Introduction to HTML and CSS

BUILDING THE FOUNDATION OF WEB DEVELOPMENT

HyperText Markup Language

Standard markup language used to create and design web pages.

It provides a way to structure content on the web, defining elements and their relationships within a document

Surrounds Content with opening and closeing tags.

Each Tag's name is called an element

- Syntax **<element>** content **</element>**
- **<p>**My first paragraph.</**p>**

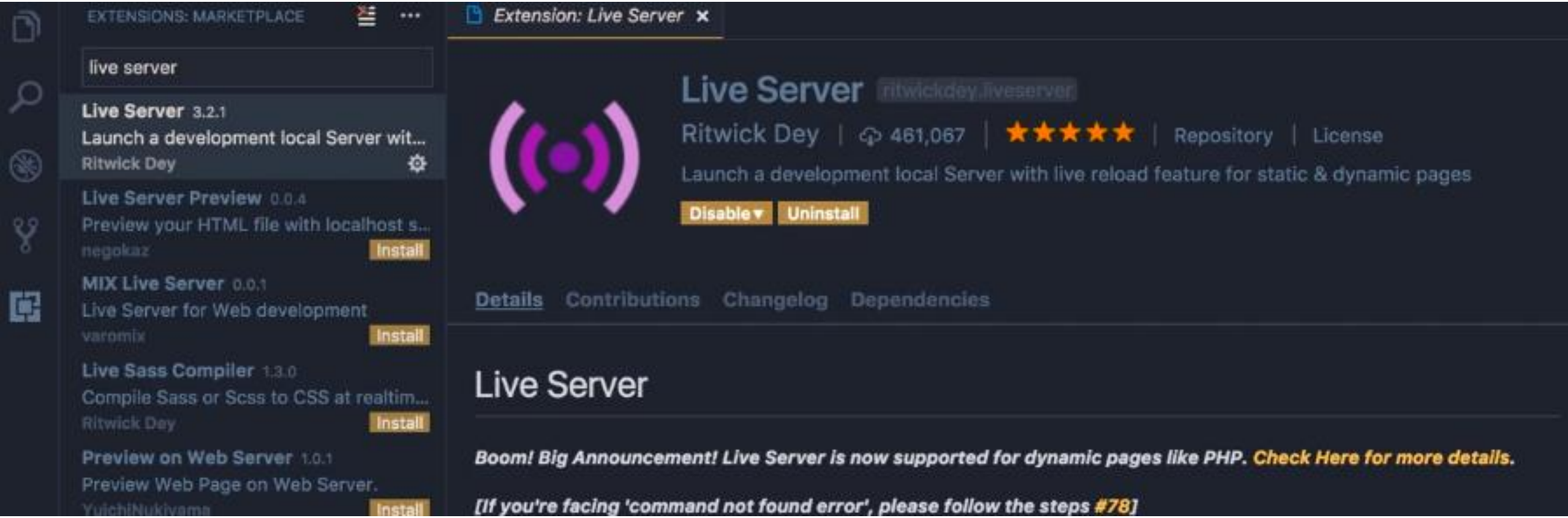
HTML Document

```
<!DOCTYPE html>
<html>
    <body>
        <h1>My First Heading</h1>
        <p>My first paragraph.</p>
    </body>
</html>
```

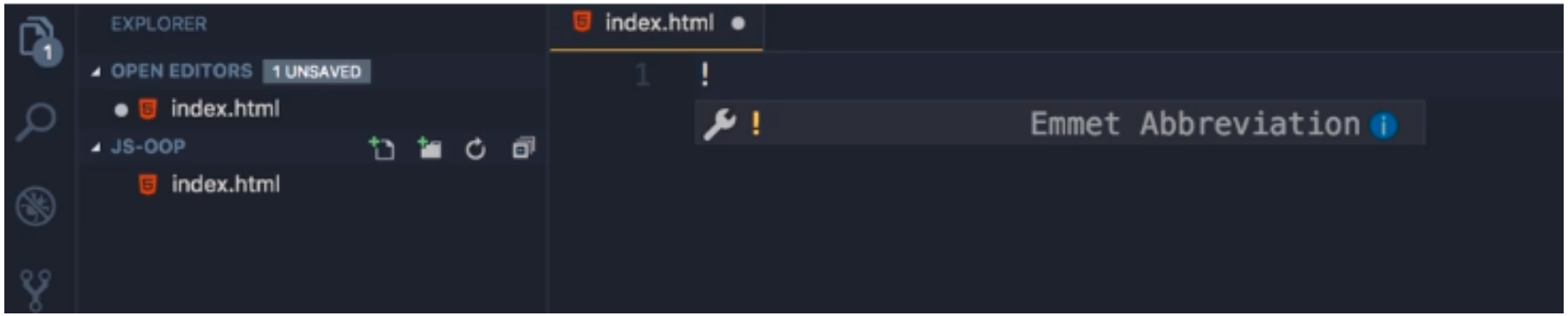
Header Describes the page

Body contains the content

Development Environment



Live Templates in CS Code (!) press tab



The <!DOCTYPE> Declaration

The **<!DOCTYPE>** declaration represents the document type, and helps browsers to display web pages correctly.

It must only appear once, at the top of the page (before any HTML tags).

The **<!DOCTYPE>** declaration is not case sensitive.

The **<!DOCTYPE>** declaration for HTML5 is:

<!DOCTYPE html>

HTML Headings

```
<h1>This is heading 1</h1>
<h2>This is heading 2</h2>
<h3>This is heading 3</h3>
```

This is heading 1

This is heading 2

This is heading 3

If H1 is too big What will you do

Convert h1 to h3?

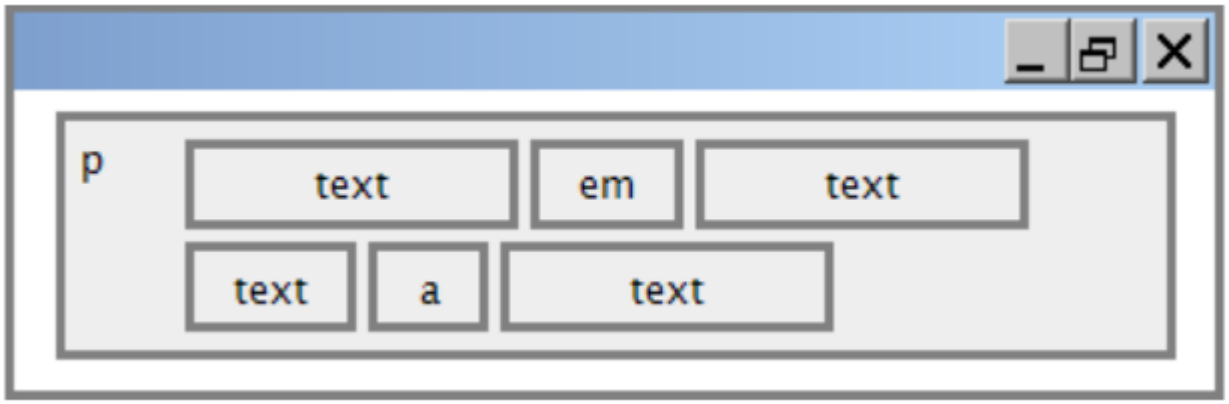
Semantic HTML

- use of HTML elements that carry meaning about the structure and content of a web page
- Use CSS for styling

HTML Comments

```
<p>This is a paragraph.</p>
<!--
<p>Look at this cool image:</p>
<img border="0" src="pic_trulli.jpg" alt="Trulli">
-->
<p>This is a paragraph too.</p>
```

Block And Inline Tags



Block

- Elements contains an entire large region (100% of available width) of content
- E.g. Paragraphs, lists, table etc

Inline

- Element effects a small amount of content
- E.g. bold, images,
- Browser allows many elements to appear on same line

HTML Attributes

All HTML elements can have attributes

Attributes provide additional information about elements

Attributes are always specified in the start tag

Attributes usually come in name/value pairs like: name="value"

The <a> tag defines a hyperlink. The href attribute specifies the URL of the page the link goes to:

```
<a href="https://www.w3schools.com">Visit W3Schools</a>
```

Links (external Vs internal)

```
<p>
  Search
  <a href="http://www.google.com/">Google</a> or
our
  <a href="lectures.html">Lecture Notes</a>
</p>
```

HTML

Link around anything

```
<a href="http://theonering.net/">
    <img src="images/gandalf.jpg" alt="Gandalf"
      title="You shall not pass!" />
</a>
```

HTML

output

The width and height Attributes

<img src="img_girl.jpg" width="200" height="200">

```
<img src="img_girl.jpg" alt="Girl with a jacket">
```

The required **alt** attribute for the **<img>** tag specifies an alternate text for an image, if the image for some reason cannot be displayed. This can be due to a slow connection, or an error in the src attribute, or if the user uses a screen reader.

The width and height attributes of the img tag, defines the width and height of the image:

The style Attribute

```
<p style="color:red;">This is a red paragraph.</p>
```

The **style** attribute is used to add styles to

<!DOCTYPE html>

<html>

<body **style="background-color:powderblue;">**

<h1>This is a heading</h1>

This is a paragraph.

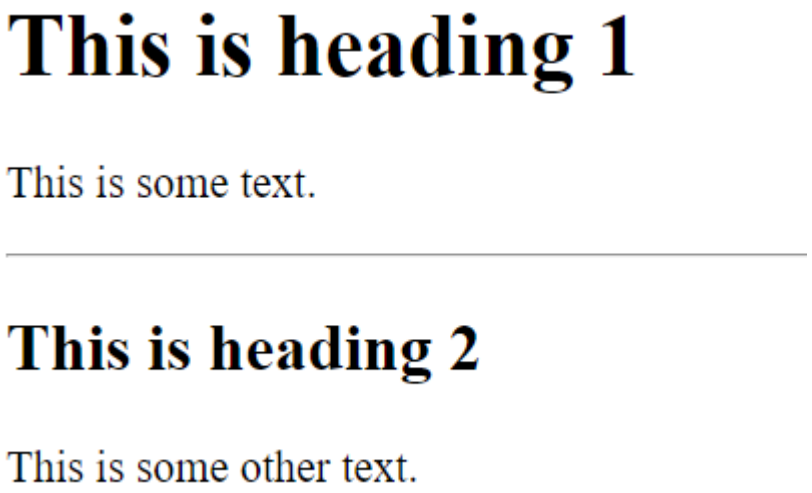
</body>

</html>

an element, such as color, font, size, and more.

Self Closing Tags

```
<!DOCTYPE html>
<html>
<body>
<h1>This is heading 1</h1>
<p>This is some text.</p>
<hr>
<h2>This is heading 2</h2>
<p>This is some other text.</p>
<hr>
</body>
</html>
```



HTML Text Formatting Elements

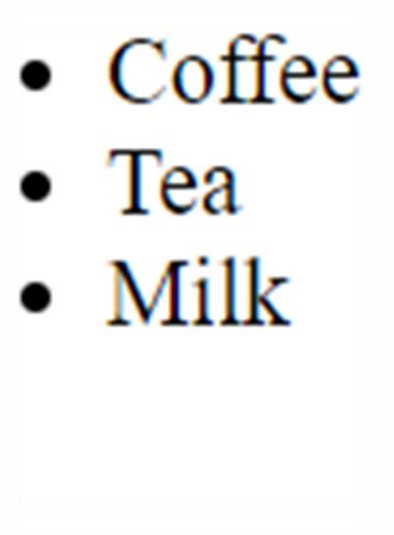
Tag	Description
<b>	Defines bold text
	Defines emphasized text
<i>	Defines a part of text in an alternate voice or mood
<small>	Defines smaller text
	Defines important text
<sub>	Defines subscripted text
<sup>	Defines superscripted text
<ins>	Defines inserted text
<del>	Defines deleted text
<mark>	Defines marked/highlighted text

Image Map

```
<img src="workplace.jpg" alt="Workplace" usemap="#workmap" width="400" height="379">
<map name="workmap">
  <area shape="rect" coords="34,44,270,350" alt="Computer" href="computer.htm">
  <area shape="rect" coords="290,172,333,250" alt="Phone" href="phone.htm">
  <area shape="circle" coords="337,300,44" alt="Cup of coffee" href="coffee.htm">
</map>
```

Unordered List

```
<ul>
  <li>Coffee</li>
  <li>Tea</li>
  <li>Milk</li>
</ul>
```



Nested Lists

```
<ul>
  <li>Simpsons:
    <ul>
      <li>Homer</li>
      <li>Marge</li>
    </ul>
  </li>
  <li>Family Guy:
    <ul>
      <li>Peter</li>
      <li>Lois</li>
    </ul>
  </li>
</ul>
```

HTML

- Simpsons:
  - Homer
  - Marge
- Family Guy:
  - Peter
  - Lois

output

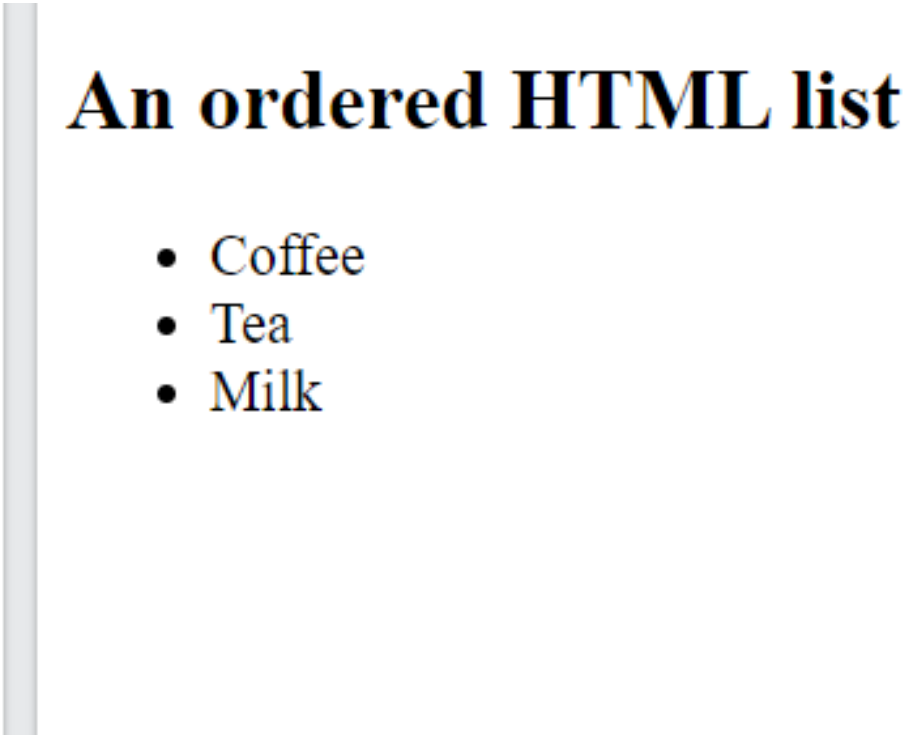
Ordered List

```
<ol>
  <li>Coffee</li>
  <li>Tea</li>
  <li>Milk</li>
</ol>
```

1. Coffee
2. Tea
3. Milk

Use CSS for looks and html for structure

```
<!DOCTYPE html>
<html>
<body>
<h2>An ordered HTML list</h2>
<ol style="list-style-type:disc">
  <li>Coffee</li>
  <li>Tea</li>
  <li>Milk</li>
</ol>
</body>
</html>
```



HTML CSS (Cascading Style Sheet)

```
body {background-color: powderblue;}
h1   {color: blue;}
p    {color: red;}
```

CSS is the language we use to style an HTML document.

CSS describes how HTML elements should be displayed.

CSS is used to define styles for your web pages, including the design, layout and variations in display for different devices and screen sizes.

Using CSS

Inline - by using the style attribute inside HTML elements

- `<h1 style="color:blue;">A Blue Heading</h1>`

Internal - by using a <style> element in the <head> section

```
<style>
body {background-color: powderblue;}
h1   {color: blue;}
p    {color: red;}
</style>
```

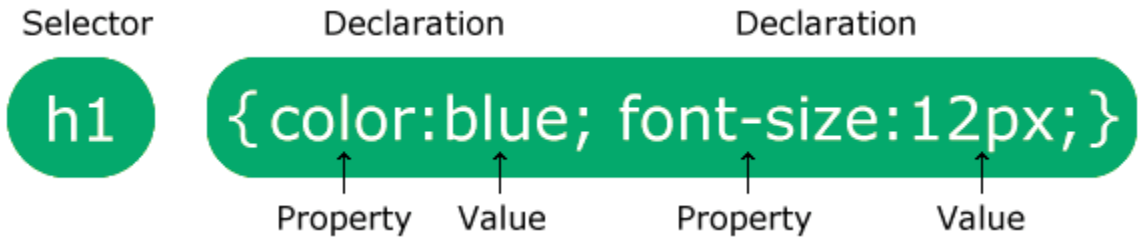
External - by using a <link> element to link to an external CSS file

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

"styles.css"

```
body {
  background-color: powderblue;
}
h1 {
  color: blue;
}
p {
  color: red;
}
```

CSS Syntax



The selector points to the HTML element you want to style.

The declaration block contains one or more declarations separated by semicolons.

Each declaration includes a CSS property **name and a value**, separated by a colon.

Multiple CSS declarations are separated with semicolons, and declaration blocks are surrounded by curly braces.

CSS Syntax

```
selector {
   property 1: value 1;
…
   property n: value n;
}
```

CSS

```
p {
   font-family: sans-serif;
   color:red;
}
```

CSS

CSS Colors

```css
p { color: red; }
h2 { color: rgb(128, 0, 196); }
h4 { color: #FF8800; }
```

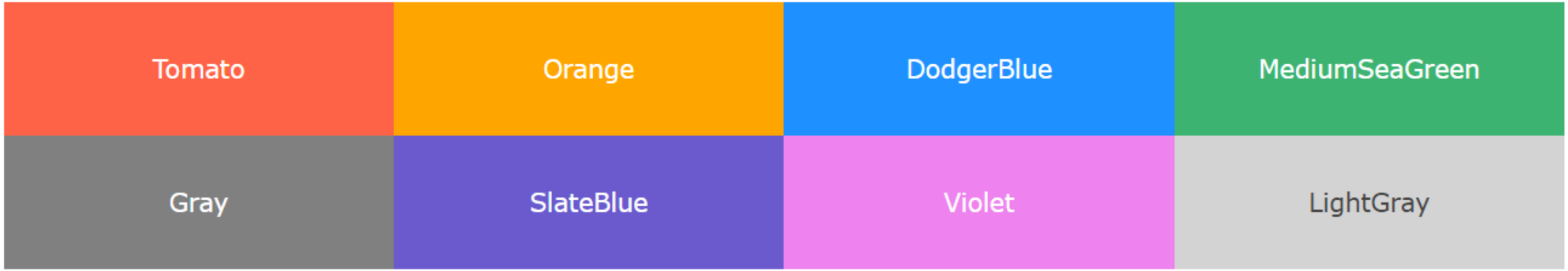
CSS

This paragraph uses the first style above.

This h2 uses the second style above.

This h4 uses the third style above.

output



font-family

```
p { font-family; Georgia; }
h2 { font-family: "Courier New"; }
```

CSS

This paragraph uses the Georgia font.

This h2 uses the Courier New font.

output

What are Web Safe Fonts?

Web safe fonts are fonts that are universally installed across all browsers and devices

- there are no 100% completely web safe fonts

```
p {
font-family: Tahoma, Verdana, sans-serif;
}
```

The second and third fonts are backups, in case the first one is not found.

font-size

```css
p {
  font-size: 20pt;
 }
```

CSS

This paragraph uses font size 20pt.

output

- **Units**: pixels (px) vs. point(pt) vs. m-size(em)
  - 16px, 16pt, 1.16em
- **Vague font size**: xx-small, x-small, small, medium, large, x-large, xx-large, smaller, larger
- **Percentage font sizes**, e.g.: 90%, 120%

Set Font Size With Em

To allow users to resize the text (in the browser menu)

1em is equal to the current font size.

The default text size in browsers is 16px.

So, the default size of 1em is 16px.

The size can be calculated from pixels to em using this formula: pixels/16=em

```
h1 {
  font-size: 2.5em; /* 40px/16=2.5em */
}

h2 {
  font-size: 1.875em; /* 30px/16=1.875em */
}

p {
  font-size: 0.875em; /* 14px/16=0.875em */
}
```

font-style

- normal - The text is shown normally
- italic - The text is shown in italics
- oblique - The text is "leaning" (oblique is very similar to italic, but less supported)