# HTML5 Authoring with Mark Lassoff

## Section 5: Using `<div>` and `<span>` Elements

`<div>` and `<span>` tags give us a convenient method to isolate parts of our content for formatting or access via JavaScript. Keep in mind that the `<div>` and `<span>` tags are not as semantic as some of the HTML5 text markup tags introduced earlier, but, they are very convenient-- and it's their convenience that likely keeps them in heavy use today.

Complex HTML5 documents may contain many different types of content and creating `<div>`s is the best way to stay organized. The element provides with a flexible organizational tool for your content that can also provide structure to your document.

After completing this section you will:

- ☐ Understand the Difference Between in-line and Block Elements
- ☐ Apply the Correct Syntax for Logical Divisions and Spans
- ☐ Use Classes and Id's
- ☐ Use Classes and Id's to Apply CSS to Divs and Spans

### Watch This: HTML5 Section 005 Video

As always your course videos are available on YouTube, Roku, and other locations. However, only those officially enrolled have access to this course guide, can submit assignments, work with the instructor, and get this guide.

Watch the section video at https://www.youtube.com/watch?v=crZKwIVRjHc

## Inline and Block Elements

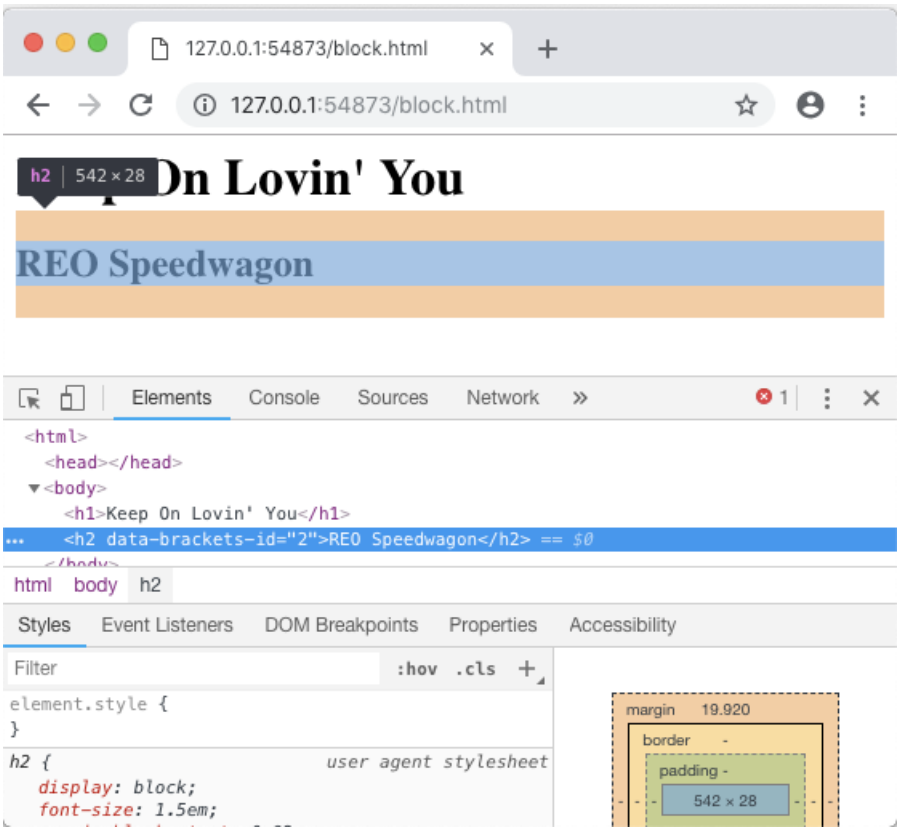
According to the default style sheet that lives in your browser most elements are designated as "inline" or "block." This setting, just like just about everything else that appears in the browser, can be altered with the use of CSS. It's still important to understand the difference between inline and block elements and know which category common elements fall into.

The image below illustrates the concept.


Block elements stretch across the width of their parent by default. Inline elements are only as wide as the content requires. For example, if you were to code the following:

```
<h1>Keep On Lovin' You</h1>
<h2>REO Speedwagon</h2>
```

Each element would appear in its own block-- because all heading tags are block elements. You can see this illustrated using the Developer tools in Chrome.



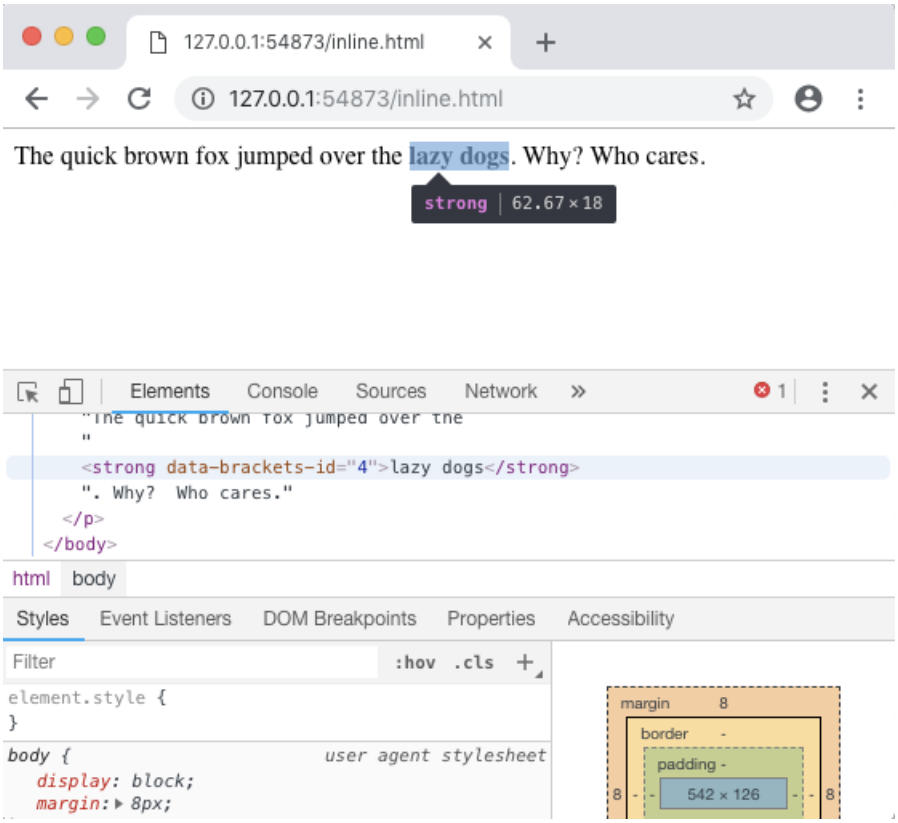

The block for the `<h2>` tag here is in blue. The orange represents additional margin space for the block. Note the block stretches all the way across the browser window.

If we were to look at an inline tag like `<strong>` the result would be different.

Let's say we had the following code:

```
<p>The quick brown fox jumped over the
<strong>lazy dogs</strong>. Why?  Who cares.</p>
```

If we were to take a look at the result with Chrome developer tools:




Typical for inline elements, the element only takes the space required by the content.

## `<div>` and `<span>` Elements

The `<div>` and `<span>` elements came along with the first big revision of HTML in the late nineties. When CSS was put into heavy use, it became apparent that developers needed ways to put sections in their documents and isolate those sections for CSS or even JavaScript.

Let's examine a code snippet:

You'll notice that the `<span>` elements are usually isolating a word or two while the `<div>` elements are usually encompassing some additional elements.

### Two Quick Tricks

Don't you love quick tricks? Here is a couple that I use for `<div>` tags.

#### Getting out of Closing Div Hell

One problem with more complex websites is that often many divs close in a row. The code often looks like this:

```
        </div>
        </div>
    </div>
    </div>
```

This can be confusing if you're trying to debug-- Or just understand your code. I like to put a quick remark next to each div closing tag so I know exactly what I'm closing. I'll use the single-line comment structure to do so as demonstrated in the code below.

```
        </div>   <-- Closing Hero Image -->
        </div>     <-- Closing Hero Box Header -->
    </div>   <-- Closing Main Header Area -->
    </div>   <-- Closing Container -->
```

Using this method, when I review the code I always know which logical division is being closed.

**Hiding Divs in Your Text Editor**

If you're using Brackets (most other text editors have a way to do do this as well), you'll notice a triangle next to our `<div>` tags. In the screenshot below, you'll see it next to line 27:


This little triangle allows you to hide and restore the `<div>` element. This is designed to hide unnecessary content in your text editor temporarily.

## Understanding Classes and Id's

Taking a second look at the code segement you'll notice that the `<span>` and `<div>` tags are paired with `id` or `class` attributes.

```
<h1><span class="frame">Framework</span> Television</h1>
       <div id="description">
           <p><span class="first">Learn digital skills
           like coding, design and app development with
           <span class="frame">Framework</span> Television.
           </span></p>

           <p>Digital skills are the "readin', writin' and
            arithmetic" of the current age. Without these
             skills you're limited in the job market.
             <span class="frame">Framework</span> helps you
             develop digital skills through television
             programming that is instructional, innovative
             and interesting.</p>

           <p>You can watch <span class="frame">Framework
           </span> anywhere you watch video-- The Web,
           your Smart TV, or streaming media device.</p>
       </div>
```

The `id` and `class` attributes are used to identify and/or group your logical divisions and spans. In the code `<div id="description">`, the id uniquely identifies the logicial division. In fact, in well coded HTML the value `description` could not be repeated in any other `id.` It's meant to be a unique identifier.

A `class,` on the other hand, is meant to create a group of elements. You might create a `class` because you want a number of `<span>` elements to be formatted the same way. What you'll notice in the code above the class `frame` is used repeatedly in the code.

## Applying CSS to `<div>` and `<span>` Elements

Let's expand the code snippet above to a formal page, including CSS.

```
<!DOCTYPE html>
<html>
    <head>
        <title>Div and Span</title>
        <style>
            #description
            {
                background-color: rgb(230,230,230);
                border: 2px solid rgb(0,0,150);
                padding: 5px;
            }

            .frame
            {
                color: rgb(0,0,150);
                font-weight: bold;
            }

            .first
            {
                font-family: Arial;
            }
        </style>
    </head>
    <body>
        <h1><span class="frame">Framework</span>
        Television</h1>
        <div id="description">
            <p><span class="first">Learn digital skills like
            coding, design and app development with <span
            class="frame">Framework</span> Television.</span>
            </p>

            <p>Digital skills are the "readin', writin' and
            arithmetic" of the current age. Without these
            skills you're limited in the job market. <span
            class="frame">Framework</span> helps you develop
            digital skills through television programming that
            is instructional, innovative and interesting.</p>

            <p>You can watch <span class="frame">
            Framework</span> anywhere you watch video-- The
            Web, your Smart TV, or streaming media
            device.</p>
        </div>
    </body>
</html>
```

Looking carefully at the CSS you'll notice that we use different selectors than in previous examples. If you want to style an element with an `id` attribute you use a `#` sign and the value of the id as in `#description`.

Any CSS will be applied to the logical division id'ed as description.

Similarly `class` level CSS is identified with a period and the name of the class as in `.frame` which will be applied to multiple elements in this case.

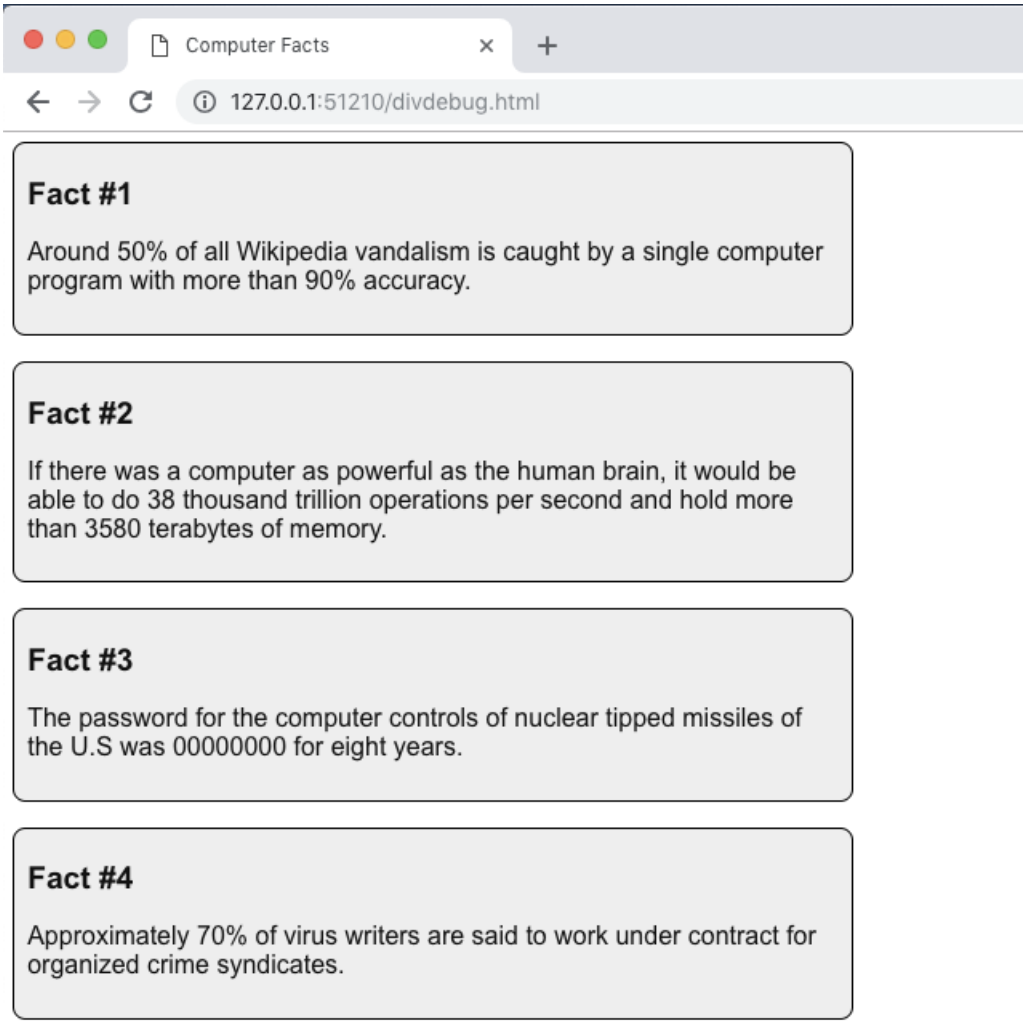
## Debug This: Facts

Note that in this script there may be two types of errors: (A) Errors that effect that output and (B) Errors that, while they do not effect the output of the document, should still be corrected. Good luck!

```
<!DOCTYPE html>
<html>
    <head>
        <title>Computer Facts</\title>
        <style>
            #fact
            {
                width: 40%;
                border: 1px solid black;
                background-color: #eee;
                color: #121212;
                padding: 10px;
                margin-bottom: 20px;
                border-radius: 10px;
            }

            hI
            {
                font-family: Arial;
                font-size: 1.5em;
            }

            p
            {
                font-family: Arial;
                font-size: 1.25em;
            }

        </style>
    </head>
    <body>
        <div class="fact">
            <h1>Fact #1</h1>
            <p>Around 50% of all Wikipedia vandalism is
            caught by a single computer program with more than
            90% accuracy.</p>
        </div> <!-- fact -->
        <div class="fact">
```

```
            <h1>Fact #2</h1>
            <p>If there was a computer as powerful as the
            human brain, it would be able to do 38 thousand
            trillion operations per second and hold more than
            3580 terabytes of memory.</p>
        </div> <!-- fact -->
        <div class="fact">
            <h1>Fact #3</h1>
            <p>The password for the computer controls of
            nuclear tipped missiles of the U.S was 00000000
            for eight years.</p>
        </div> <!-- fact -->
        <div class="fact>
            <h1>Fact #4</h1>
            <p>Approximately 70% of virus writers are said to
            work under contract for organized crime syndicates.
            </p>
        </div> <!-- fact -->
    </body>

</html>
```

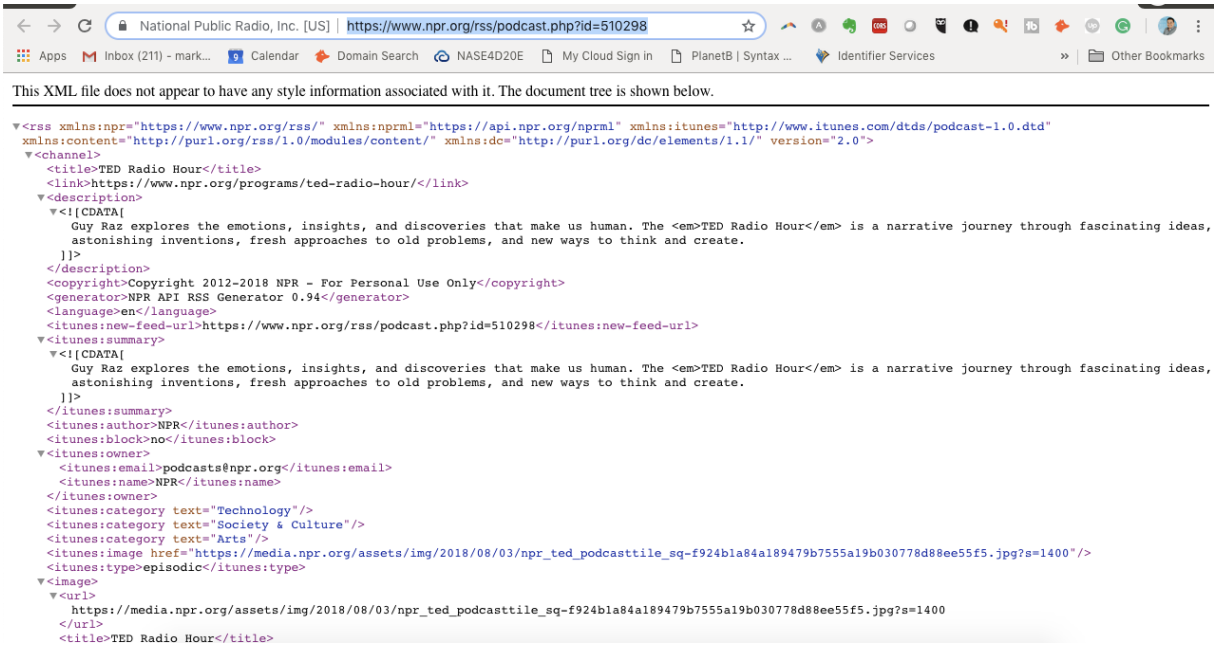
Here's what the result should look like:




## Submit This: My Friend Ted

The XML feed for the Ted Radio Hour from NPR appears at https://www.npr.org/rss/podcast.php?id=510298.

When you click that link your browser should open to a page of code like this:




XML (eXtensible Markup Language) is similar in structure to HTML, but its meant to carry data in a non-proprietary format from one system to another. , and you should be able to derive meaning from the tags and content.

Create a web page that lists the upcoming Ted Radio Hour shows. (Hint: Look at the `<item>` tags in the XML. For each show list the name of the individual show, the description, the date it was published and the copyright information.

Your page must use logical division, spans, CSS and whatever other markup you need to make it readable and usable.

Remember, when submitting the work, please use the following naming convention for your file: `HTMLAUTHORING_LastName_SectionNumber.html`. So if your last name is Smith and your submitting section 8, your file name should be `HTMLAUTHORING_Smith_8.html`.

For this course visit https://www.dropbox.com/request/RhW9kBDXtisq2Fsvg3hY to submit your assignments.