- Go for a walk (clear head and come back to it)
- Talk to coding friends

REVIEW: TAKEAWAYS

- How the Internet and web pages work
- Introducing HTML / CSS `Code along`
- Getting your site on the Internet
- How to modify the code of existing websites
- Techniques to use at-home when stuck on code

RESOURCES

Today's slides and web demo files can be found here:

https://github.com/leoniedunnett/workshop-html-css

HTML RESOURCES

- HTML Dog
- Mozilla's web guide for HTML5
- Mozilla's List of HTML elements

CSS RESOURCES

- CSS Tricks
- AtoZ CSS
- Mozilla's guide for CSS
- Mozilla's CSS reference

OTHER RESOURCES

- Can I Use
- HTML 5 Please
- HTML Validator
- CODEPEN - Testing your own code, looking at cool things other people have done.

WHERE TO FROM HERE?

The next step: Intensive Front End course

https://generalassemb.ly/education/front-end-web-development/melbourne

QUESTIONS?