HTML TAGS, ATTRIBUTES AND ELEMENTS

APRIL 2022

YONATAN BENEZRA

HTML Tags

During the previous lessons we have been writing HTML.
We even created entire web pages all by ourselves!

But how did we create these web pages, and what exactly did we create?

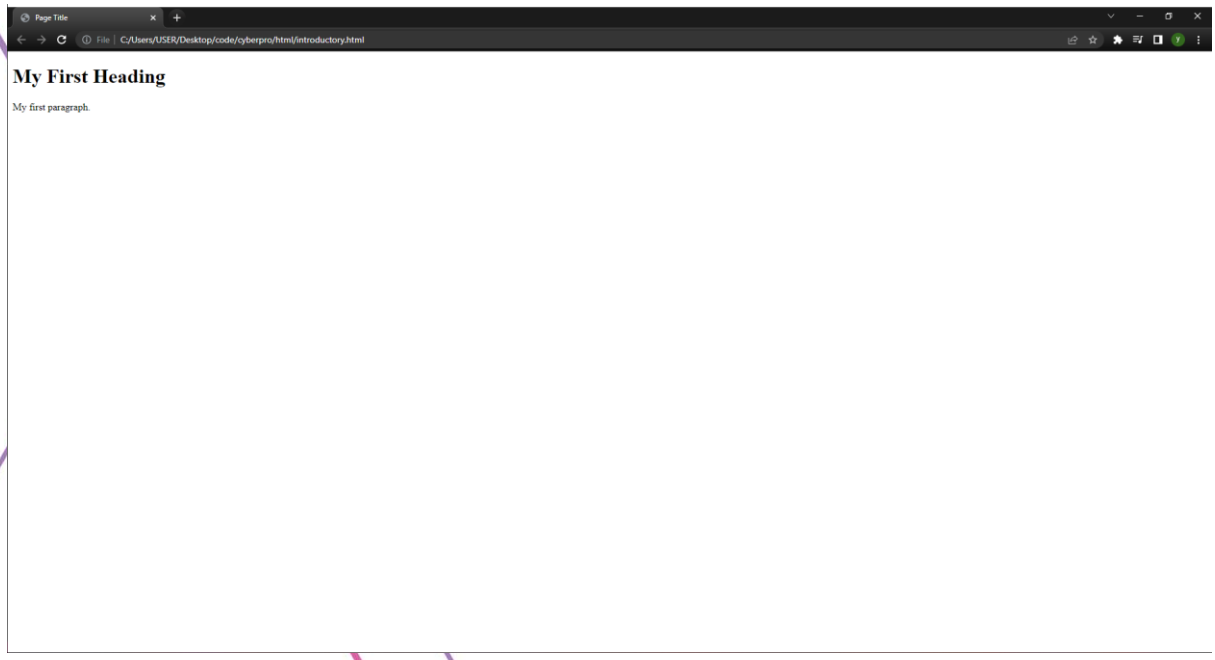
We now know that the following:

```
<!DOCTYPE html>
<html>
<head>
<title>Page Title</title>
</head>
<body>

<h1>My First Heading</h1>
<p>My first paragraph.</p>

</body>
</html>
```

Will turn out to look as follows:



We will now discuss how this process works.
Let's first begin with several ground rules for HTML.

- All HTML documents must start with a document type declaration: <!DOCTYPE html>.
- The HTML document itself must begin with <html> and end with </html>.
- The visible part of the HTML document will be between <body> and </body>.
- Every word that starts with “<” and ends with “>” is an HTML tag or element.

Nodes and Elements

Inside HTML, everything is a node. For example:

- Comments
- Text
- Elements

An element is made up of an "opening tag" and a "closing tag". For example, a "paragraph" tag will use the "p" element.

```
<p>Here will be the text inside of this paragraph.</p>
```

In the above example, we have:

- An opening tag, which is the "p" element
- A text which is a node
- And a closing tag.

How many elements are in the below code? What are their tags?

```
<div>
  <h2>I am a header</h2>
  <p>it's cozy here inside of a div</p>
</div>
```

Highlight the colored text below for the answer:

There are 3 elements: A "div", inside of it an "h1" element, and next to it a "p" element.

Attributes

We can give each element a specific attribute, like an id:

```
<p id="title">presentense</p>
```

We added an ID attribute to a "p" element inside of the opening tag, which lets us identify each element. This doesn't change anything about the element, but some attributes can change the functionality or effect of the element, and some even require certain attributes.

Another common identifier attribute is class. The difference between the two is that the same Class should be given to more than one element, and each element can be given more than one class. This is something that cannot happen with the ID attribute, as there cannot be more than one element with the same ID).

To give an element more than one class, just separate them with a space, like so:
my classes are "paragraph" and "element" .
Let's review some other tags.

Tags

Each tag has different use cases.
Let's discuss some of the most popular tags we can use in HTML.

- **IMG tag**
  The IMG (image) tag is one of those elements that require a certain attribute. To show an image we need to enter the image link inside of an "src" attribute, as follows:

Notice anything strange? This element does not have a closing tag! Most of the elements don't need a closing tag, so that's cool.

HTML tag list
At this point, the most useful tags will be:

- **div**
  - h1-h6
  - p
  - ul/ol&li
  - input
  - button

div

The div tag is a block element. It will help us organize elements on the page and style them later on with CSS. We can nest any element within a div. Go ahead and try!

```
<div>

 <div>nested div</div>

 <span>nested span</span>

 <p>nested p</p>

</div>
```

Nesting the elements will help us better style our elements!

Headers

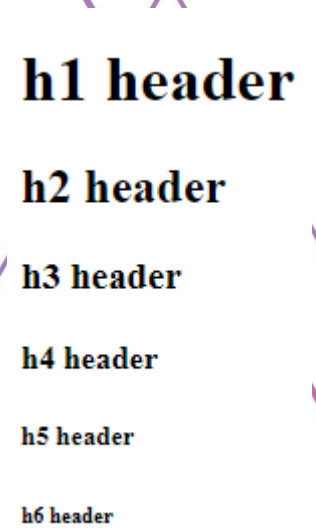
The h tag is a block element. They let us display text in different sizes in an easy way.
Here is a code example that will demonstrate how headers work:

```
<!DOCTYPE html>
<html>
<head>
<title>Headers</title>
</head>
<body>

<h1>h1 header</h1>
<h2>h2 header</h2>
<h3>h3 header</h3>
<h4>h4 header</h4>
<h5>h5 header</h5>
<h6>h6 header</h6>

</body>
</html>
```

It will look as follows:



You can play with the code at this link.

p

The p tag is a block element.
It should not contain any other nested element, or anything else besides text.

Input

We use inputs to input data that we can later use.
There are two important attributes that we will discuss:

- Placeholder - a text which will be displayed until the user types something else inside.
- type - the type of input allowed for use, which can be text, number...

Example:

```
<input type="text" placeholder="name">
```

Which will look as follows:

Button

The button tag will operate as a pre-styled button which will allow the user to interact with our page. As an example, let's create a “buy” button.

```
<button>buy</button>
```

It will look like this:

It will not do anything at this time, because we did not add any logic, which we will learn how to do later on.

You cannot be sure how HTML will be displayed.
Larger or smaller screens and resized windows will create different results.

With HTML, you cannot change the display by adding extra spaces or extra lines in your HTML code.

The browser will automatically remove any extra spaces and lines when the page is displayed,
For example, the following code:

```
<p>
This paragraph
contains a lot of lines
in the source code,
but the browser
ignores it.
</p>

<p>
This paragraph
contains          a lot of spaces
in the source           code,
but the     browser
ignores it.
</p>
```

Will appear as follows:

This paragraph contains a lot of lines in the source code, but the browser ignores it.

This paragraph contains a lot of spaces in the source code, but the browser ignores it.

We can use different HTML elements in order to tell the browser how we want the code to appear. Let's discuss two elements that can help with this:

HR

The <hr> tag defines a thematic break in an HTML page and is most often displayed as a horizontal rule.
The <hr> element is used to separate content (or define a change) in an HTML page.
For example, the following code:

```
<p>This is some text.</p>
<hr>
<h2>This is heading 2</h2>
```

will look like this:

This is some text.

This is heading 2

Break

In order to skip a line, we can use the "
" element.
This will break the paragraph and move down one line.

For example:

```
<p>
    Lorem Ipsum is simply dummy text of the printing and typesetting industry.
    Lorem Ipsum has been the industry's standard dummy text ever since the
    <br />
    unchanged. It was popularized in the 1960s with the release of Letraset
    sheets containing Lorem Ipsum passages, and more recently with desktop
  </p>
```

the following code will look like this:

Lorem Ipsum is simply dummy text of the printing and typesetting industry. Lorem Ipsum has been the industry's standard dummy text ever since the unchanged. It was popularised in the 1960s with the release of Letraset sheets containing Lorem Ipsum passages, and more recently with desktop

Notice how it went down to the line below after the word "the"?

Structure & Elements

Thanks to HTML elements, we can control the layout of a webpage. Let's try it out!
Copy the below block of code into your HTML file and see how it looks.

```
<h1>h1 is the largest-sized heading</h1>
<p>this is a "p" element, mostly it will contain long text</p>

<h2>this is an h2</h2>
<p>This is a subheading, it is about the sizes of headings.
the largest heading element is h1, as the number of the heading is bigger, the size of the font is smaller</p>

<h3>for instance, this is an h3</h3>
<p>cool.</p>
```

Block Elements:

The elements we used above are block elements, which means they take up the entire width of their parent element. Each of these elements is a child of the body element.
The width of the body element is the whole page, so the width of all its children is the whole page too!

Inline Elements:

Inline elements only take up as much space as they need. Let's see an example of an inline element. Copy this text to your body and pay attention to the width of the span element.

```
<p>I take up all the width of my parent <span> I take up space only as much as needed because I'm an inline element.</span> </p>
```