ATEC 3361 INTERNET STUDIO

CLASS 03
RIGHT-BRAIN WARM UP
CSS BASICS

Welcome back. We are on week 3: we'll continue our exploration of HTML and begin looking at CSS.

BRAIN WARM UPS

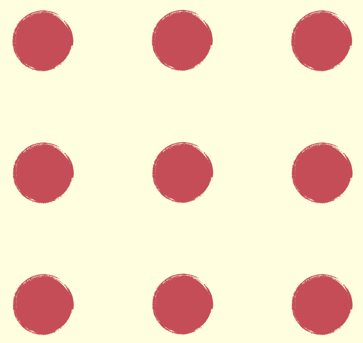
Let’s begin by warming up your brain.

BRAIN WARM UPS

Viewing a situation in conventional terms can create problems that don't actually exist.

This isn't a trick, but a demonstration of the human tendency to make assumptions.

Link all the dots in four straight lines without lifting your pen. Angles are permitted but not wobbles or bends.

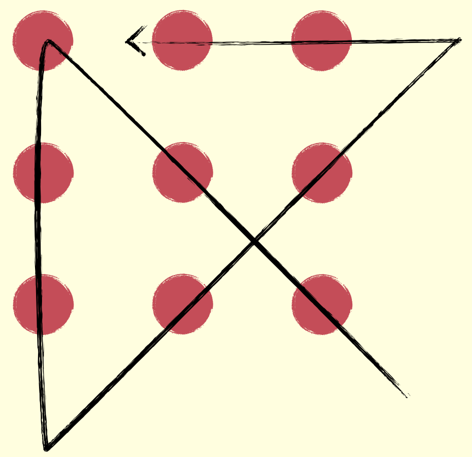


BRAIN WARM UPS

Viewing a situation in conventional terms can create problems that don't actually exist.

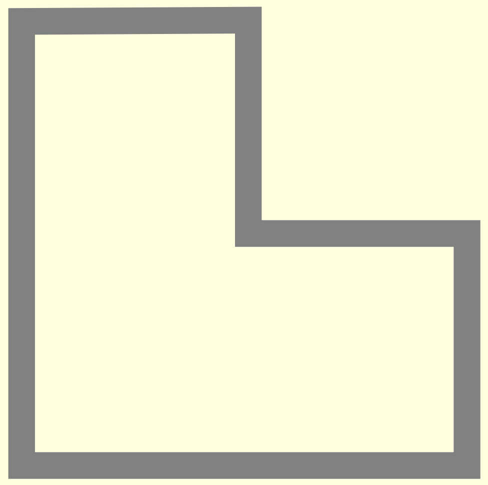
This isn't a trick, but a demonstration of the human tendency to make assumptions.

Link all the dots in four straight lines without lifting your pen. Angles are permitted but not wobbles or bends.



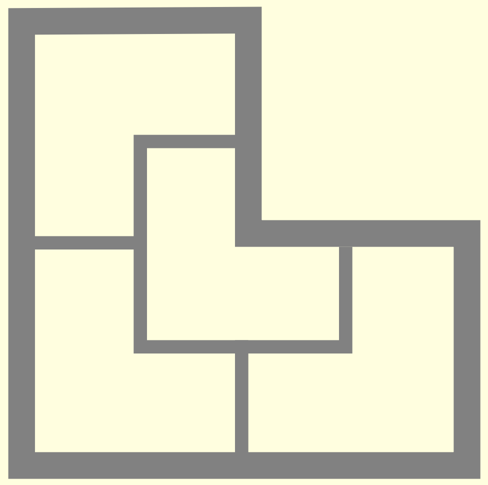
BRAIN WARM UPS

Slice this object into four (4) identical sections.



BRAIN WARM UPS

Slice this object into four (4) identical sections.

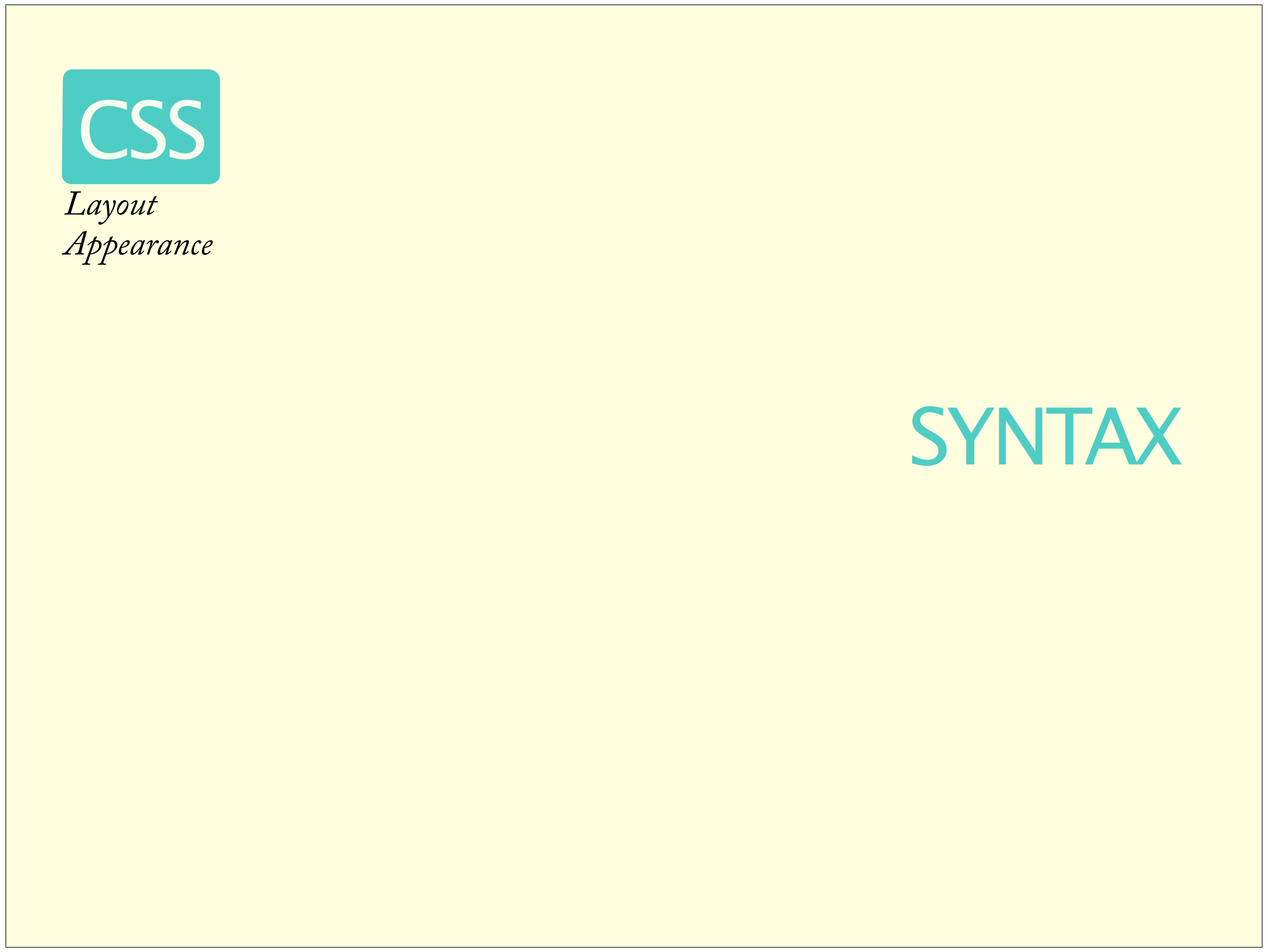


LESSON IN A TWEET

Finding answers means looking for what's not there. The answers are simple — but simple doesn't mean easy.

CSS

Let’s take a closer at CSS: cascading style sheets.

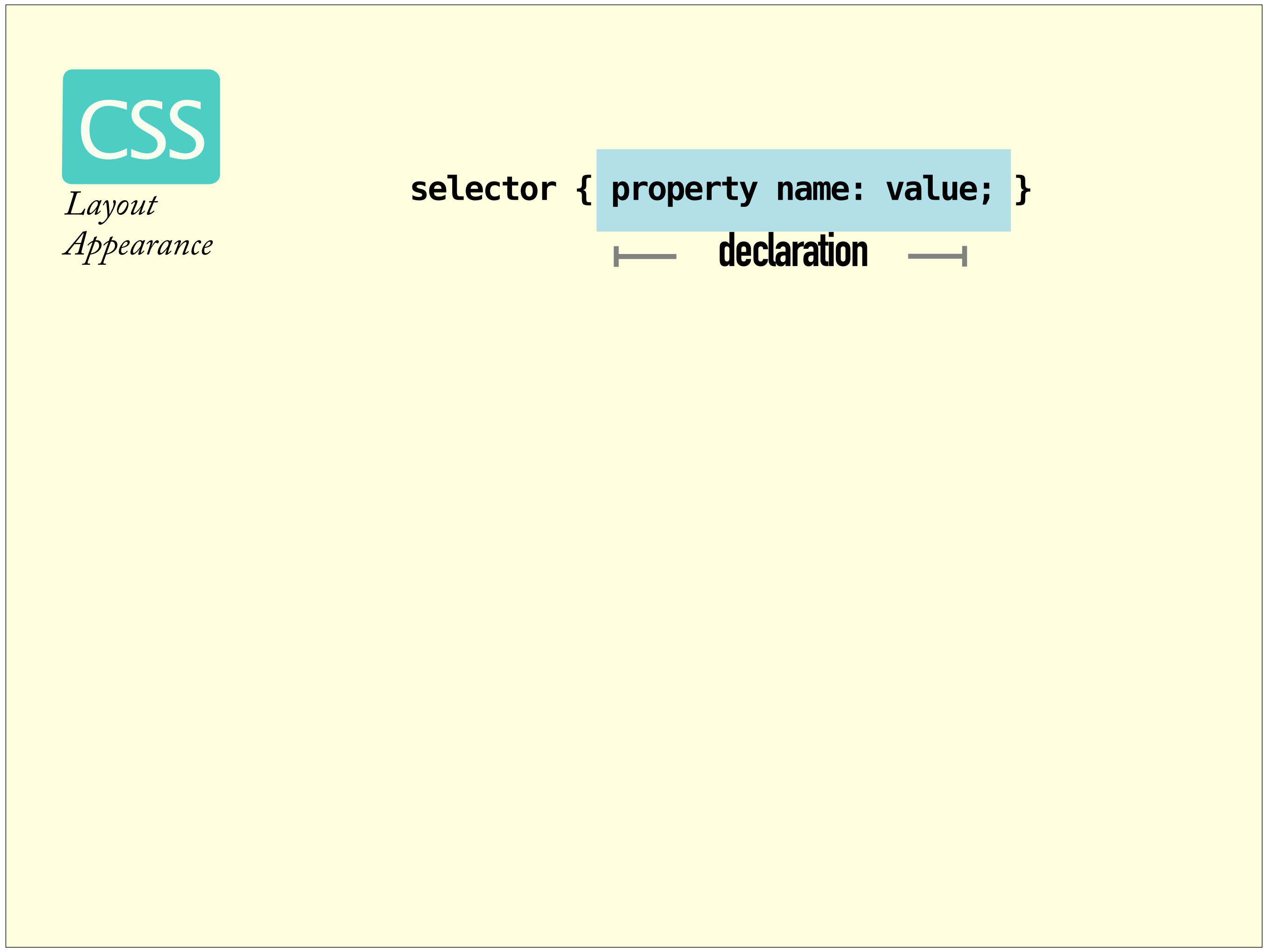


CSS

Layout
Appearance

SYNTAX

Remember that CSS handles layout and appearance. Let’s take a look at correct syntax.


CSS

Layout
Appearance

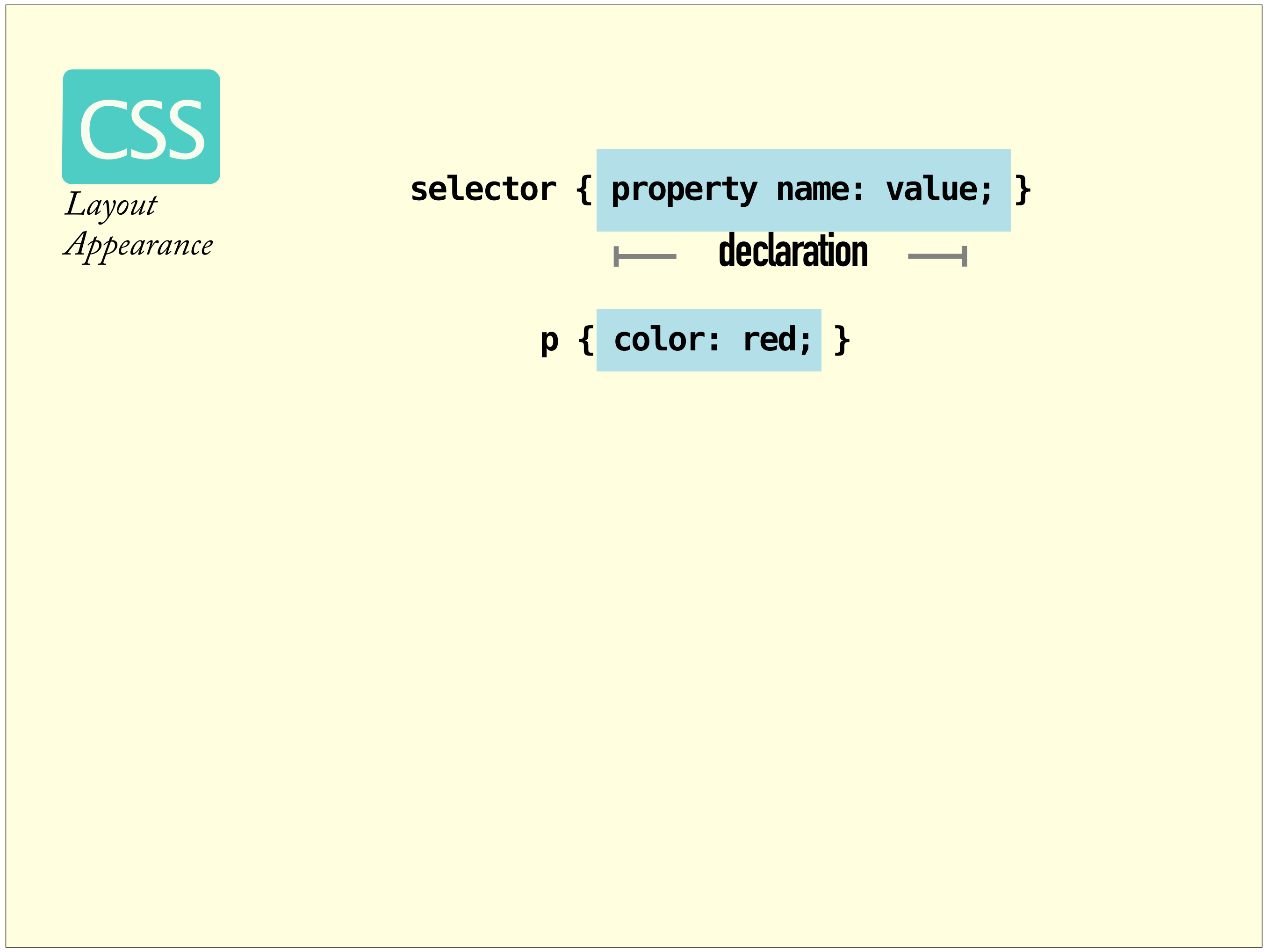
```
selector { property name: value; }
```

declaration

The selector specifies the HTML target that will be styled. It can be any HTML tag: <p>, <h1>, etc.

The declaration specifies the property and value, i.e. the styles that will be applied to the specified HTML element.

The declaration is contained between the opening and closing bracket. Declarations end with a semicolon.



CSS

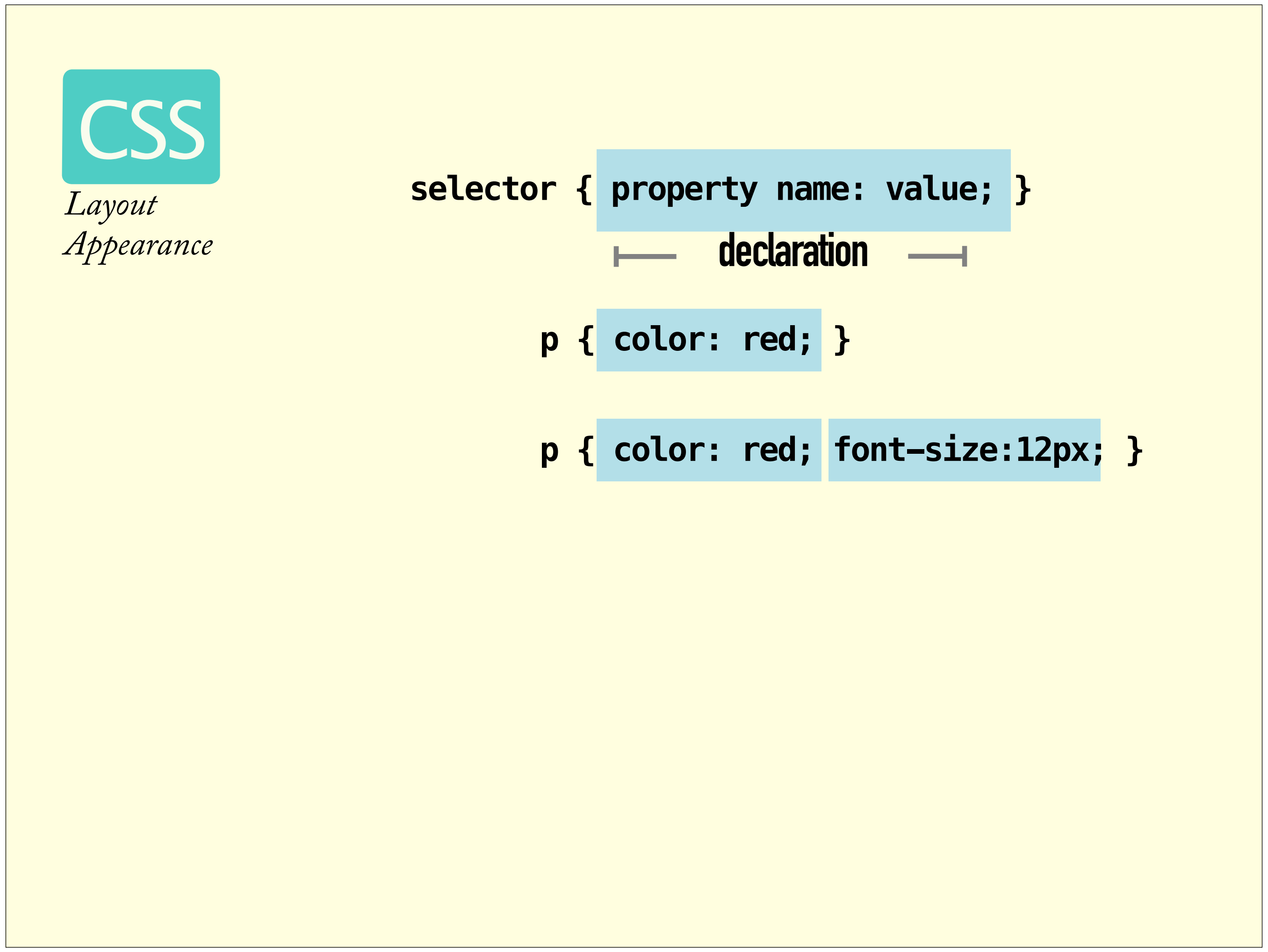
Layout
Appearance

```
selector { property name: value; }
```

$\vdash$ declaration $\dashv$

```
p { color: red; }
```

In this example, every <p> element on the page will be shaded red instead of the generic black text.



CSS

Layout
Appearance

```
selector { property name: value; }
           declaration

p { color: red; }

p { color: red; font-size:12px; }
```

Here, every <p> element on the page will be shaded red instead of the generic black text and will be sized at 12 pixels.

Notice also that there are two declarations (1) color: red, and (2) font-size:12px. You can have multiple declarations for each selection.

CSS

Layout
Appearance

```
selector { property name: value; }
```

declaration

```
p { color: red; }

p { color: red; font-size:12px; }

p { color: red;
    font-size: 32px;
}
```

CSS

Layout
Appearance

```
selector { property name: value; }
```

$\underbrace{\texttt{property name: value;}}_{\text{declaration}}$

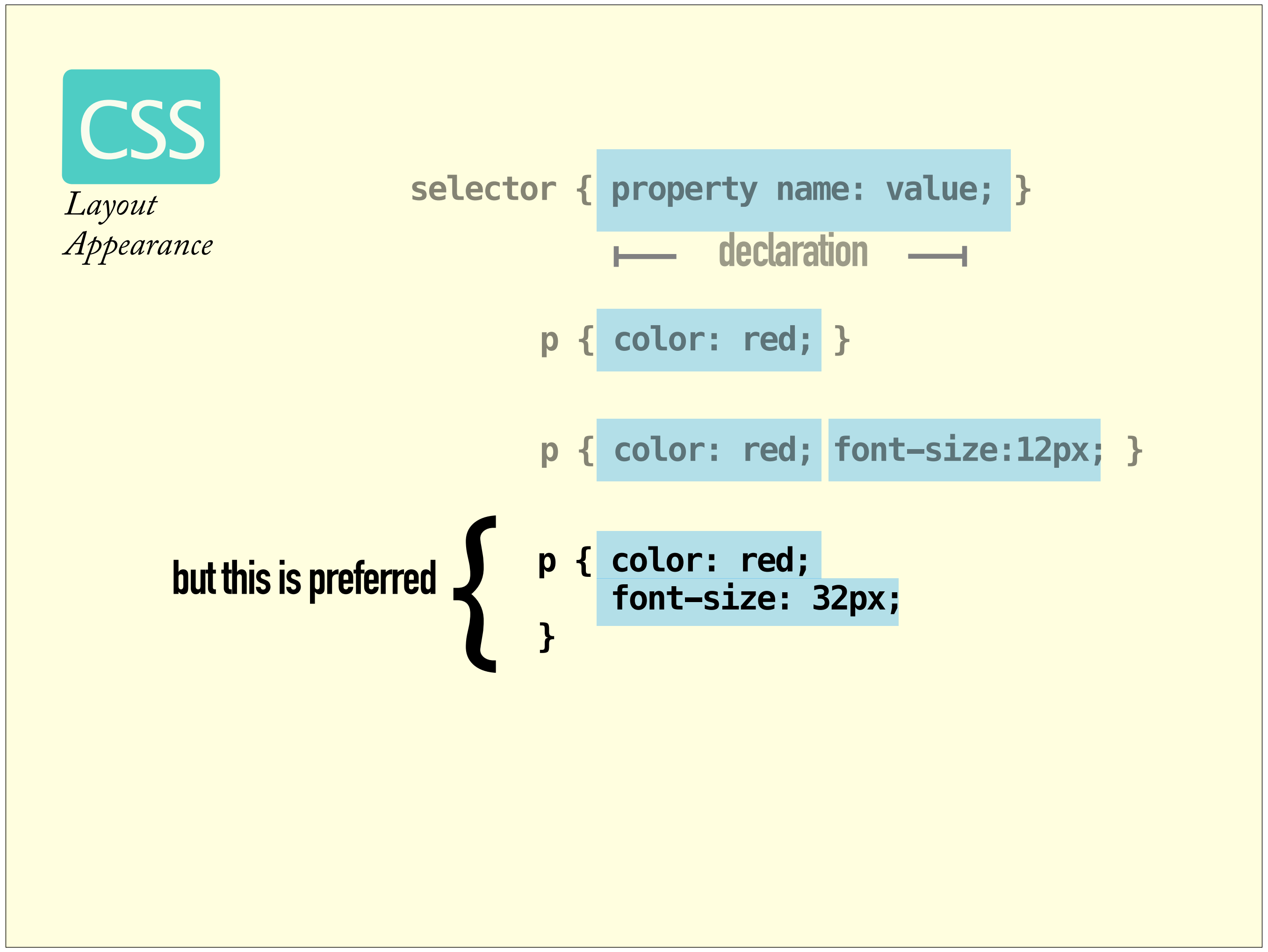
```
p { color: red; }
```

both are correct

```
p { color: red; font-size:12px; }

p { color: red;
    font-size: 32px;
}
```

You can write the styles on either one line as in the top example, or on multiple lines ...



CSS

Layout
Appearance

```
selector { property name: value; }
           ⊢——   declaration   ——⊣
```

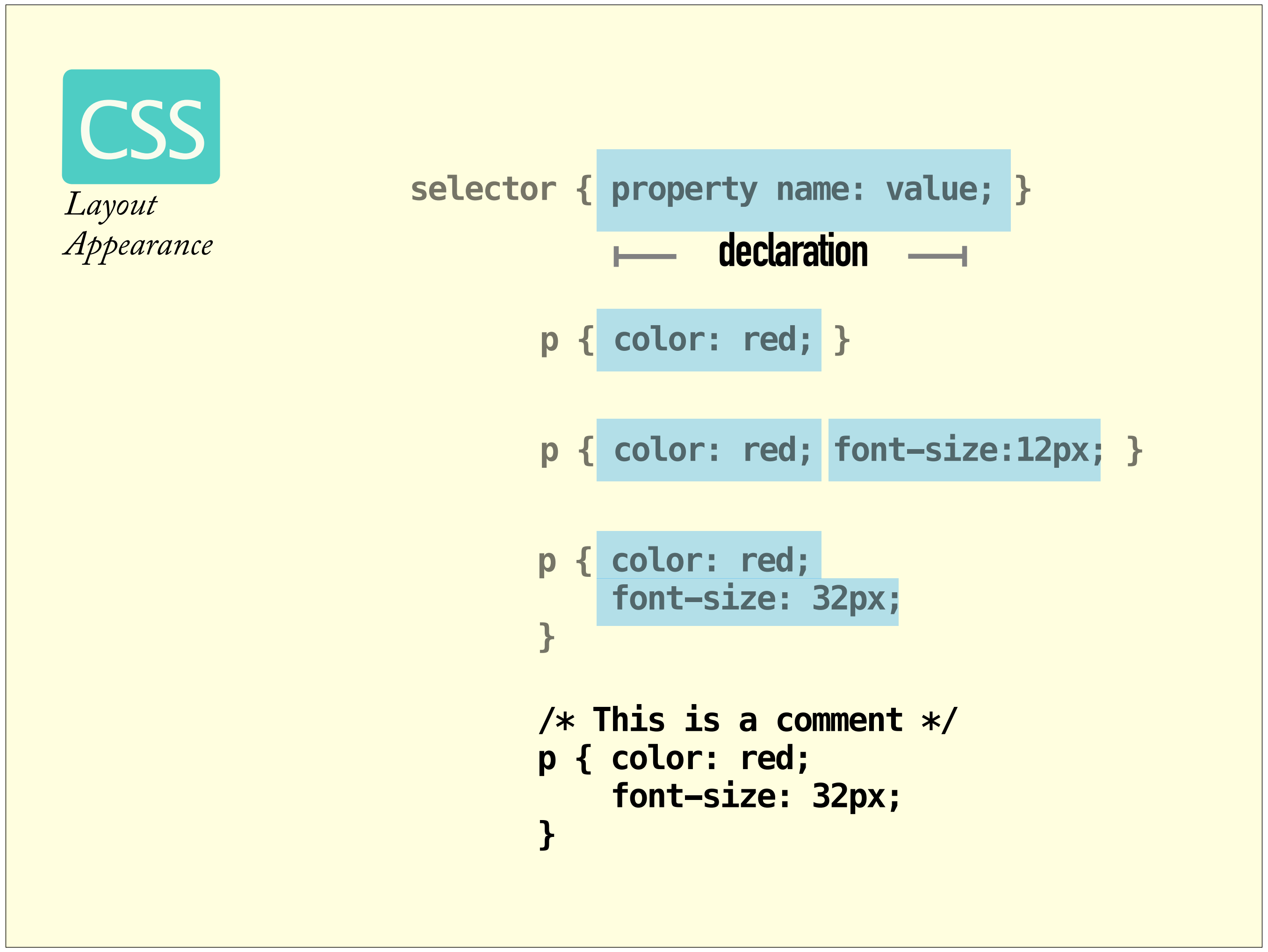
```
p { color: red; }
```

```
p { color: red; font-size:12px; }
```

but this is preferred

```
p { color: red;
    font-size: 32px;
}
```

... but writing each declaration on a separate line is preferred. When you have many declarations, it is far easier to scan and read.



CSS

Layout
Appearance

```
selector { property name: value; }
           declaration

p { color: red; }

p { color: red; font-size:12px; }

p { color: red;
    font-size: 32px;
}

/* This is a comment */
p { color: red;
    font-size: 32px;
}
```

And you can write comments in CSS by putting /* ... */ around the text you want commented. Make sure you open it with a /* and ending it with a */

CSS

Layout
Appearance

SELECTORS

Let’s talk about selectors.

CSS

Layout
Appearance

```
selector { property name: value; }
```

EXAMPLES

```
p { color: red; }
h1 { font-size: 12px; }
```

Remember that selectors specify the HTML target that will be styled.

It can be any HTML tag: <p>, <h1>, etc.

CSS

Layout
Appearance

HTML CODE

```
<p>This is a paragraph.</p>
<p>This is a paragraph.</p>
<p>This is a paragraph.</p>
<p>This is a paragraph.</p>
```

CSS

```
p { color: red;
    font-size: 32px;
}
```

This CSS code will make all four paragraphs red and sized at 32 pixels ...

CSS

Layout
Appearance

HTML CODE

```
<p class="blue">This is a paragraph.</p>
<p>This is a paragraph.</p>
<p>This is a paragraph.</p>
<p>This is a paragraph.</p>
```

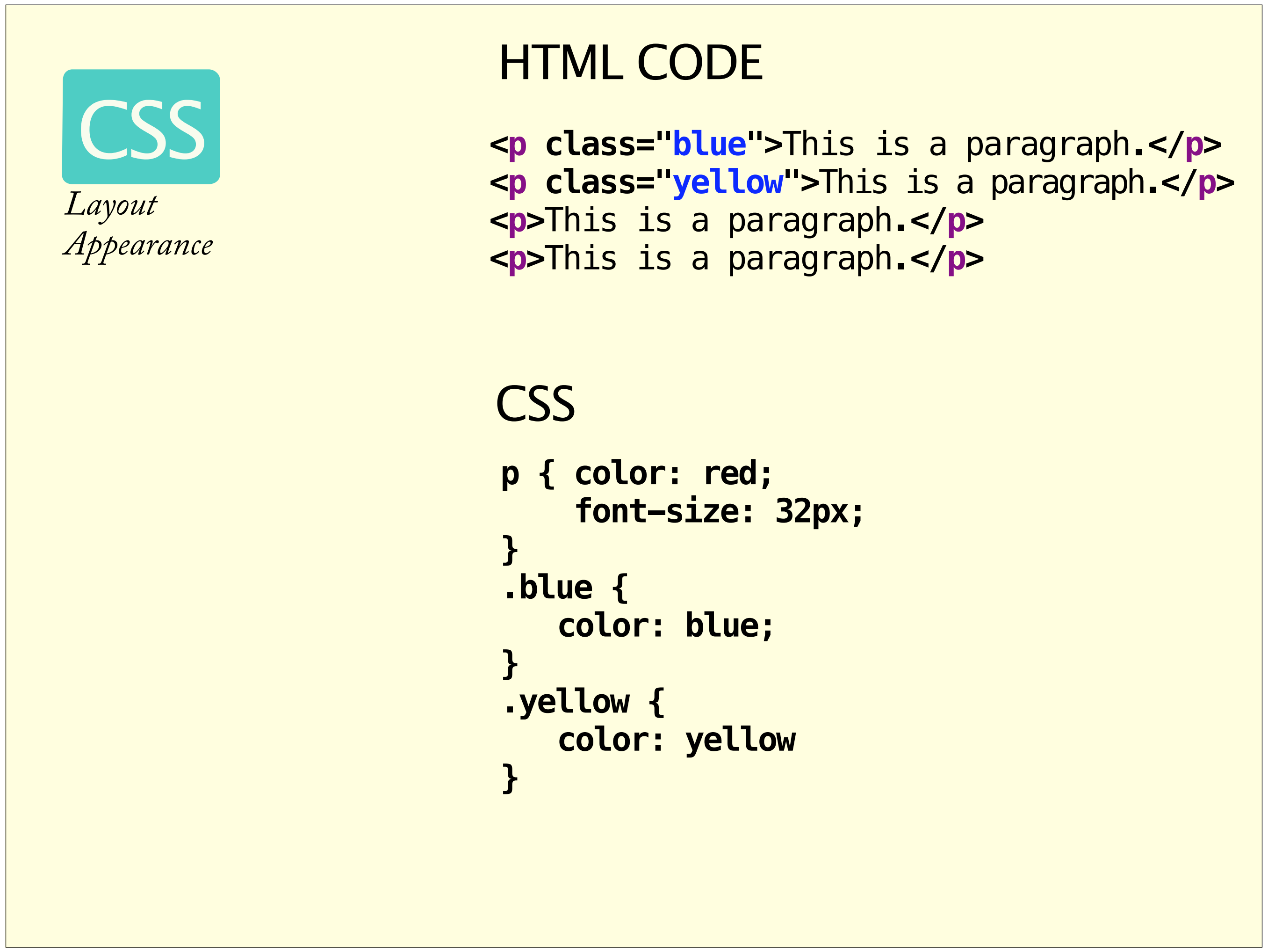
CSS

```
p { color: red;
    font-size: 32px;
}
.blue { color: blue; }
```

... and adding a class to the first <p> tag will change the color of that paragraph to blue.

Note that the HTML name of the class is provided by you, the coder. A class name can be anything.

In CSS, you denote a class with a period before the name of the class (here, .blue).



CSS

Layout
Appearance

HTML CODE

```
<p class="blue">This is a paragraph.</p>
<p class="yellow">This is a paragraph.</p>
<p>This is a paragraph.</p>
<p>This is a paragraph.</p>
```

CSS

```
p { color: red;
    font-size: 32px;
}
.blue {
   color: blue;
}
.yellow {
   color: yellow
}
```

And you can add many classes. Here we’ve add a yellow class that changes the text color of one paragraph to yellow.

CSS

Layout
Appearance

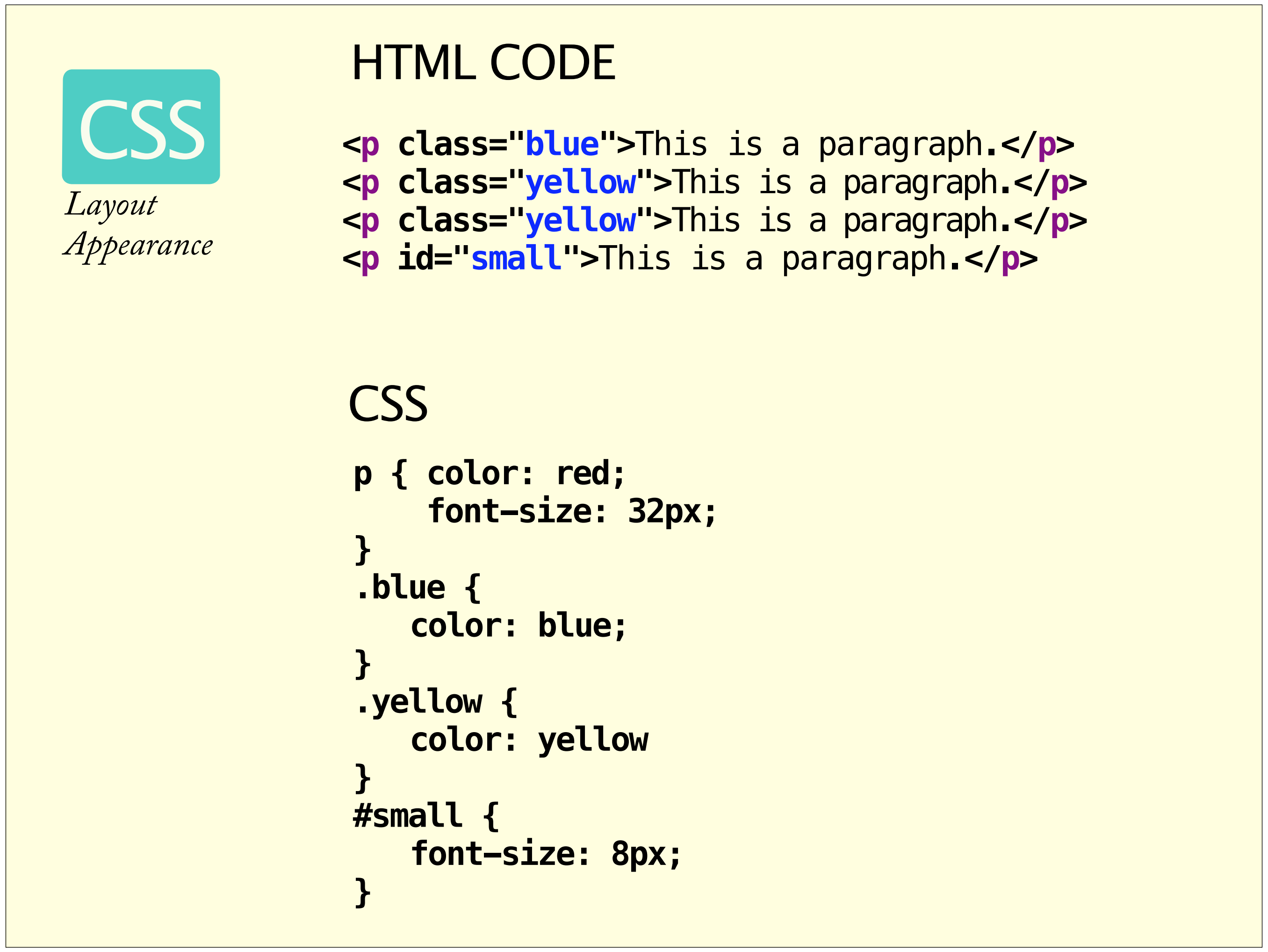
HTML CODE

```
<p class="blue">This is a paragraph.</p>
<p class="yellow">This is a paragraph.</p>
<p class="yellow">This is a paragraph.</p>
<p>This is a paragraph.</p>
```

CSS

```
p { color: red;
    font-size: 32px;
}
.blue {
   color: blue;
}
.yellow {
   color: yellow
}
```

Notice, you can use classes more than once in your HTML page. Here we have two paragraphs using the yellow class.



HTML CODE

```
<p class="blue">This is a paragraph.</p>
<p class="yellow">This is a paragraph.</p>
<p class="yellow">This is a paragraph.</p>
<p id="small">This is a paragraph.</p>
```

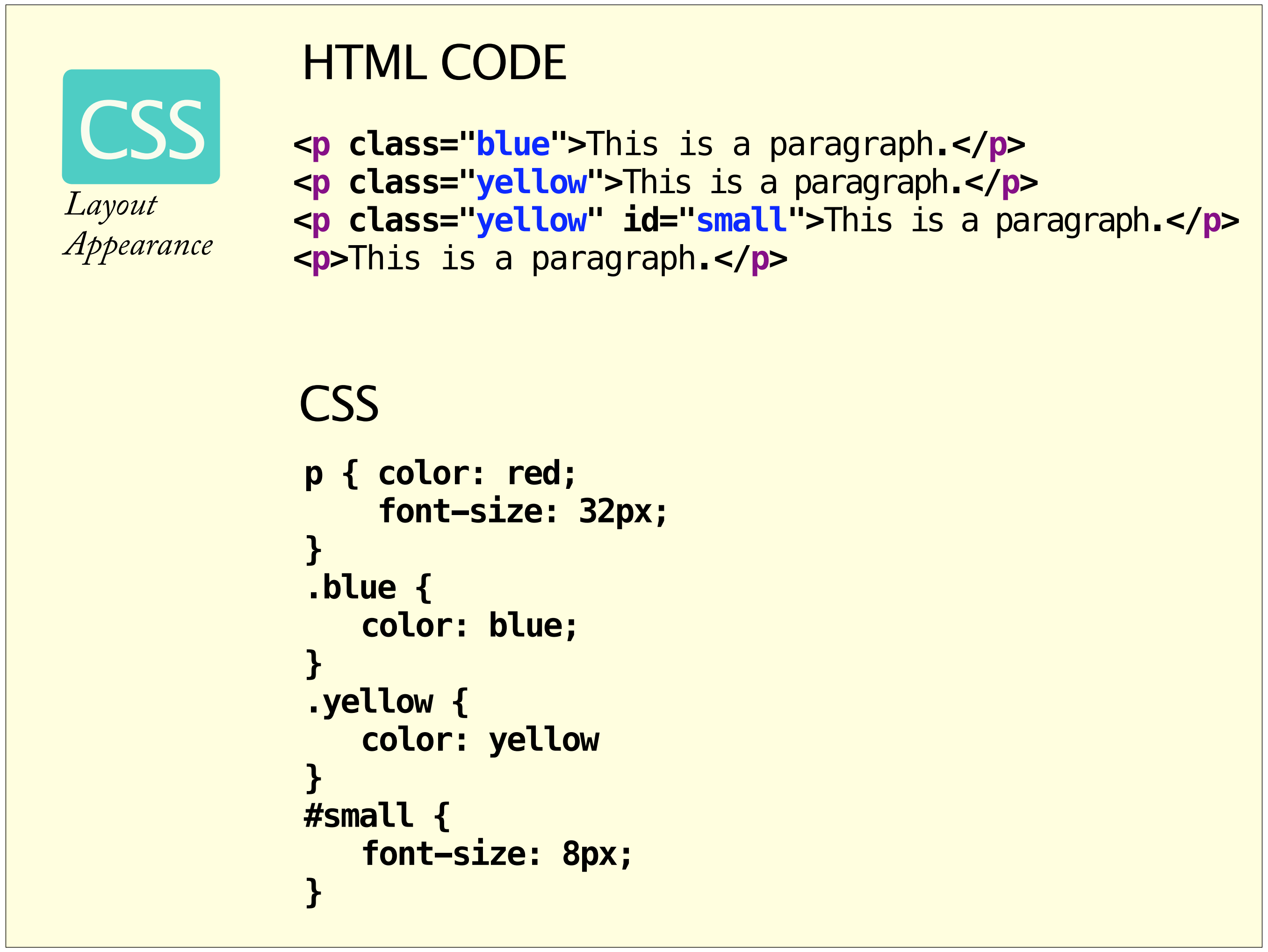
CSS

```
p { color: red;
    font-size: 32px;
}
.blue {
   color: blue;
}
.yellow {
   color: yellow
}
#small {
   font-size: 8px;
}
```

In this example, we've added an ID that makes the font size smaller. IDs are different from classes in that they can be used only once on a web page.

To help you differentiate between class and ID, think about your graduating class ... maybe you're part of the class of 2012. There are many people in the class 2012. Similarly, a class on a webpage can be used multiple times.

When you think of IDs, think about your Social Security Identification number (SSN). Every person has only one SSN -- similarly an ID can be used only once on a HTML page.



Layout Appearance

HTML CODE

```html
<p class="blue">This is a paragraph.</p>
<p class="yellow">This is a paragraph.</p>
<p class="yellow" id="small">This is a paragraph.</p>
<p>This is a paragraph.</p>
```

CSS

```css
p { color: red;
    font-size: 32px;
}
.blue {
   color: blue;
}
.yellow {
   color: yellow
}
#small {
   font-size: 8px;
}
```

On line 3 we've combined the "yellow" class with the "small" ID. This will apply both styles to paragraph 3. It's possible to combine class and IDs this way.

CSS

Layout
Appearance

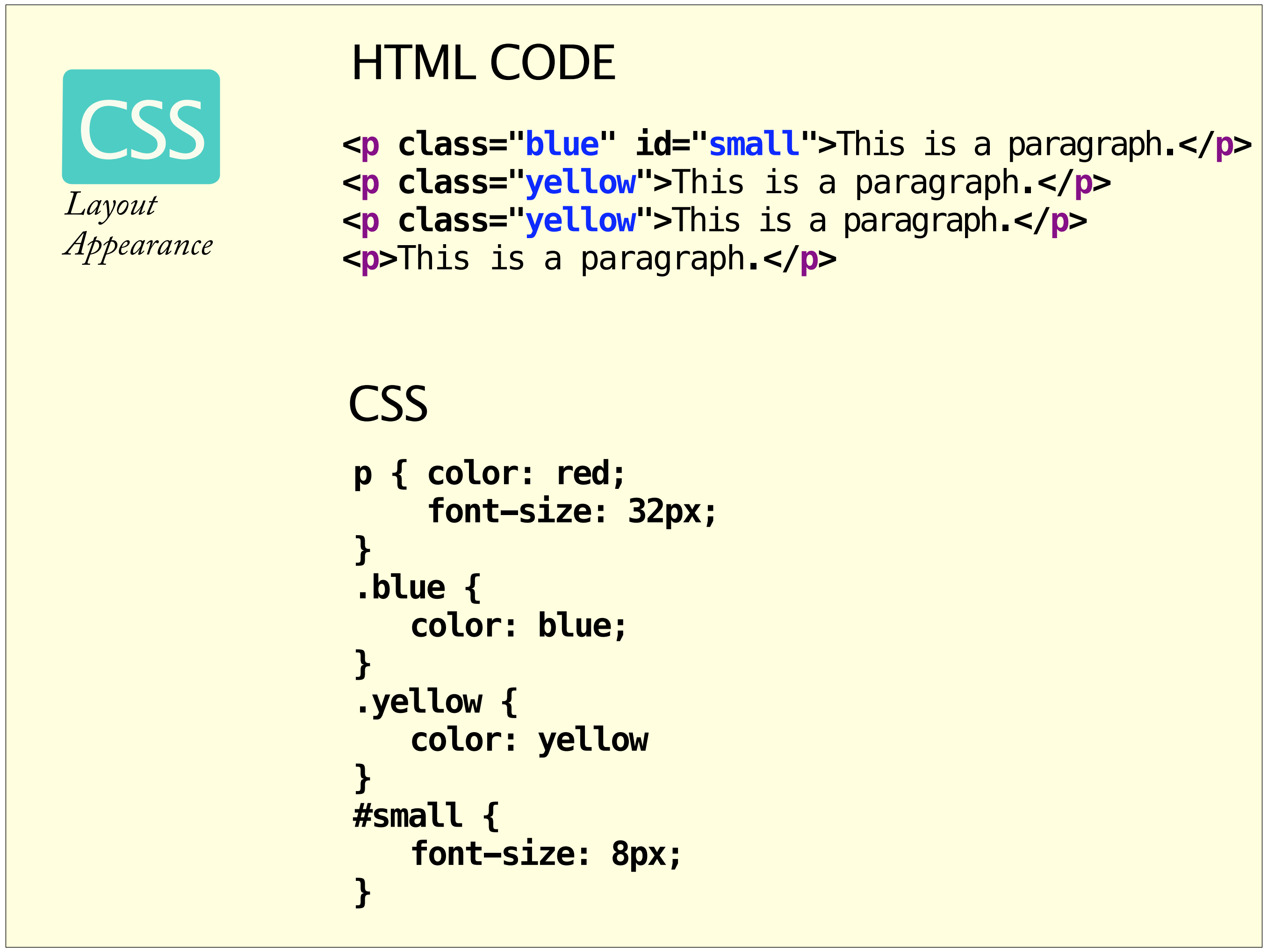
HTML CODE

```
<p class="blue">This is a paragraph.</p>
<p class="yellow" id="small">This is a paragraph.</p>
<p class="yellow">This is a paragraph.</p>
<p>This is a paragraph.</p>
```

CSS

```
p { color: red;
    font-size: 32px;
}
.blue {
   color: blue;
}
.yellow {
   color: yellow
}
#small {
   font-size: 8px;
}
```

Congratulations, you’re now using the colors on the flag of Romania.



CSS

Layout
Appearance

HTML CODE

```
<p class="blue" id="small">This is a paragraph.</p>
<p class="yellow">This is a paragraph.</p>
<p class="yellow">This is a paragraph.</p>
<p>This is a paragraph.</p>
```

CSS

```
p { color: red;
    font-size: 32px;
}
.blue {
   color: blue;
}
.yellow {
   color: yellow
}
#small {
   font-size: 8px;
}
```

And in this example, the “small” properties would be applied to line 1, with the “blue” class.

CSS

Layout
Appearance

HTML CODE

```
<p class="blue" id="small">This is a paragraph.</p>
<p class="yellow">This is a paragraph.</p>
<p class="yellow" id="small">This is a paragraph.</p>
<p>This is a paragraph.</p>
```

What's wrong with this code?

CSS
Layout
Appearance

HTML CODE

```
<p class="blue" id="small">This is a paragraph.</p>
<p class="yellow">This is a paragraph.</p>
<p class="yellow" ~~id="small"~~>This is a paragraph.</p>
<p>This is a paragraph.</p>
```

That's right. You can have the ID only once on the page.

CSS

Layout
Appearance

DESCENDANT COMBINATIONS

Let’s talk about descendant combinations.

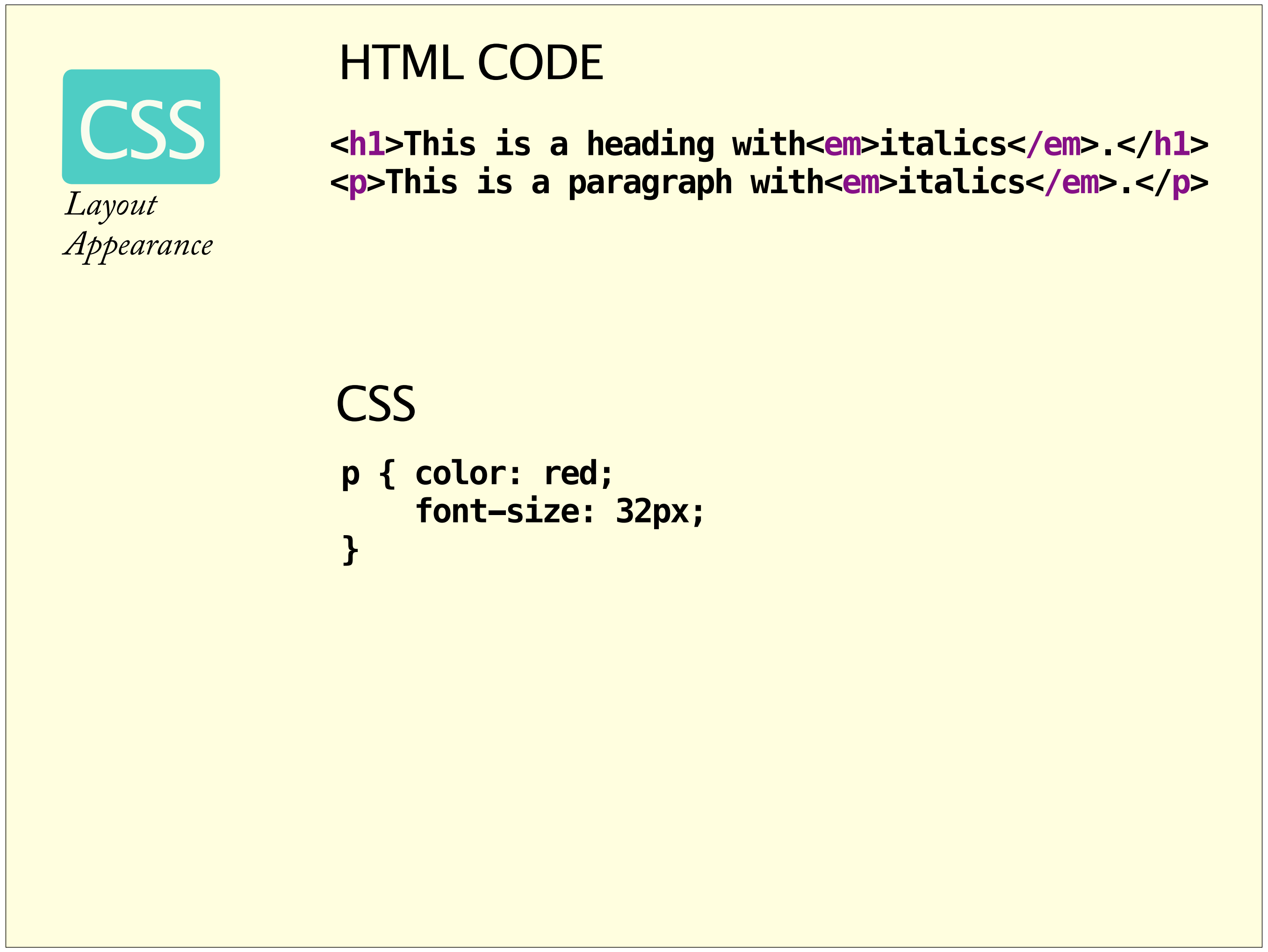
CSS
Layout
Appearance

HTML CODE

```
<h1>This is a heading with<em>italics</em>.</h1>
<p>This is a paragraph with<em>italics</em>.</p>
```

By descendant combinations, I'm referring to the <em> element inside the <h1> and <p> elements.

In line 1, the <em> is a descendant of the <h1> element.
In line 2, the <em> is a descendant of the <p> element.



CSS

Layout
Appearance

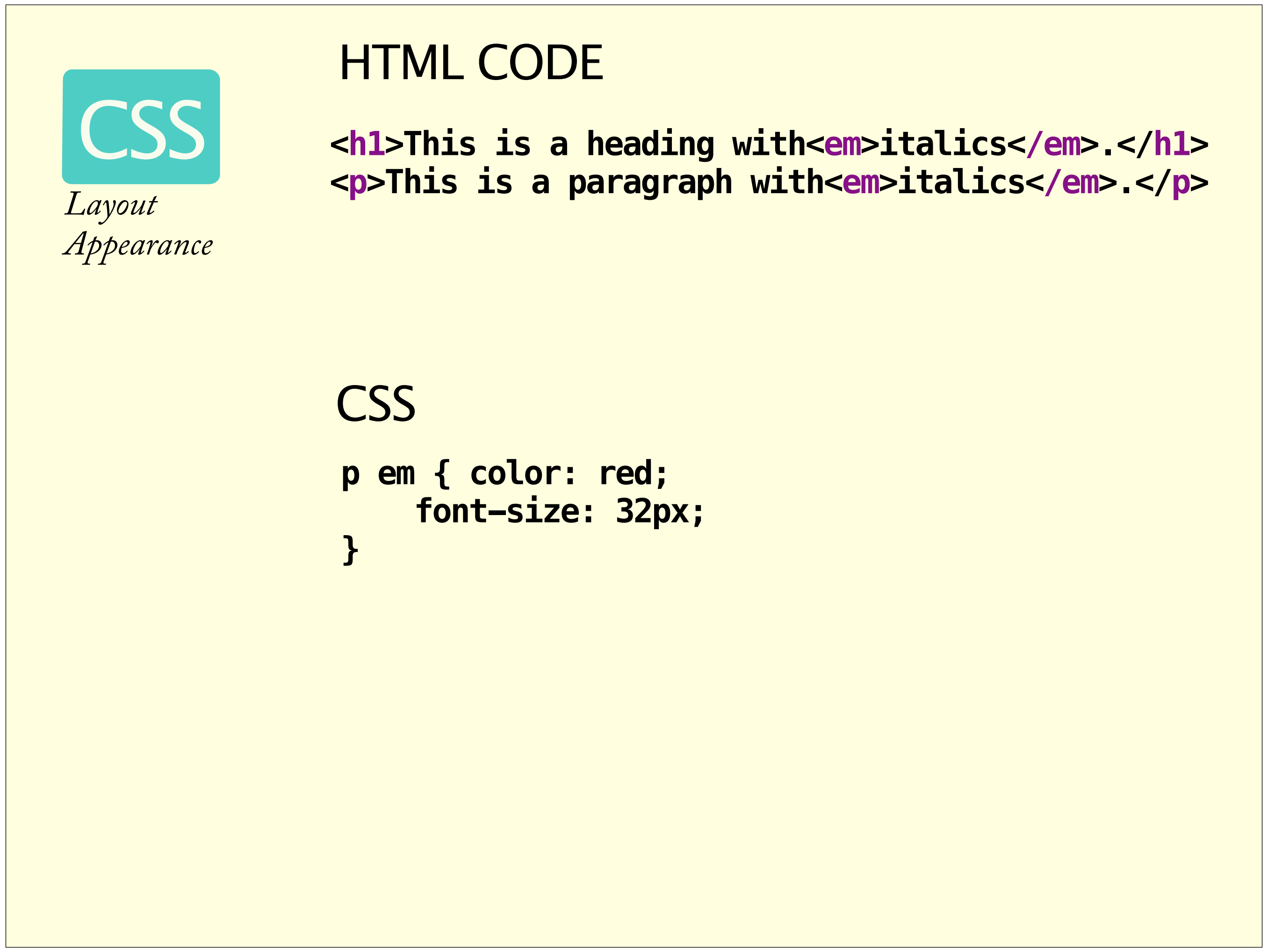
HTML CODE

```
<h1>This is a heading with<em>italics</em>.</h1>
<p>This is a paragraph with<em>italics</em>.</p>
```

CSS

```
p { color: red;
    font-size: 32px;
}
```

Conversely, this line of CSS code will make only the content in the <em> inside the <p> element red and sized at 32 pixels. In other words, only the word “paragraph” in line 2 will be red and sized at 32 pixels.


CSS

Layout
Appearance

HTML CODE

```
<h1>This is a heading with<em>italics</em>.</h1>
<p>This is a paragraph with<em>italics</em>.</p>
```

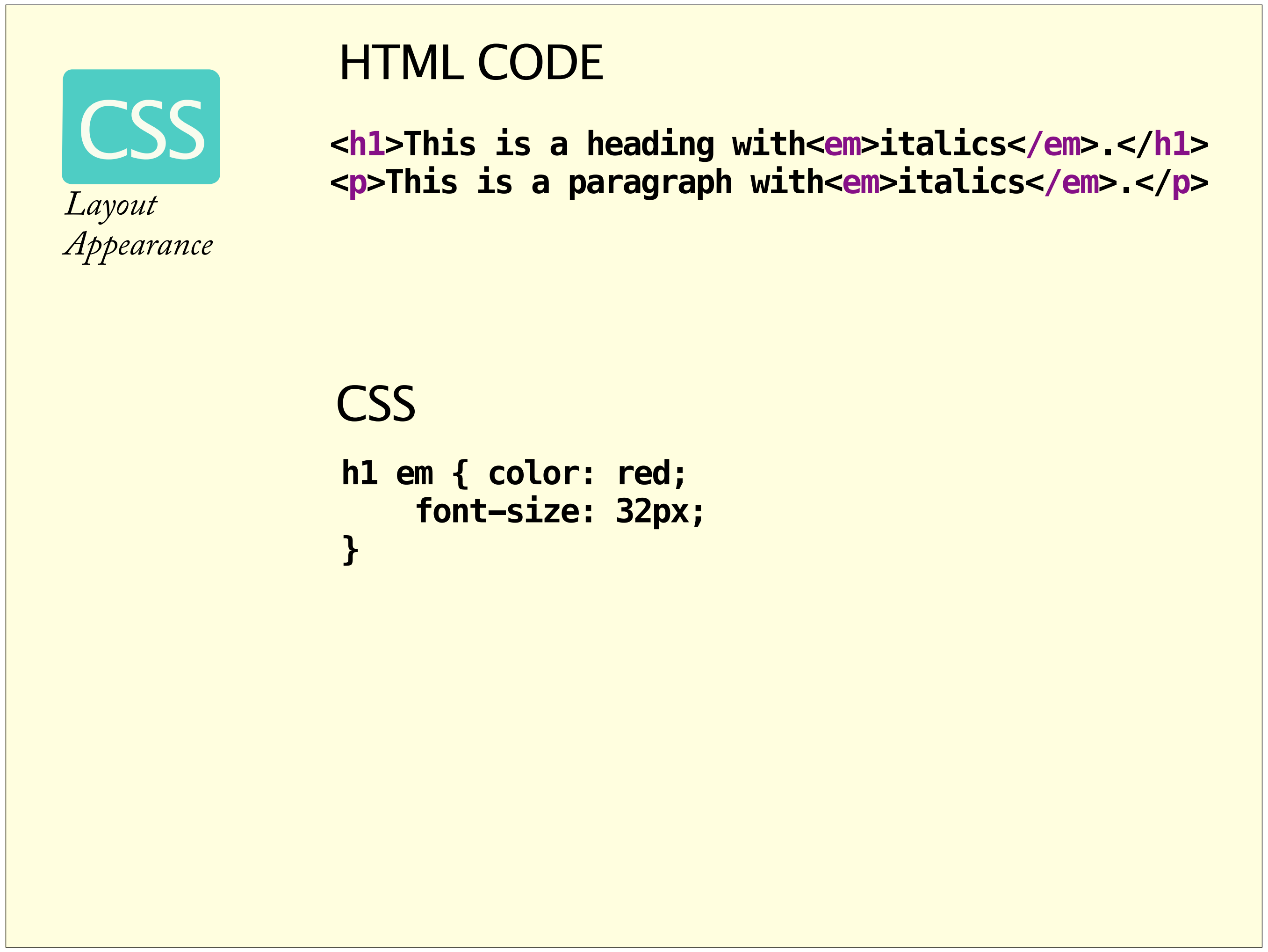
CSS

```
p em { color: red;
    font-size: 32px;
}
```

This line of CSS code will make the content in the <h1> element red and sized at 32 pixels. In other words

This is a heading

will be red and sized at 32 pixels.



CSS

Layout
Appearance

HTML CODE

```
<h1>This is a heading with<em>italics</em>.</h1>
<p>This is a paragraph with<em>italics</em>.</p>
```

CSS

```
h1 em { color: red;
    font-size: 32px;
}
```

While this line of CSS code will make only the content in the <em> inside the <h1> element red and sized at 32 pixels. In other words, only the word “heading” in line 1 will be red and sized at 32 pixels.

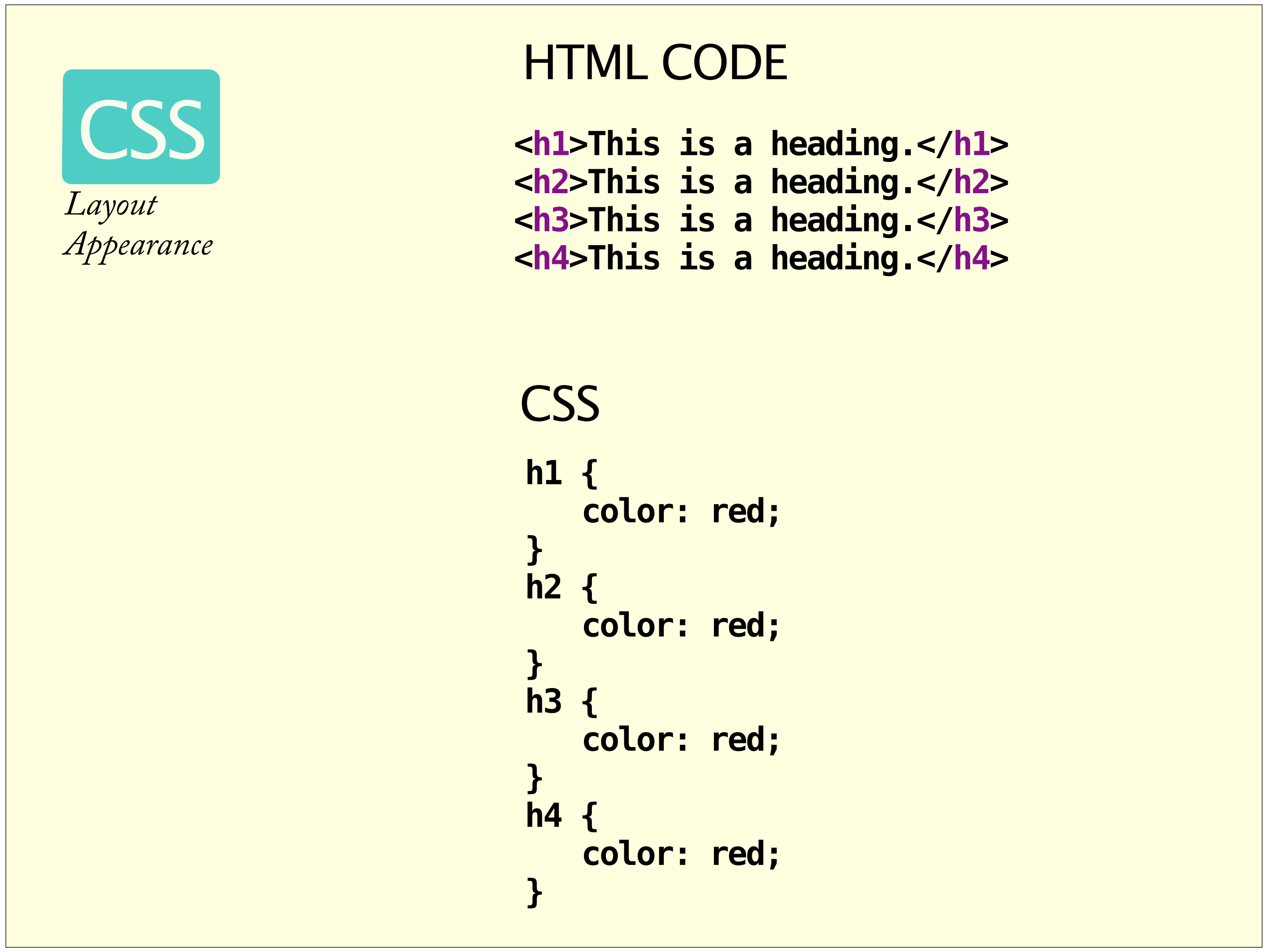
CSS

Layout
Appearance

HTML CODE

```
<h1>This is a heading.</h1>
<h2>This is a heading.</h2>
<h3>This is a heading.</h3>
<h4>This is a heading.</h4>
```

Here are four heading tags, <h1> through <h4> respectively.



CSS

Layout
Appearance

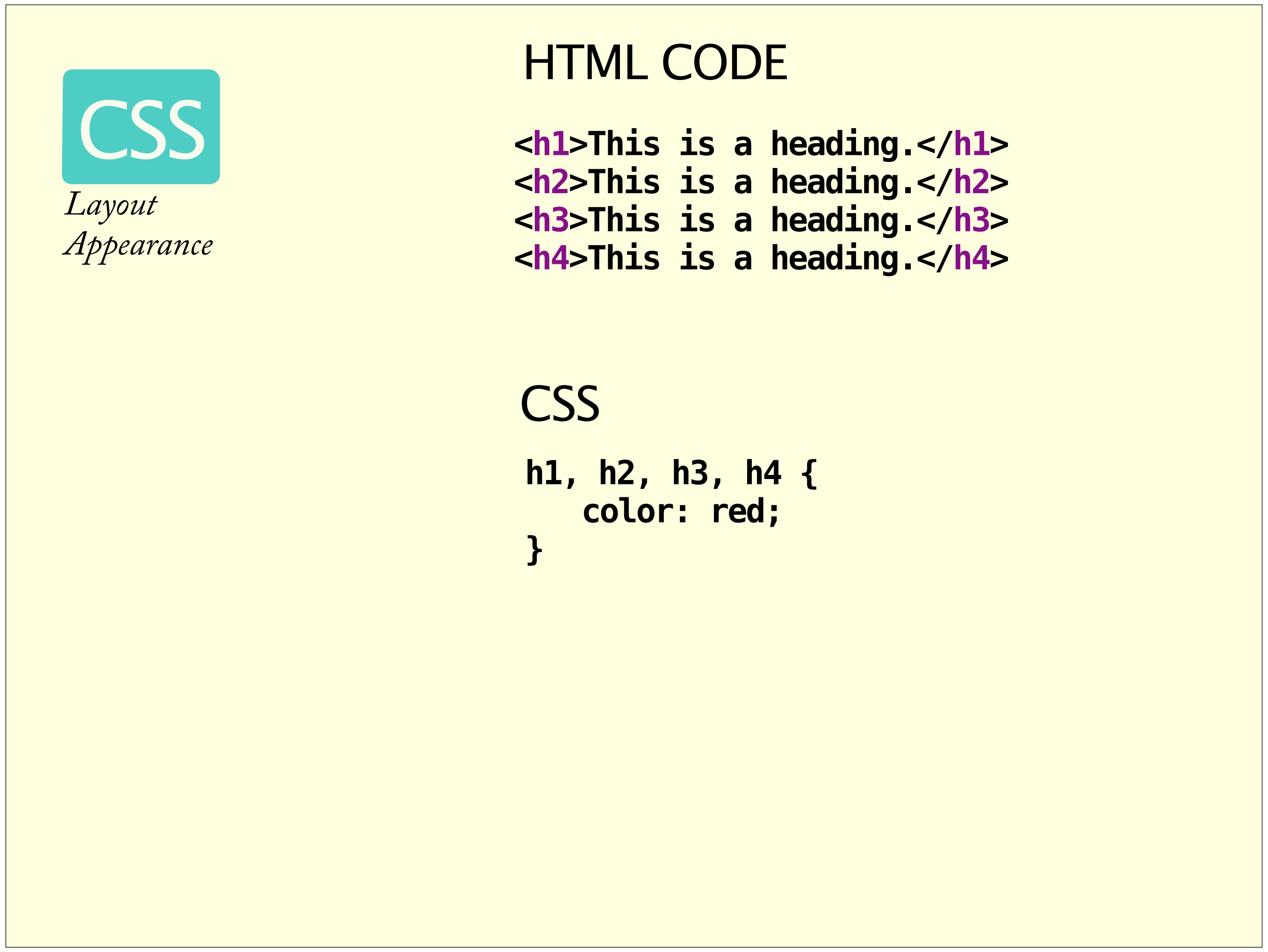
HTML CODE

```html
<h1>This is a heading.</h1>
<h2>This is a heading.</h2>
<h3>This is a heading.</h3>
<h4>This is a heading.</h4>
```

CSS

```css
h1 {
   color: red;
}
h2 {
   color: red;
}
h3 {
   color: red;
}
h4 {
   color: red;
}
```

Let’s say we want all four of these headings to be in the color red. Here’s one way we can write this ...



CSS

Layout
Appearance

HTML CODE

```
<h1>This is a heading.</h1>
<h2>This is a heading.</h2>
<h3>This is a heading.</h3>
<h4>This is a heading.</h4>
```

CSS

```
h1, h2, h3, h4 {
   color: red;
}
```

... or we can write more efficient code this way for these four elements.

CSS

Layout
Appearance

HTML CODE

STYLE SHEET

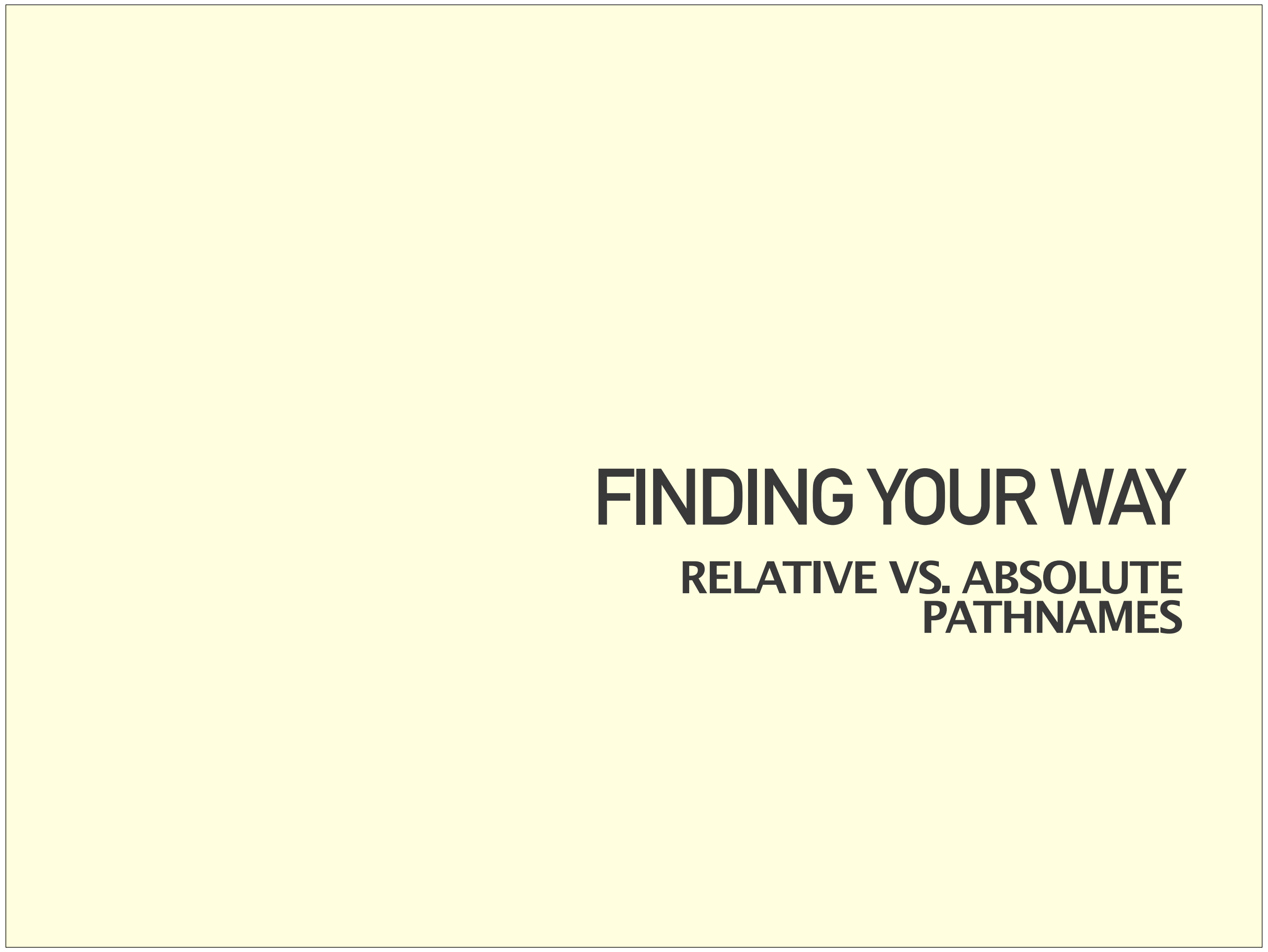
```
<!doctype html>
<html
<head

  <

  <style>
    /* add your CSS here */
    p {
   font-size:12px;
    }
  </style>
</
<body

</body
</html
```

And finally, here’s the code you need to add styles to your web page. Inside the <head> element, simply add a <style> element. Within the <style> element, you can add in your CSS styles.



FINDING YOUR WAY

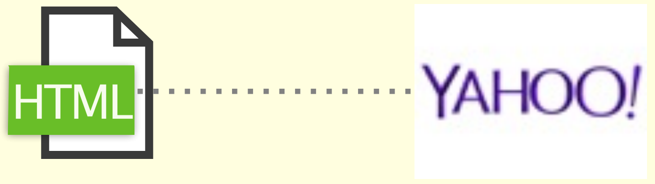
RELATIVE VS. ABSOLUTE PATHNAMES

One of the most common things newcomers to HTML get confused about is linking to other pages and sites, especially when absolute and relative paths come into play. But worry not! Creating links — relative and absolute alike — is actually fairly easy.

RELATIVE VS. ABSOLUTE PATHNAMES

Absolute pathname includes the full address (http://...) to the site.

Used to link to an external site.



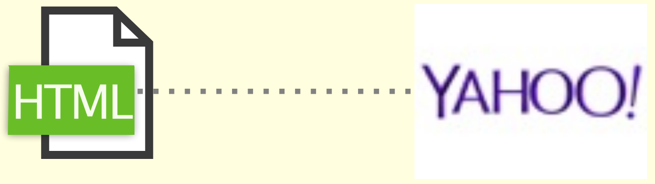
```
<a href=“http:www.yahoo.com”>yahoo.com</a>
```

As you may have guessed, an absolute path does provide the full website address.

RELATIVE VS. ABSOLUTE PATHNAMES

Absolute pathname includes the full address (http://…) to the site.

Used to link to an external site.



```
<a href="http:www.yahoo.com">yahoo.com</a>
```

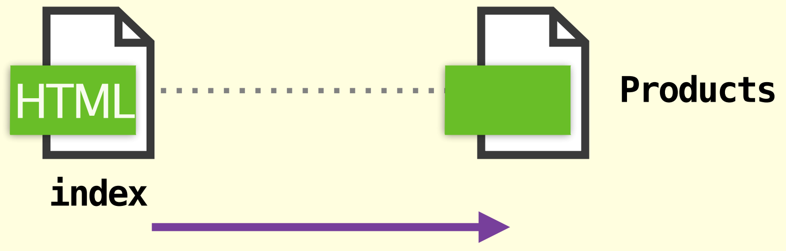
⊢ pathname ⊣

As you may have guessed, an absolute path does provide the full website address.

RELATIVE VS. ABSOLUTE PATHNAMES

A relative pathname does not use the full address.

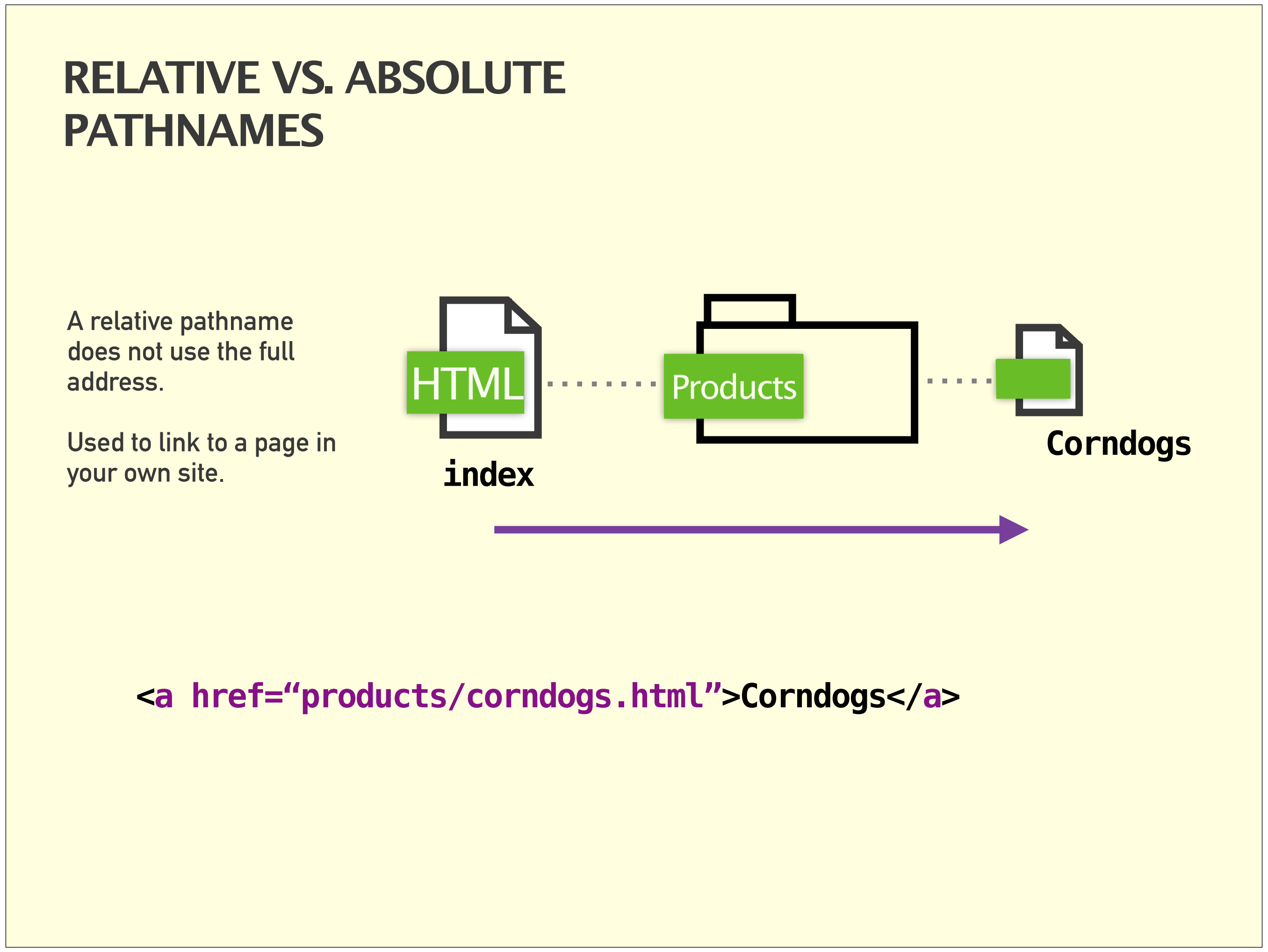
Used to link to a page in your own site.



```
<a href="products.html">Products</a>

<a href="http://www.website.com/products.html">Products</a>
```

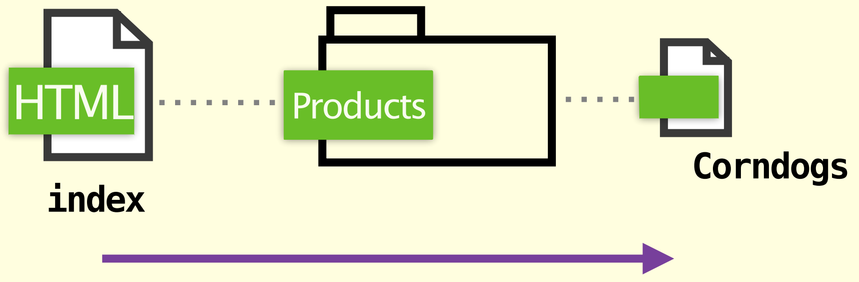
When a user clicks a relative link, the browser takes them to that location on the current site. This link points to a filename, with no path provided. This means that products.html is located in the same folder as the page where this link appears.



RELATIVE VS. ABSOLUTE PATHNAMES

A relative pathname does not use the full address.

Used to link to a page in your own site.

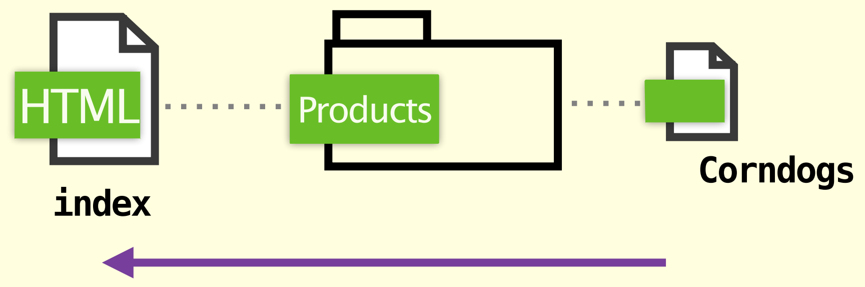


```
<a href="products/corndogs.html">Corndogs</a>
```

How about another example? Let's say we our http://www.website.com domain had a subfolder called products. Inside the products folder is a file called corndogs.html. The full path to this page would be: website.com/products/corndogs.html

RELATIVE VS. ABSOLUTE PATHNAMES

Move up one folder in your relative link by putting two periods and a slash (../) in front of the filename or path.

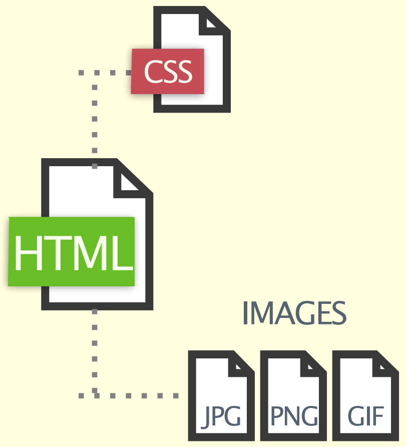


```
<a href="../index.html">Corndogs</a>
```

What if you want to link to a file in a folder above the current folder? You have to tell the browser to move up one folder in your relative link by putting two periods and a slash (../) in front of the filename or path. When the browser sees ../ in front of the filename, it looks in the folder above the current folder. You can use this as many times as you need to. You can also tell the browser to look in a subfolder of the directory above the current one.

RELATIVE VS. ABSOLUTE PATHNAMES

Pathnames also work for CSS stylesheets and images.



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

<img src=“image.jpg” alt=“Image Name”>
```

One of the most common things newcomers to HTML get confused about is linking to other pages and sites, especially when absolute and relative paths come into play. But worry not! Creating links — relative and absolute alike — is actually fairly easy. <link href="css/bootstrap.css" rel="stylesheet">