

Web Programming

Web Programming

Rodrigo Martins Pagliares

Introduction to HTML

Getting started with HTML

Lecture: Getting started with HTML

Module: Introduction to HTML

Introduction

What is HTML?

HTML (HyperText Markup Language) is a **markup language** used to tell your **browser how to structure the web pages** you visit.

HTML

- **HTML is a language made up of elements.**
- **HTML can be applied to pieces of text to:**
 - ✓ **Give** them different **meaning** in a **document**.
 - Is it a **paragraph**? Is it a **bulleted list**? Is it part of a **table**?
 - ✓ **Structure** a **document** into **logical sections**.
 - Does it have a **header**? **Three columns** of **content**? A **navigation menu**?
 - ✓ **Embed content** such as **images** and **videos** into a **page**.

Objective of the lecture

Objective: To gain basic familiarity with the **HTML** language, and get some **practice** writing a few **HTML** elements.

The `<p>` element

- If we want a **line of text to stand by itself**, we can **specify** that it is a **paragraph** by enclosing it in a **paragraph element (`<p>`)**

1	My cat is very grumpy
---	-----------------------



Text content

1	<code><p>My cat is very grumpy</p></code>
---	---



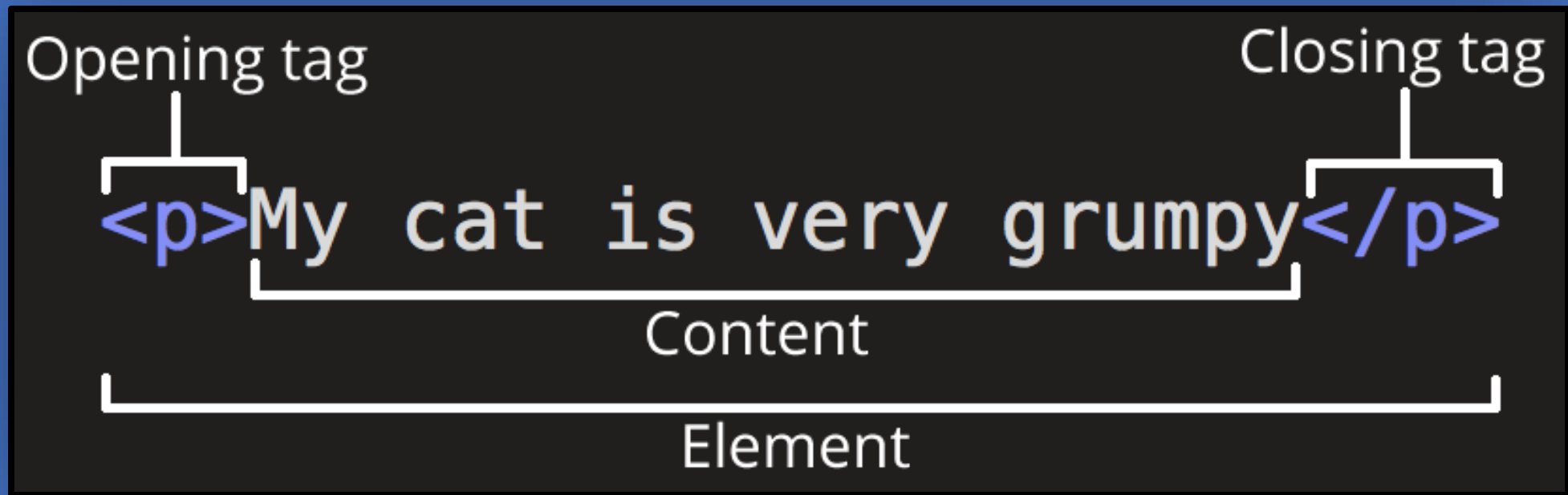
Text content “marked up”

Note: HTML tags are case-insensitive

Note: Tags in **HTML** are **case-insensitive**, i.e. they can be written in **uppercase** or **lowercase**. For example, a `<title>` tag could be written as `<title>`, `<TITLE>`, `<Title>`, `<TiTlE>`, etc., and it will work fine. **Best practice**, however, is to write all tags in **lowercase** for consistency, readability, and other reasons.

Anatomy of an HTML element

Anatomy of an HTML element



Active learning: Creating your first HTML element

Go to the link indicated below for the Active Learning statement

Active learning: creating your first HTML element

Edit the line below in the "Editable code" area by wrapping it with the tags `` and ``. To *open the element*, put the opening tag `` at the start of the line. To *close the element*, put the closing tag `` at the end of the line. Doing this should give the line italic text formatting! See your changes update live in the *Output* area.

If you make a mistake, you can clear your work using the *Reset* button. If you get really stuck, press the *Show solution* button to see the answer.

Live output

This is my text.

Editable code

Press Esc to move focus away from the code area (Tab inserts a tab character).

```
This is my text.
```

Reset

Show solution

<https://mzl.la/2OSmQtO>

Nesting elements

1	<code><p>My cat is very grumpy.</p></code>
---	---



1	<code><p>My cat is very grumpy.</p></code>
---	---



Block versus inline elements

- **Two important categories of elements in HTML:**
 - ✓ **Block-level elements**
 - ✓ **Inline elements**
- **Block-level elements form a visible block on a page.**
- **Inline elements are those that are contained within block-level elements and surround only small parts of the document's content**
 - ✓ **Not entire paragraphs and groupings of content.**

Block versus inline elements

1	<code>firstsecondthird</code>
2	
3	<code><p>fourth</p><p>fifth</p><p>sixth</p></code>

`` is an inline element

firstsecondthird

fourth

fifth

sixth

1	<code>firstsecondthird</code>
2	
3	<code><p>fourth</p><p>fifth</p><p>sixth</p></code>

`<p>` is a block-level element

Block versus inline elements

Note: HTML5 redefined the element categories: see [Element content categories](#). While these definitions are more accurate and less ambiguous than the ones that went before, they are a lot more complicated to understand than "**block**" and "**inline**", so we will stick with these throughout this topic.

Empty elements

Not all elements follow the pattern of an opening tag, content, and a closing tag. Some elements consist only of a single tag, which is usually used to insert/embed something in the document at the place it is included.


```
1 
```



Attributes

Attributes

The **class attribute** allows you to give the **element** an identifying **name**, that can be **used later** to **target** the **element** with **style information** and other **things**.



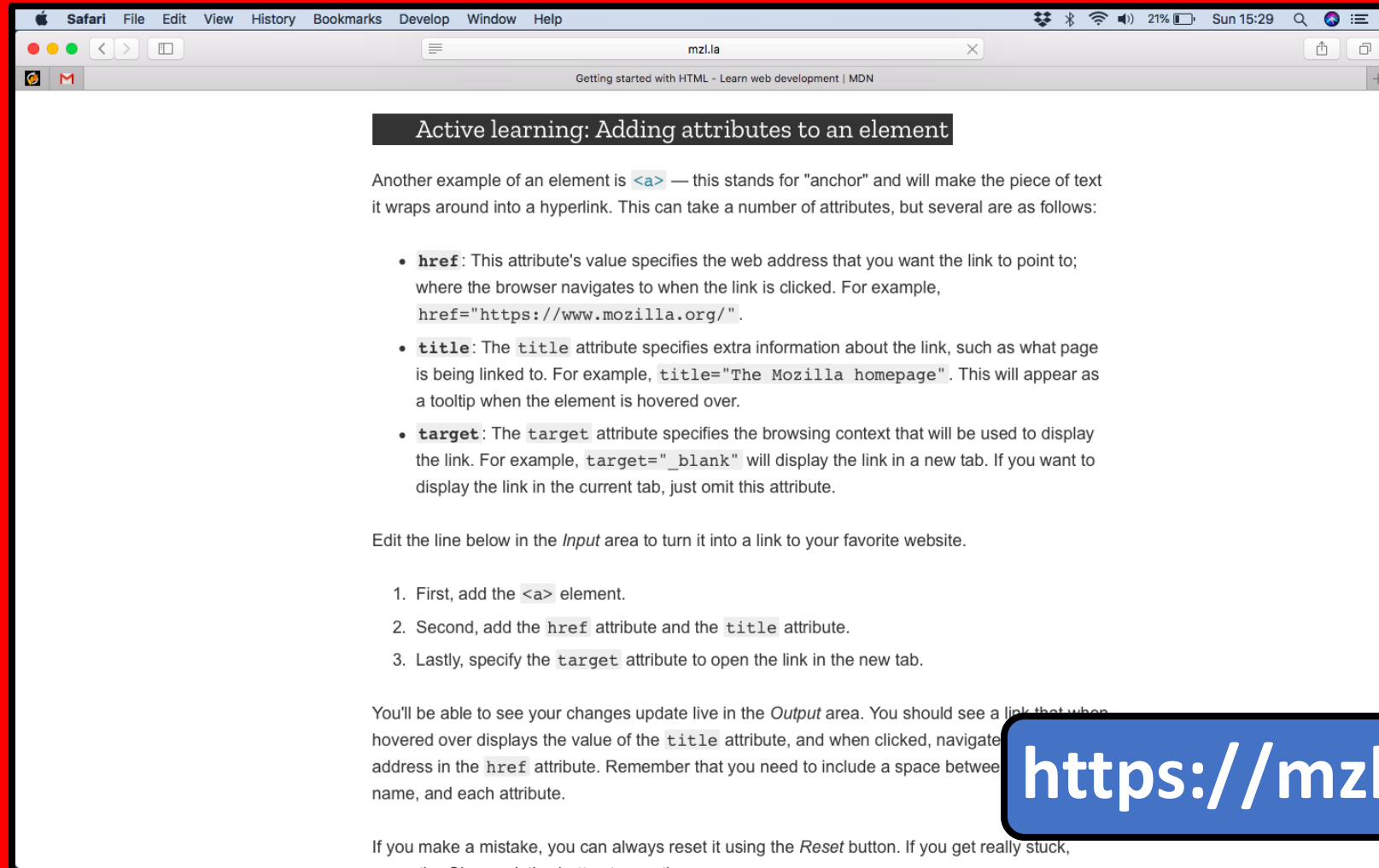
Attribute

```
<p class="editor-note">My cat is very grumpy</p>
```

Attributes contain **extra information** about the **element** that you **don't want** to appear in the **actual content**.

Active learning:
Adding attributes to an element

Go to the link indicated below for the Active Learning statement



Active learning: Adding attributes to an element

Another example of an element is `<a>` — this stands for "anchor" and will make the piece of text it wraps around into a hyperlink. This can take a number of attributes, but several are as follows:

- **href**: This attribute's value specifies the web address that you want the link to point to; where the browser navigates to when the link is clicked. For example, `href="https://www.mozilla.org/"`.
- **title**: The `title` attribute specifies extra information about the link, such as what page is being linked to. For example, `title="The Mozilla homepage"`. This will appear as a tooltip when the element is hovered over.
- **target**: The `target` attribute specifies the browsing context that will be used to display the link. For example, `target="_blank"` will display the link in a new tab. If you want to display the link in the current tab, just omit this attribute.

Edit the line below in the *Input* area to turn it into a link to your favorite website.

1. First, add the `<a>` element.
2. Second, add the `href` attribute and the `title` attribute.
3. Lastly, specify the `target` attribute to open the link in the new tab.

You'll be able to see your changes update live in the *Output* area. You should see a link that when hovered over displays the value of the `title` attribute, and when clicked, navigate address in the `href` attribute. Remember that you need to include a space between name, and each attribute.

If you make a mistake, you can always reset it using the *Reset* button. If you get really stuck,

<https://mzl.la/2wfpcwd>

Go to the link indicated for the Active Learning statement

<https://mzl.la/2wfp cwd>

Active learning: Adding attributes to an element

Another example of an element is `<a>`. This stands for *anchor*. An anchor can make the text it encloses into a hyperlink. Anchors can take a number of attributes, but several are as follows:

- **href**: This attribute's value specifies the web address for the link. For example:
`href="https://www.mozilla.org/"`.
- **title**: The `title` attribute specifies extra information about the link, such as a description of the page that is being linked to. For example, `title="The Mozilla homepage"`. This appears as a tooltip when a cursor hovers over the element.
- **target**: The `target` attribute specifies the browsing context used to display the link. For example, `target="_blank"` will display the link in a new tab. If you want to display the linked content in the current tab, just omit this attribute.

Edit the line below in the *Input* area to turn it into a link to your favorite website.

1. Add the `<a>` element.
2. Add the `href` attribute and the `title` attribute.
3. Specify the `target` attribute to open the link in the new tab.

You'll be able to see your changes update live in the *Output* area. You should see a link—that when hovered over—displays the value of the `title` attribute, and when clicked, navigates to the web address in the `href` attribute. Remember that you need to include a space between the element name, and between each attribute.

If you make a mistake, you can always reset it using the *Reset* button. If you get really stuck, press the *Show solution* button to see the answer.

Live output

A link to my favorite website.

Boolean attributes

The `disabled` **attribute**, which you can **assign** to **form input elements**, if you **want** them to be **disabled** (greyed out) so the **user can't enter** any **data** in them.

1	<code><input type="text" disabled="disabled"></code>
---	--

OR

1	<code><input type="text" disabled></code>
---	---

The disabled attribute of the <input> element

```
1 <!-- using the disabled attribute prevents the end user from entering text into the input box -->
2 <input type="text" disabled>
3
4 <!-- The user can enter text into the follow input, as it doesn't contain the disabled attribute -->
5 <input type="text">
```

Disabled text field (grey out)

Enabled text field

Omitting quotes around
attribute values

Always include attribute quotes !

```
1 <a href=https://www.mozilla.org/>favorite website</a>
```

No problem !



```
1 <a href=https://www.mozilla.org/ title=The Mozilla homepage>favorite website</a>
```



The browser will misinterpret your markup, thinking that the **title attribute** is **actually three attributes** — a **title attribute** with the value "The", and **two boolean attributes**, **Mozilla** and **homepage**.

Single or double quotes?

The use of single or double quotes is purely a **matter of style**, and you can **feel free to choose** which one you **prefer**,

1	<code>A link to my example.</code>
2	
3	<code>A link to my example.</code>



but **don't mix them together !**

1	<code>A link to my example.</code>
---	--



Right or Wrong?

1 A link to my example.



1 A link to my example.



1 A link to my example.



Anatomy of an HTML element

Anatomy of an HTML document

```
01 <!DOCTYPE html>
02 <html>
03   <head>
04     <meta charset="utf-8">
05     <title>My test page</title>
06   </head>
07   <body>
08     <p>This is my page</p>
09   </body>
10 </html>
```

We'll look at how **individual elements** are **combined** to form an entire **HTML page**

Active learning:

Adding some features to an HTML document


Go to the **link**
indicated for
the **Active**
Learning
statement

<https://mzl.la/3hylYq3>

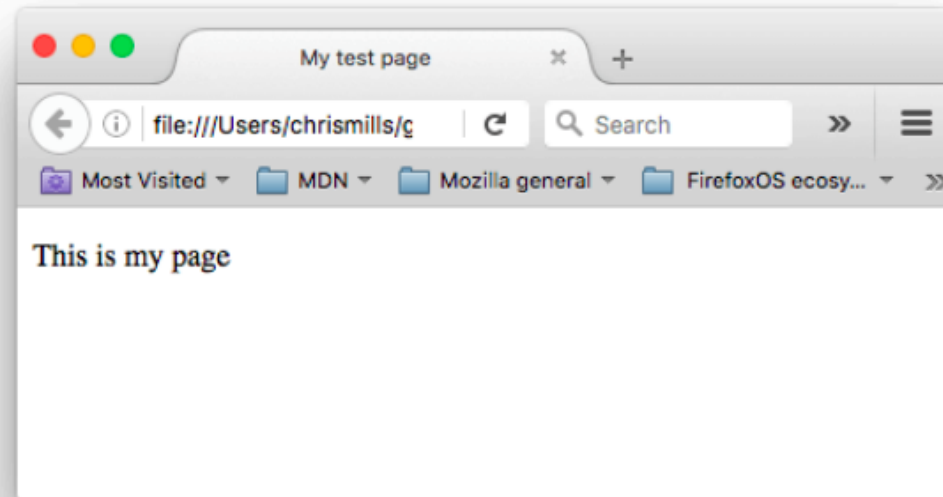
Active learning: Adding some features to an HTML document

If you want to experiment with writing some HTML on your local computer, you can:

1. Copy the HTML page example listed above.
2. Create a new file in your text editor.
3. Paste the code into the new text file.
4. Save the file as `index.html`.

Note: You can also find this basic HTML template on the [MDN Learning Area Github repo](#) .

You can now open this file in a web browser to see what the rendered code looks like. Edit the code and refresh the browser to see what the result is. Initially the page looks like this:



Whitespaces in HTML

- Why use so much whitespace?*
- The **two** following **code snippets** are **equivalent**

01	<code><p>Dogs are silly.</p></code>
02	
03	<code><p>Dogs are</code>
04	<code> silly.</p></code>

- No matter how much whitespace you use (which can include space characters, but also line breaks), the HTML parser reduces each one down to a single space when rendering the code.

* The answer is **readability**

**Entity references: including
Special characters in HTML**

Including special characters in HTML

Literal character	Character reference equivalent
<	<
>	>
“	"
‘	'
&	&

Including special characters in HTML

1	<code><p></code> In HTML, you define a paragraph using the <code><p></code> element. <code></p></code>
2	
3	<code><p></code> In HTML, you define a paragraph using the <code>&lt;p&gt;</code> element. <code></p></code>

In HTML, you define a paragraph using the
element.

In HTML, you define a paragraph using the `<p>` element.

Note: A chart of all the available HTML character entity references can be found on Wikipedia: Note that you don't need to use entity references for any other symbols, as modern browsers will handle the actual symbols just fine as long, as your HTML's character encoding is set to UTF-8.

The screenshot shows a web browser window displaying the Wikipedia article "List of XML and HTML character entity references". The browser's address bar shows the URL "https://en.wikipedia.org/wiki/List_of_XML_and_HTML_character_entity_references". The Wikipedia logo and navigation menu are visible on the left. The article title is "List of XML and HTML character entity references". Below the title, it says "From Wikipedia, the free encyclopedia". The main text explains that in SGML, HTML and XML documents, logical constructs known as *character data* and *attribute values* consist of sequences of characters, which can manifest directly or be represented by a series of characters called a *character reference*. It lists the character entity references valid in HTML and XML documents. A character entity reference refers to the content of a named entity. An entity declaration is created by using the `<!ENTITY name "value">` syntax in a Document Type Definition (DTD). A table of contents is provided, listing sections from "Character reference overview" to "External links". A sidebar on the right lists various HTML-related topics under the heading "HTML", including "Dynamic HTML", "XHTML", "Character encodings", "Language code", "Document Object Model", "Browser Object Model", "Style sheets (CSS)", "Font family", "Web colors", "HTML scripting", "JavaScript", "WebGL", "WebCL", "W3C (Validator)", "WHATWG", "Quirks mode", "Web storage", "Rendering engine", "Comparisons", "Document markup languages", "HTML support", and "XHTML (1.1)".

https://bit.ly/38hoiwp

HTML comments

HTML comments

- **Comments** are **ignored** by the **browser** and are **invisible** to the **user**.
- **The purpose of comments** is to **allow** you to include **comments** in the **code** to say **how** your **code** **works**, **what** the **different parts** of the code **do**, and so on.

1	<p>I'm not inside a comment</p>
2	
3	<!-- <p>I am!</p> -->

I'm not inside a comment

Questions?