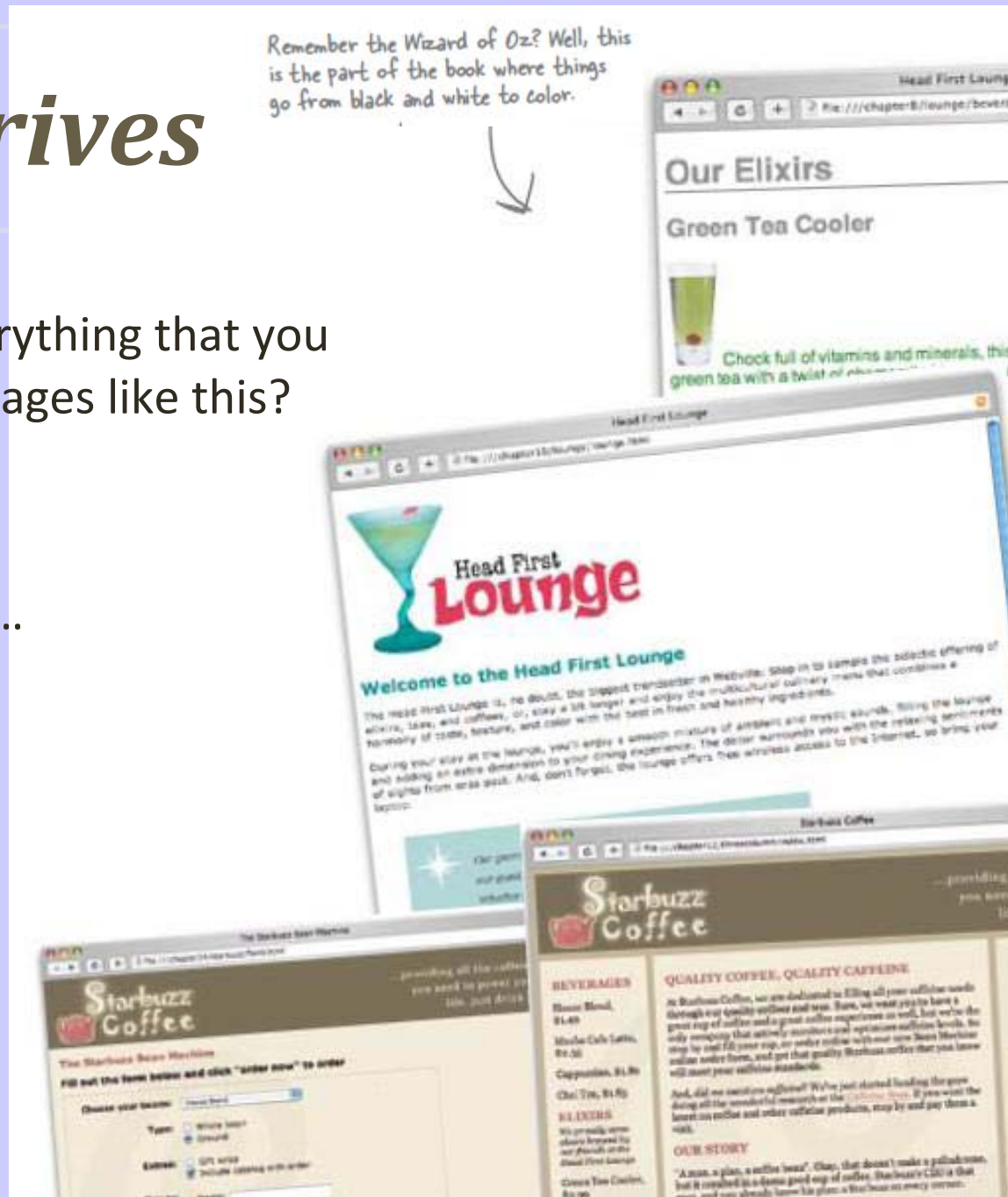


WEB APPLICATIONS

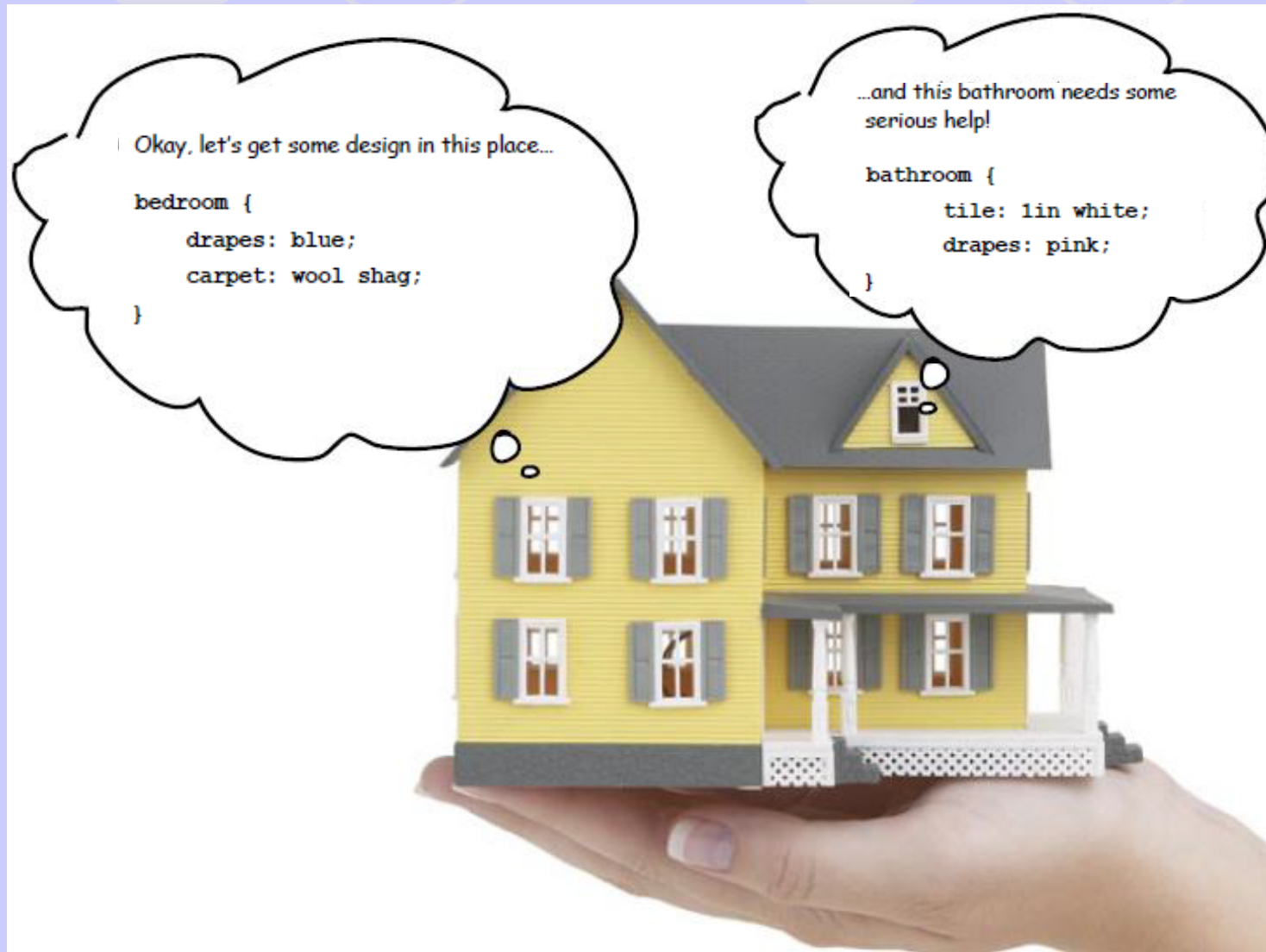
Adding a little style
CSS

Magic arrives

- Do you know everything that you need to do web pages like this?
- You only need...
- ... to learn CSS.
- Let get's started!...



Ideal house



CSS Syntax

- Each **CSS sentence** consists in:
 - A **place**: Like a bedroom
 - A property of this place: Drapes and carpets
 - A style to apply this property: color blue or red
- HTML have elements and CSS:
 - An element: like `<p>`
 - A property of this element: background color
 - Style to apply: red

Using CSS with HTML

The first thing you do is select the element you want to style, in this case the `<p>` element. Notice in CSS, you don't put `<>` around the name.

↓
`p {`

`background-color: red;`

`}`

↑
Place all the styles for the `<p>` element in between `{ }` braces.

Then you specify the property you want to style, in this case the `<p>` element's background color.

And you're going to set the background-color to red.

At the end, put a semicolon.

There's a colon in between the property and its value.

↑
We call the whole thing a RULE.

You could also write the rule like this:

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

Adding more style

- You can add as many properties and values as you like in each CSS rule. Say you wanted to put a border around your paragraphs, too. Here's how you do that:

```
p {  
    background-color: red;  
    border: 1px solid gray;  
}
```

All you have to do is add another property and value.

The <p> element will have a border...

...that is 1 pixel thick, solid, and gray.

There are no Dumb Questions

- **Q: Does every `<p>` element have the same style? Or can I, say, make two paragraphs different colors?**

A: The CSS rules we've used so far define the style for all paragraphs, but CSS is very expressive: it can be used to specify styles in lots of different ways, for lots of different elements—even subsets of elements. You'll see how to make paragraphs two different colors later in this chapter

- How do we know which one are the property of an element?
 - <http://es.html.net/tutorials/css/>
 - <http://www.w3schools.com/css/>

Getting CSS into your HTML

```
<!doctype html>
<html>
  <head>
    <meta charset="utf-8">
    <title>Head First Lounge</title>

    <style>
    </style>

  </head>
  <body>
    <h1>Welcome to the Head First Lounge</h1>
    <p>
      
    </p>
```

Here's what we're interested in: the `<style>` element.

To add CSS style directly to your HTML, add opening and closing style tags in the `<head>` element.

And your CSS rules are going to go right in here.

Adding style

```
<!DOCTYPE html>
<html lang="en">
  <head>
    <meta charset="utf-8">
    <title>Head First Lounge</title>
    <style>
```

Here's the rule that is going to specify the font color of the paragraphs.

We're selecting just the <p> element to apply this style to.

```
p {
  color: maroon;
}
```

The property to change the font color is named "color" (you might think it would be "font-color" or "text-color", but it's not).

We're setting the text to a lovely maroon color that happens to match the lounge couches.

```
</style>
```

```
</head>
```

```
<body>
```

```
<h1>Welcome to the Head First Lounge</h1>
```

```
<p>
```

```

```

```
</p>
```

```
<p>
```

```
Join us any evening for refreshing
<a href="beverages/elixir.html">elixirs</a>,
conversation and maybe a game or two of
<em>Dance Dance Revolution</em>.
Wireless access is always provided;
BYOWS (Bring your own web server).
```

```
</p>
```

```
<h2>Directions</h2>
```

```
<p>
```

```
You'll find us right in the center of downtown
Webville. If you need help finding us, check out our
<a href="about/directions.html">detailed directions</a>.
Come join us!
```


```
</p>
```

```
</body>
```

```
</html>
```

The p selector selects all the paragraphs in the HTML.


Adding style



The screenshot shows a web browser window titled "Head First Lounge" with the address bar displaying "file:///chapter7/lounge/lounge.html". The page content includes a heading "Welcome to the Head First Lounge", an image of four glasses of different colored liquids, a paragraph of text, a heading "Directions", and another paragraph of text. Handwritten annotations with arrows point to specific elements: "Here's our new maroon paragraph text" points to the first paragraph; "Everything else is as it should be: the headings are still black, because all we selected to style were the <p> elements" points to the heading and the first paragraph; "Notice that the color of the links didn't change. Keep that in the back of your mind..." points to the links in the "Directions" paragraph.

Here's our new maroon paragraph text

Welcome to the Head First Lounge



Join us any evening for refreshing [elixirs](#), conversation and maybe a game or two of *Dance Dance Revolution*. Wireless access is always provided; BYOWS (Bring your own web server).

Directions

You'll find us right in the center of downtown Webville. If you need help finding us, check out our [detailed directions](#). Come join us!

Everything else is as it should be: the headings are still black, because all we selected to style were the <p> elements

Notice that the color of the links didn't change. Keep that in the back of your mind...

Style the heading

```
h1 {  
  font-family: sans-serif;  
  color:      gray;  
}  
  
h2 {  
  font-family: sans-serif;  
  color:      gray;  
}  
  
p {  
  color: maroon;  
}
```

Here's the rule to select `<h1>` elements and change the font-family to sans-serif and the font color to gray. We'll talk a lot more about fonts later.

And here's another rule to do the exact same thing to the `<h2>` element

```
h1, h2 {  
  font-family: sans-serif;  
  color:      gray;  
}  
  
p {  
  color: maroon;  
}
```

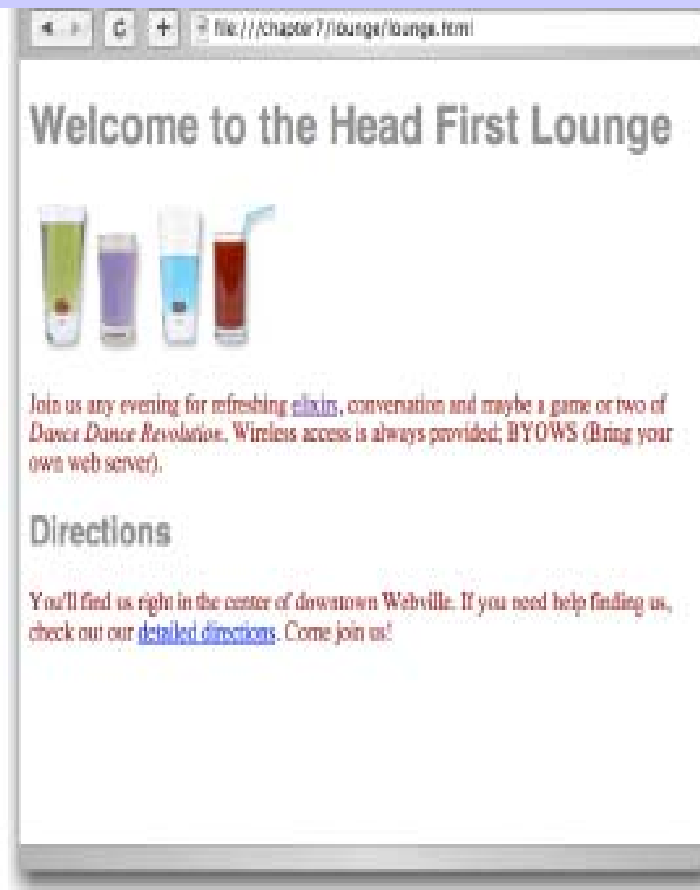
To write a rule for more than one element, just put commas between the selectors, like "h1, h2".

Test drive...

Test drive...

Add this new CSS to your "lounge.html" file and reload. You'll see that with one rule, you've selected both the `<h1>` and `<h2>` headings.

Both of the headings on the page are now styled with a sans-serif font and colored gray.



Let's put a line under the welcome message too

```
border-bottom: 1px solid black;
```

This property controls how the border under an element looks.

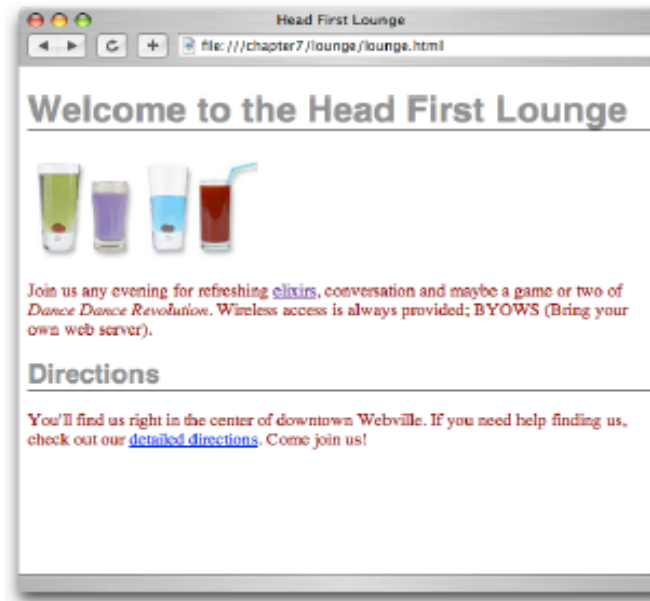
We're going to style the bottom border so that it is a 1-pixel-thick, solid black line.

```
h1, h2 {  
  font-family: sans-serif;  
  color: gray;  
  border-bottom: 1px solid black;  
}  
  
p {  
  color: maroon;  
}
```

Here we're adding a property to change the bottom border for both the <h1> and <h2> elements.

If we do this...
...we get bottom borders on both our headings. Not what we want.

So, how can we set the bottom border on just the <h1> element, without affecting the <h2> element? Do we have to split up the rules again? Turn the page to find out...



A new rule, just for the <h1>

```
h1, h2 {  
  font-family: sans-serif;  
  color: gray;  
}
```

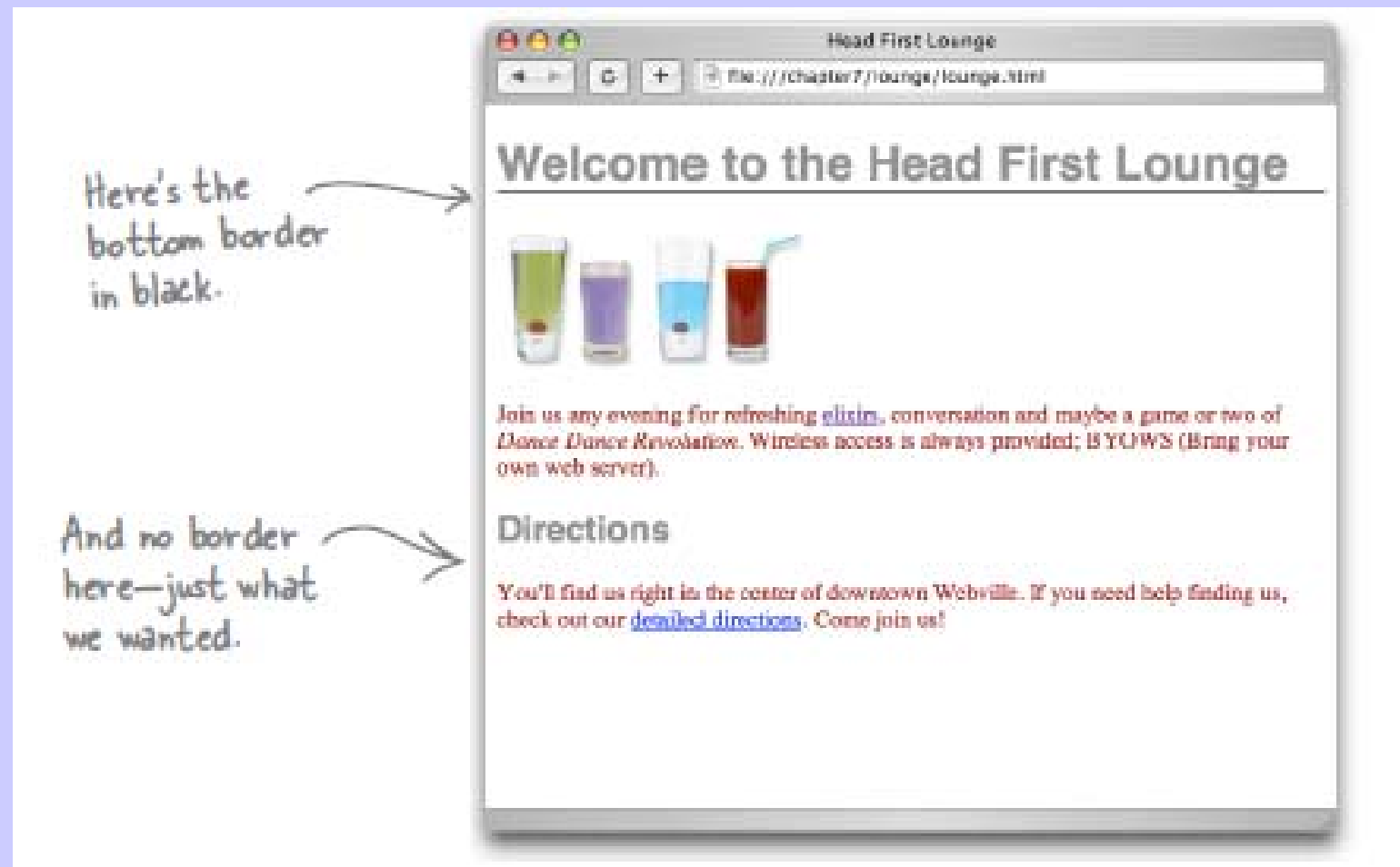
← The first rule stays the same. We're still going to use a combined rule for the font-family and color for both <h1> and <h2>.

```
h1 {  
  border-bottom: 1px solid black;  
}
```

← But now we're adding a second rule that adds another property just to <h1>: the border-bottom property.

```
p {  
  color: maroon;  
}
```

A new rule, just for the <h1>



There are no dumb questions

Q: So how does that work when you have more than one rule for an element?

A: You can have as many rules as you want for an element. Each rule adds to the style information of the rule before it. In general, you try to group together all the common styles between elements, like we did with `<h1>` and `<h2>`, and then any style that is specific to an element, you write in another rule, like we did with the border bottom style for the main heading.

Q: What's the advantage of that approach? Isn't it better to organize each element separately, so you know exactly what styles it has?

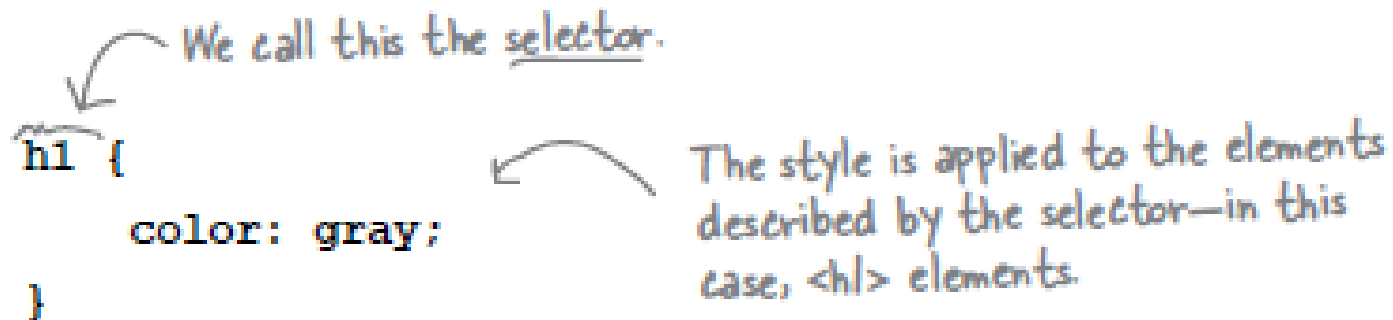
A: Not at all. If you combine common styles together, then if they change, you only have to change them in one rule. If you break them up, then there are many rules you have to change, which is error-prone.

Q: Why do we use a bottom border to underline text? Isn't there an underline style for text?

A: Good question. There is an underline style for text and we could use that instead. However, the two styles have slightly different effects on the page: if you use border-bottom, then the line will extend to the edge of the page. An underline is only shown under the text itself. The property to set text underline is called text-decoration and has a value of "underline" for underlined text. Give it a try and check out the differences.

CSS Selectors

You've seen how to select an element to style it, like this:

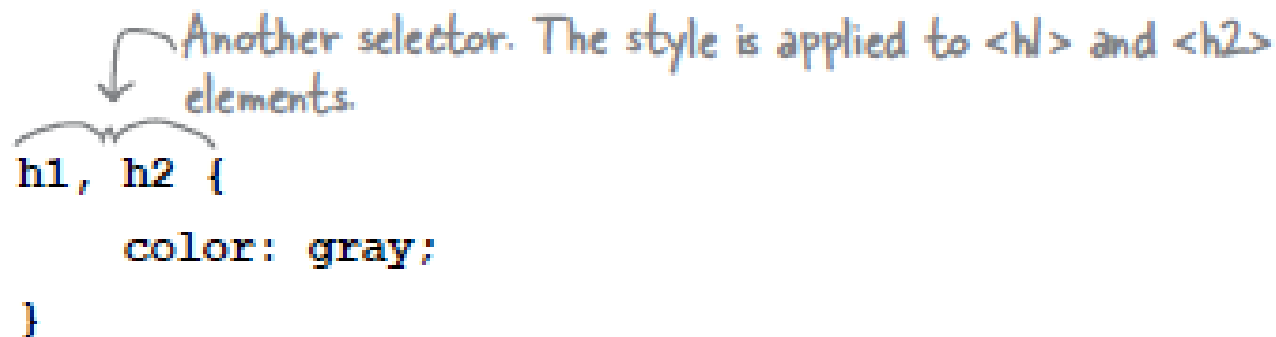


```
h1 {  
  color: gray;  
}
```

We call this the selector.

The style is applied to the elements described by the selector—in this case, `<h1>` elements.

Or how to select more than one element, like this:



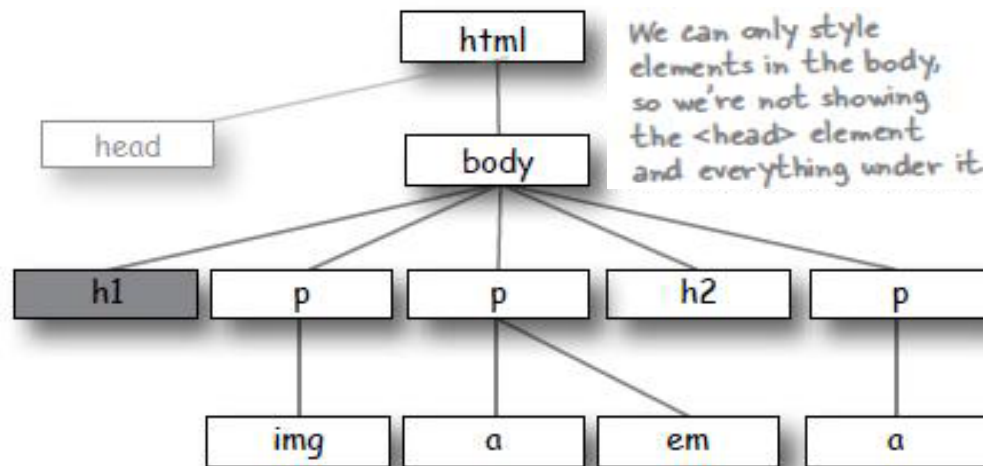
```
h1, h2 {  
  color: gray;  
}
```

Another selector. The style is applied to `<h1>` and `<h2>` elements.

Seeing selectors visually

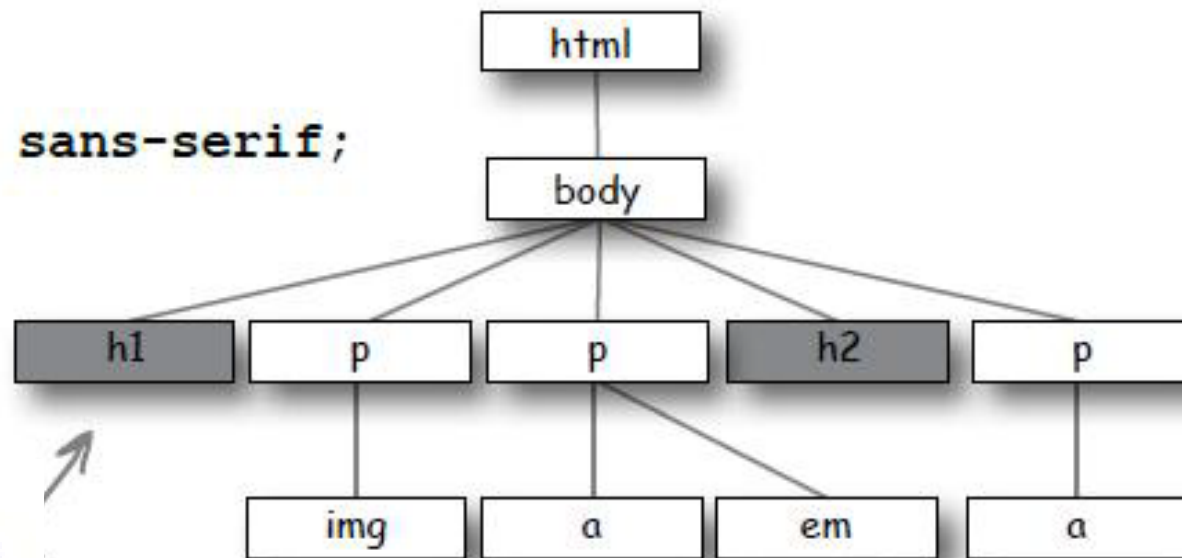
```
h1 {  
  font-family: sans-serif;  
}
```

This selector matches any `<h1>` elements in the page, and there's only one.



Seeing selectors visually

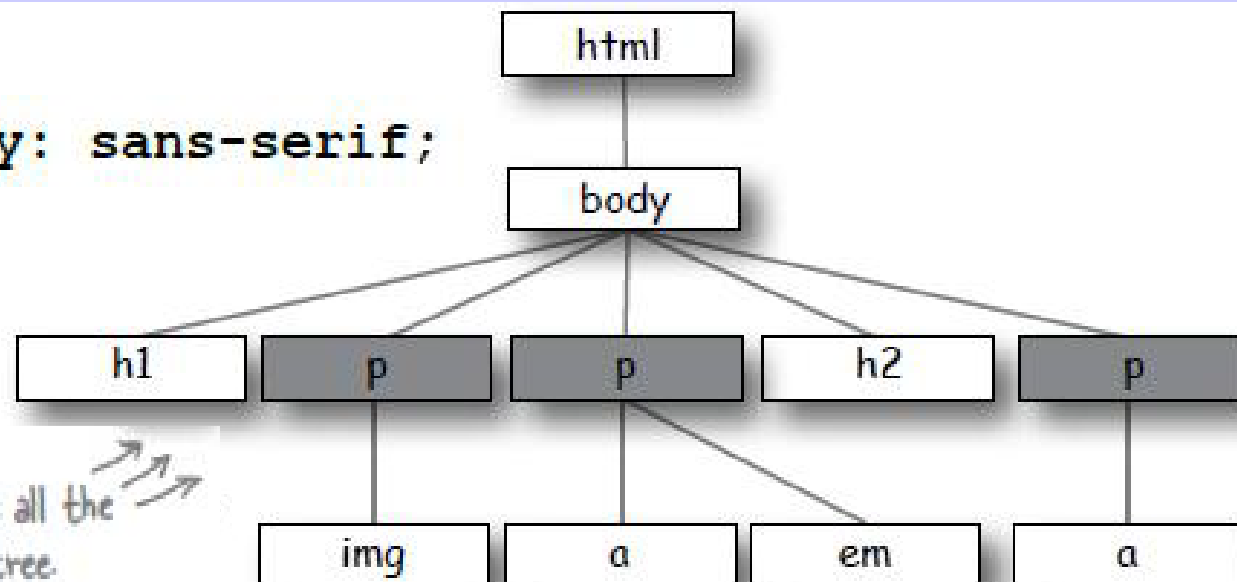
```
h1, h2 {  
  font-family: sans-serif;  
}
```



Now the selector
matches both `<h1>`
and `<h2>` elements

Seeing selectors visually

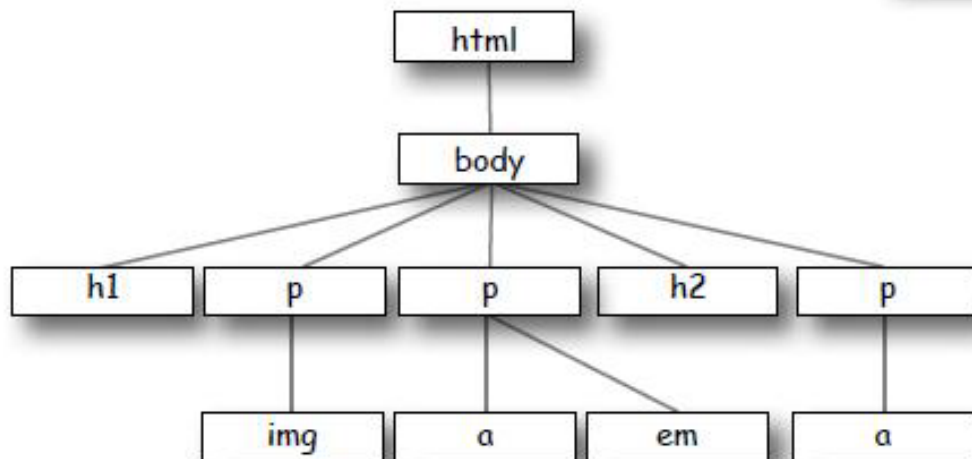
```
p {  
  font-family: sans-serif;  
}
```



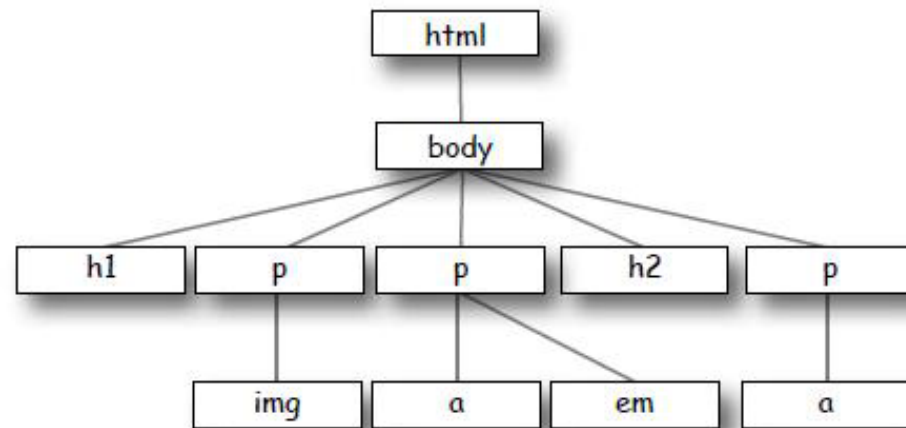
This selector matches all the `<p>` elements in the tree.

Exercise: Color in the elements that are **selected** by these selectors:

```
p, h2 {  
  font-family: sans-serif;  
}
```

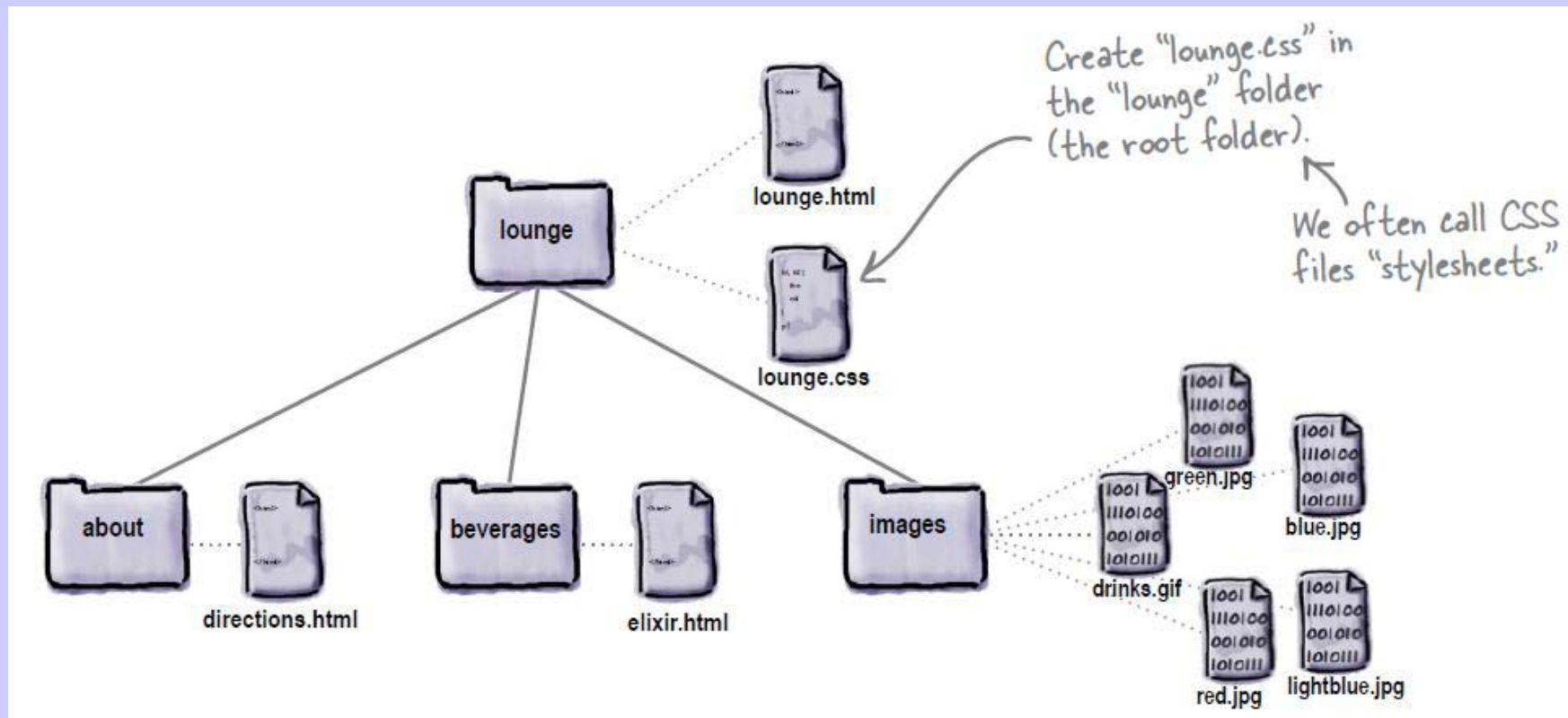


```
p, em {  
  font-family: sans-serif;  
}
```



Creating the lounge.css file

You're going to create a file called "lounge.css" to contain the style rules for all your Head First Lounge pages. To do that, create a new text file named "lounge.css" in your text editor.



Creating the lounge.css file

Now type, or copy and paste from your “lounge.html” file, the CSS rules into the “lounge.css” file. Delete the rules from your “lounge.html” file while you’re at it.

Note that you should not copy the <style> and </style> tags because the “lounge.css” file contains only CSS, not HTML.

```
h1, h2 {  
  font-family: sans-serif;  
  color: gray;  
}
```

```
h1 {  
  border-bottom: 1px solid black;  
}
```

```
p {  
  color: maroon;  
}
```

Your “lounge.css” file
should look like this.
Remember, no <style> tags!

Linking from lounge.html to external stylesheet

```
<!doctype html>
<html>
  <head>
    <meta charset="utf-8">
    <title>Head First Lounge</title>
    <link type="text/css" rel="stylesheet" href="lounge.css">
    <style>
    </style>
  </head>
  <body>
    <h1>Welcome to the Head First Lounge</h1>
    <p>
      
    </p>
    .
    .
    .
  </p>
</body>
</html>
```

Here's the HTML that links to the external stylesheet

You don't need the <style> element anymore—just delete it.

The rest of the HTML is the same.

Element `<link>`

Use the link element to "link in" external information.

The type of this information is "text/css" — in other words, a CSS stylesheet. As of HTML5, you don't need this anymore (it's optional), but you may see it on older pages.

And the stylesheet is located at this href (in this case, we're using a relative link, but it could be a full-blown URL).

```
<link type="text/css" rel="stylesheet" href="lounge.css">
```

The rel attribute specifies the relationship between the HTML file and the thing you're linking to. We're linking to a stylesheet, so we use the value "stylesheet".

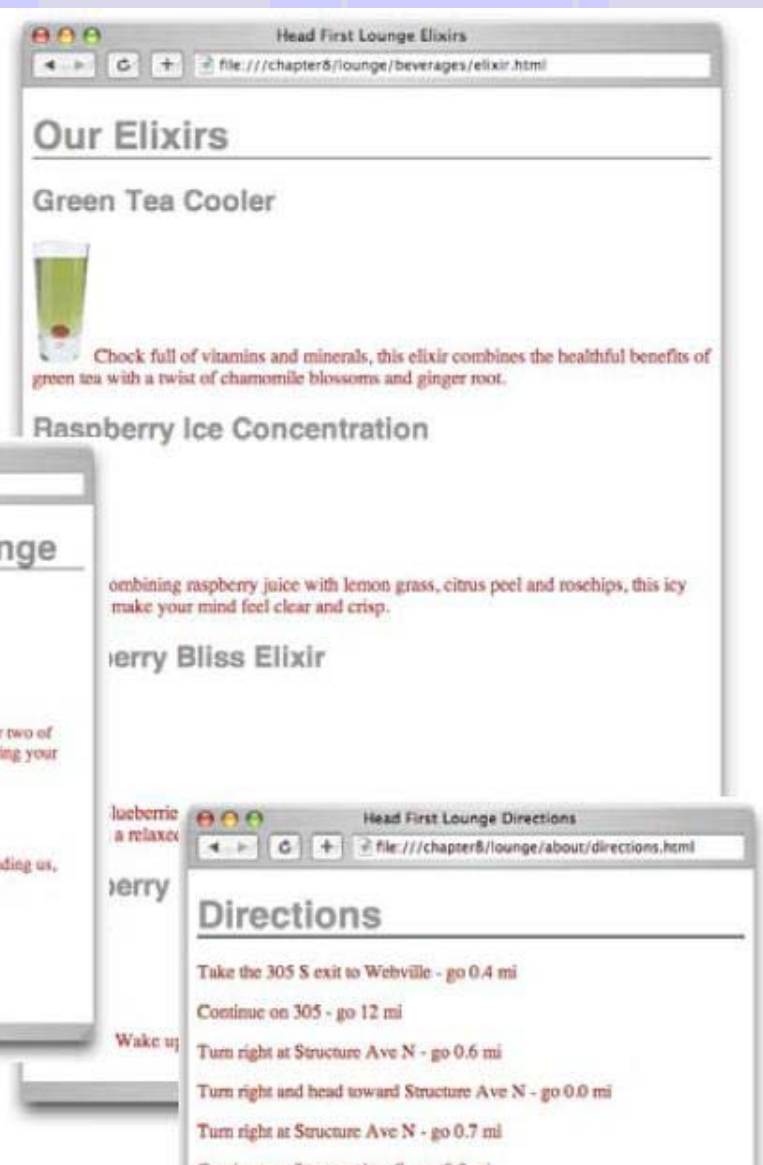
`<link>` is a void element. It has no closing tag.

Exercise

- Link all the html files from lounge project with lounge.css file.
- Pay attention to introduce relative paths.

CSS POWER

- With one file, we have done style to all the web pages.



Exercise: Change the font style to *sans-serif*



¿Is the best option?

- I'm wondering if that is really the best solution. Why are we specifying the font-family for EACH element?
- What if someone added a `<blockquote>` to the page—would we have to then add a rule for that too?
- Can't we just tell the whole page to be sans-serif?

It's time to talk about your inheritance...

- Did you notice when you added the font-family property to your p selector that it also affected the font family of the elements inside the <p> element?

When you added the font-family property to your CSS p selector, it changed the font family of your <p> elements. But it also changed the font family of the two links and the emphasized text.



Inheritance

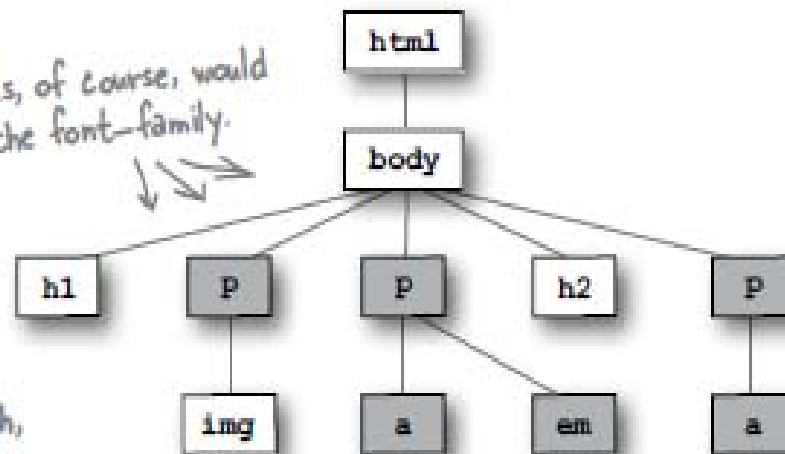
- The elements inside the p element inherit the font-family style from <p>
- In this case, the <a> and elements inherited the font-family style from the <p> element, which is their parent element.
- Elements that they don't have any text, they don't inherit anything.

Inheritance

If we set the font-family of all the `<p>` elements, here are all the elements that would be affected.

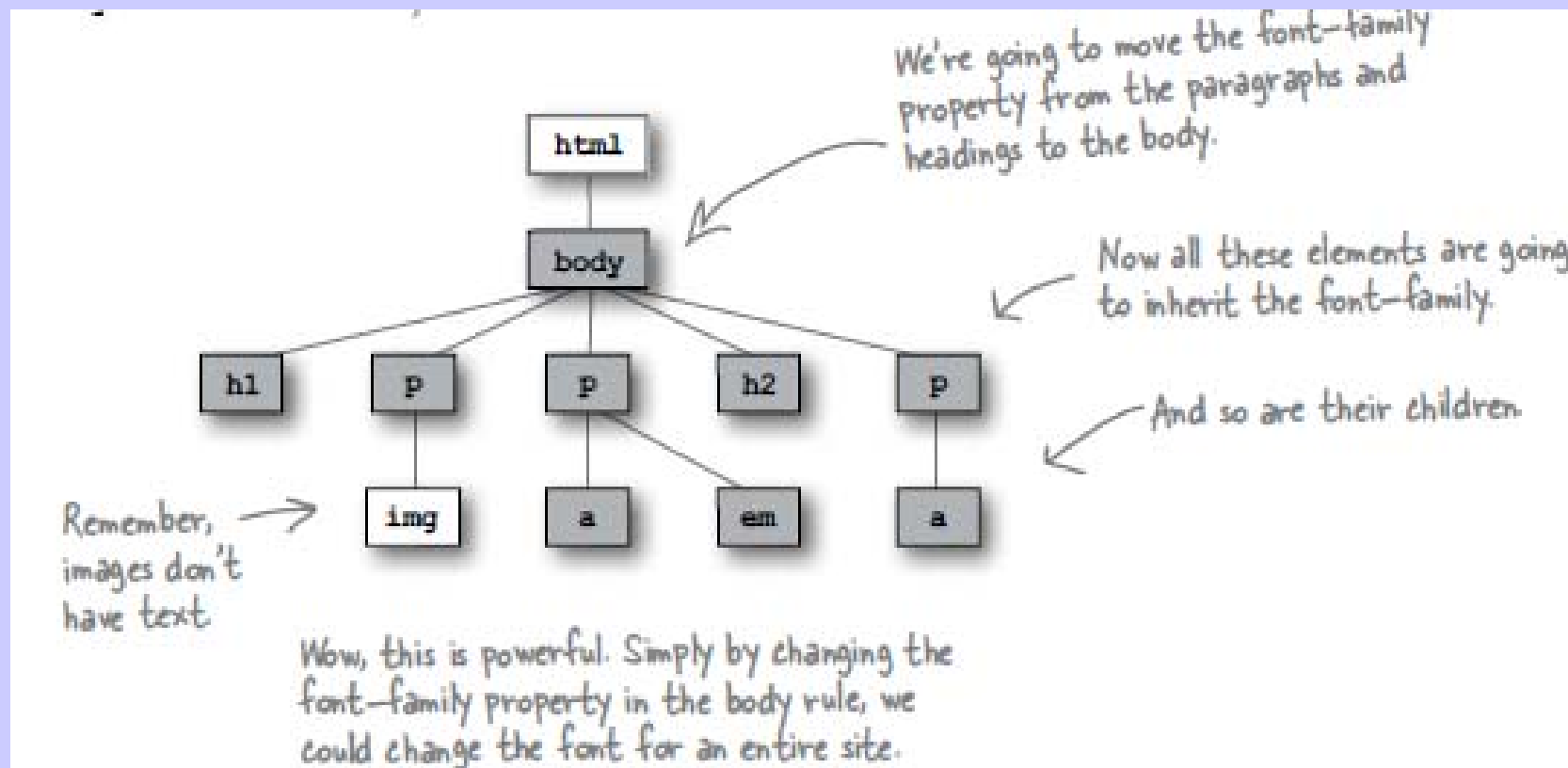
The `<p>` elements, of course, would be styled with the font-family.

The `` element is a child of a paragraph, but it doesn't have any text, so it's not affected.



The `<a>`, ``, and `<a>` elements in the two paragraphs inherit the font-family from their parent elements, the `<p>` elements.

What if we move the font up the family tree?



What are you waiting for...give it a try

```
body {  
  font-family: sans-serif;  
}
```

```
h1, h2 {  
  font-family: sans-serif;  
  color: gray;  
}
```

```
h1 {  
  border-bottom: 1px solid black;  
}
```

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

Here's what you're going to do.

First, add a new rule that selects the `<body>` element. Then add the `font-family` property with a value of `sans-serif`.

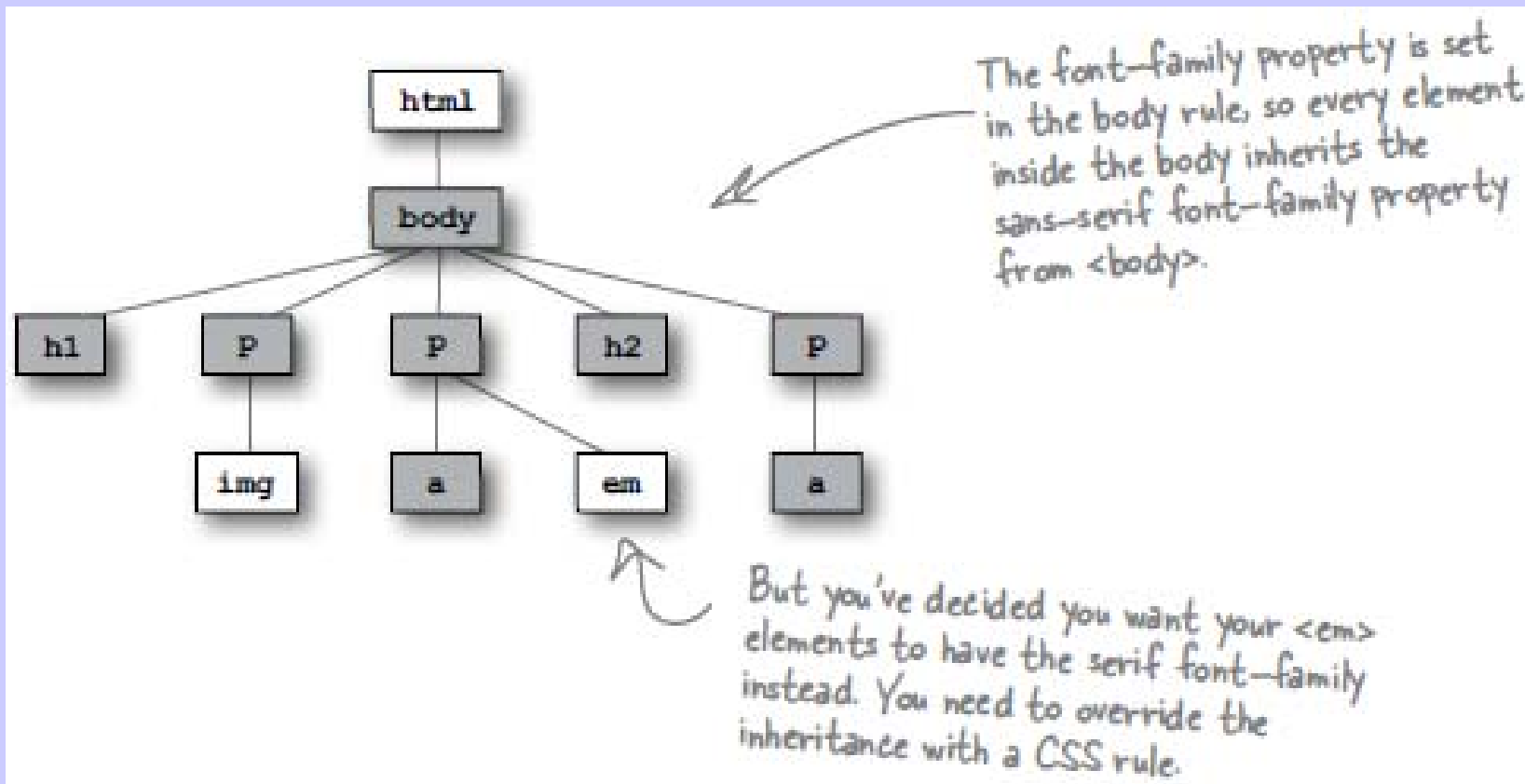
Then, take the `font-family` property out of the `h1`, `h2` rule, as well as the `p` rule.

What happens?

- Surprise, surprise!. This doesn't look any difference at all.
- All you've done is move the sans-serif from up into the body rule and let all the other elements inherit that.
- Okay, so now that the whole site is set to sans-serif with the body selector, what if I want one element to be a different font? Do I have to take the font-family out of the body and add rules for every element separately again?

Overriding inheritance

- You could decide that you want `` elements to use the serif font instead.



Overriding inheritance

```
body {  
    font-family: sans-serif;  
}  
  
h1, h2 {  
    color: gray;  
}  
  
h1 {  
    border-bottom: 1px solid black;  
}  
  
p {  
    color: maroon;  
}  
  
em {  
    font-family: serif;  
}
```

To override the font-family property inherited from body, add a new rule selecting em with the font-family property value set to serif.

The are not Dumb questions

Q: How does the browser know which rule to apply to `` when I'm overriding the inherited value?

A: With CSS, the most specific rule is always used. So, if you have a rule for `<body>`, and a more specific rule for `` elements, it is going to use the more specific rule. We'll talk more later about how you know which rules are most specific.

Q: How do I know which CSS properties are inherited and which are not?

A: This is where a good reference really comes in handy, like O'Reilly's *CSS Pocket Reference*. In general, all of the styles that affect the way your text looks, such as font color (the color property), the font-family, as you've just seen, and other font-related properties such as font-size, font-weight (for bold text), and font-style (for italics) are inherited. Other properties, such as border, are not inherited, which makes sense, right? Just because you want a border on your `<body>` element doesn't mean you want it on *all* your elements. A lot of the time, you can follow your common sense (or just try it and see), and you'll get the hang of it as you become more familiar with the various properties and what they do.

Q: Can I always override a property that is being inherited when I don't want it?

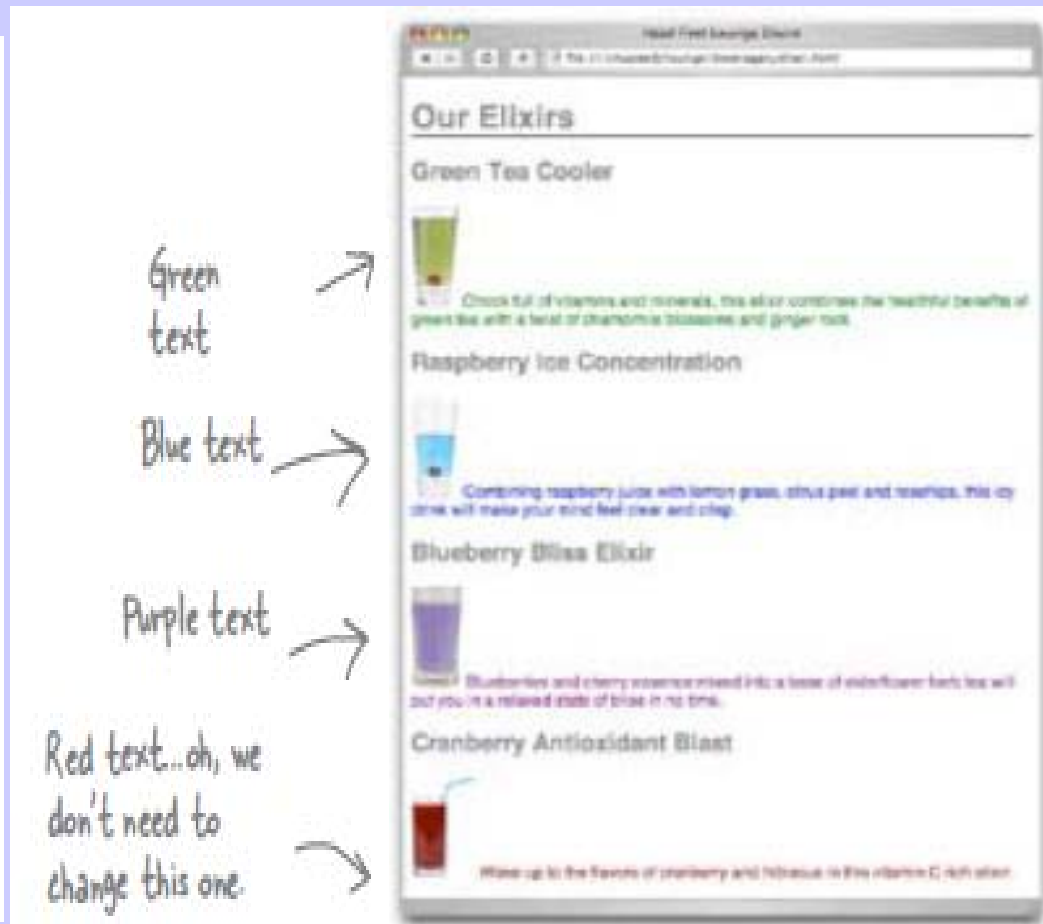
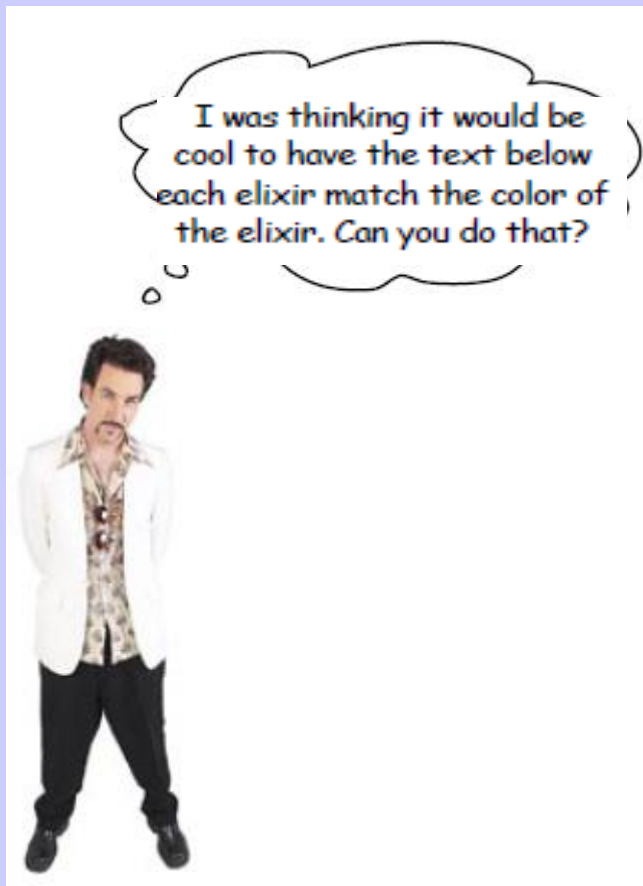
A: Yes. You can always use a more specific selector to override a property from a parent.

Q: This stuff gets complicated. Is there any way I can add comments to remind myself what the rules do?

A: Yes. To write a comment in your CSS, just enclose it between `/*` and `*/`. For instance:

```
/* this rule selects all  
paragraphs and colors them  
blue */
```

Applying styles individually

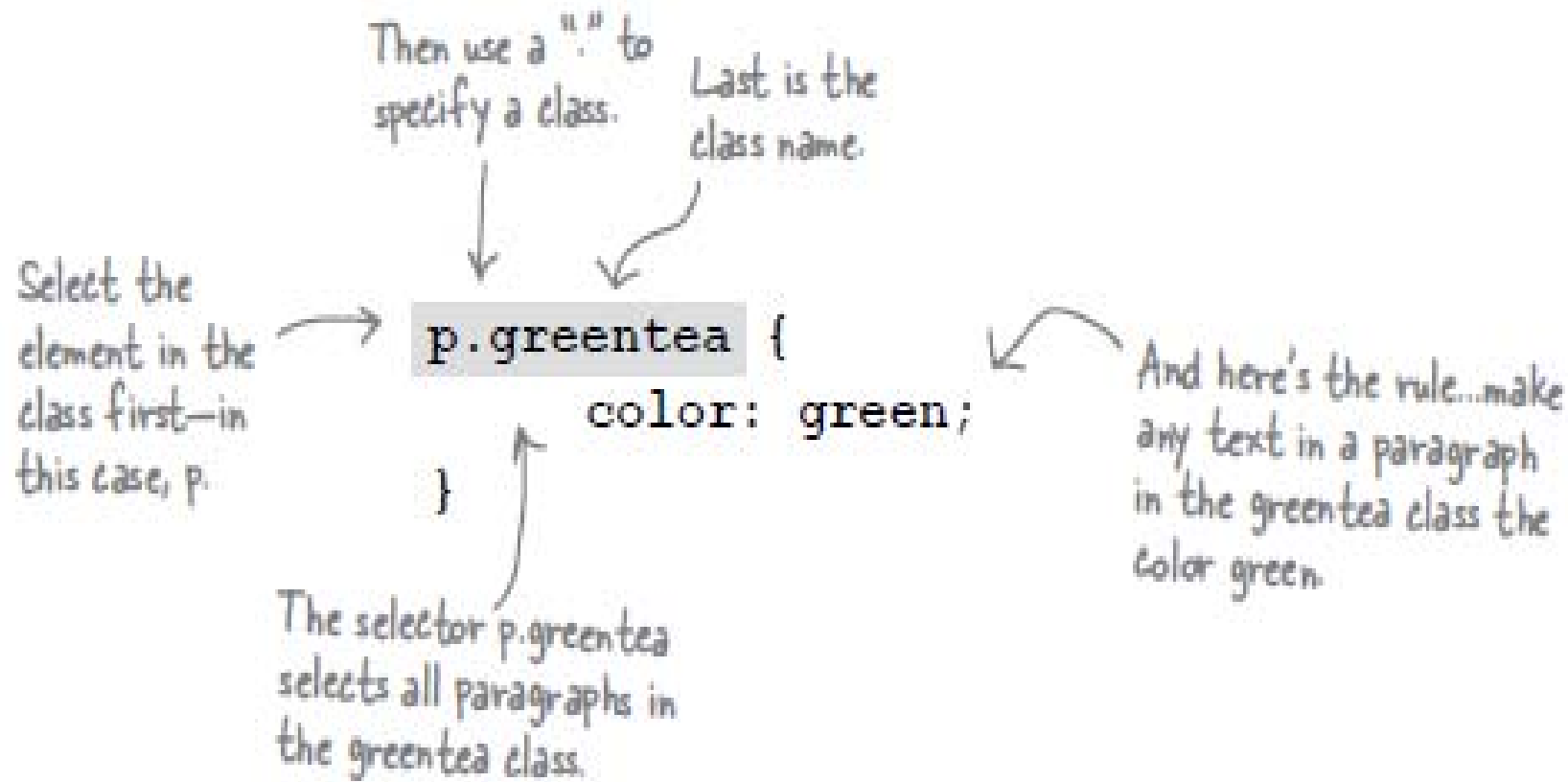


Adding an element to the greentea class

```
<html>
  <head>
    <meta charset="utf-8">
    <title>Head First Lounge Elixirs</title>
    <link type="text/css" rel="stylesheet" href="../lounge.css">
  </head>
  <body>
    <h1>Our Elixirs</h1>
    <h2>Green Tea Cooler</h2>
    <p class="greentea">
      
      Chock full of vitamins and minerals, this elixir
      combines the healthful benefits of green tea with
      a twist of chamomile blossoms and ginger root.
    </p>
    <h2>Raspberry Ice Concentration</h2>
    <p>
      
      Combining raspberry juice with lemon grass,
      citrus peel and rosehips, this icy drink
      will make your mind feel clear and crisp.
    </p>
```

To add an element to a class, just add the attribute "class" along with the name of the class, like "greentea".

Creating a class selector



Creating a class selector

```
body {  
    font-family: sans-serif;  
}  
  
h1, h2 {  
    color: gray;  
}  
  
h1 {  
    border-bottom: 1px solid black;  
}  
  
p {  
    color: maroon;  
}  
  
p.greentea {  
    color: green;  
}
```



Exercise

- Add a new class to each elixir and Apply to the relevant paragraph.

Adding other selector

- if you wanted to do the same to all <blockquote>s?

```
blockquote.greentea, p.greentea {  
    color: green;  
}
```

Just add another selector to handle <blockquote>s that are in the greentea class. Now this rule will apply to <p> and <blockquote> elements in the greentea class.

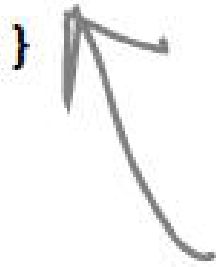
- And in your HTML you'd write:
 - <blockquote class="greentea">

Adding other selector

- If we want to add the same style for elements `<h1>`, `<h2>` y `<h3>`? Are we have to write a enormous selector?

```
.greentea {
```

```
    color: green;
```



If you leave out all the element names, and just use a period followed by a class name, then the rule will apply to all members of the class.

- We define the class like:
 - `<element class="greentea">`

Can an element be in more than one class?

- Yes, elements can be in more than one class. It's easy to put an element into more than one class. Say you want to specify a `<p>` element that is in the `greentea`, `raspberry`, and `blueberry` classes. Here's how you would do that in the opening tag:
 - `<p class="greentea bluetea redtea">`
- All the classes separate by an space, order doesn't matter.
- What style is applying? The more specific. In case there are two at the same specific level, so the last one.
- But in general, it have more sense if each class affects different properties:
 - `<h1 class="products special">`
- Where ***products*** define ***font_size*** and ***weight*** properties and ***special*** define the text color

Gold rules – Applying styles

1. We want to apply a style to one element... Does exist any selector to modify the property of the element?
 - **Yes:** The value in the property is his value
 - **No:** We trust in inheritance
2. **Inheritance:** We look their parent elements and the parents of their parents, ¿Are they have the property?
3. If any element have the property, the element will have the default value of the browser.
4. What happens if they have multiply selectors?

Multiple selectors - conflict

- There are multiple selectors that match this element and define the same color property. That's what we call a *conflict*. Which rule breaks the tie? Well, if one rule is more *specific* than the others, then it wins.

```
p { color: black; }
```

Here's a rule that selects any old paragraph element

```
.greentea { color: green; }
```

This rule selects members of the greentea class. That's a little more specific.

```
p.greentea { color: green; }
```

And this rule selects only paragraphs that are in the greentea class, so that's even more specific.

```
p.raspberry { color: blue; }
```

These rules also select only paragraphs in a particular class. So they are about the same in specificity as the p.greentea rule.

```
p.blueberry { color: purple; }
```


Multiple selectors



Exercise

In your "elixir.html" file, change the greentea paragraph to include all the classes, like this:

```
<p class="greentea raspberry blueberry">
```

Save and reload. What color is the Green Tea Cooler paragraph now? _____

Next, reorder the classes in your HTML:

```
<p class="raspberry blueberry greentea">
```

Save and reload. What color is the Green Tea Cooler paragraph now? _____

Next, open your CSS file and move the p.greentea rule to the bottom of the file.

Save and reload. What color is the Green Tea Cooler paragraph now? _____

Finally, move the p.raspberry rule to the bottom of the file.

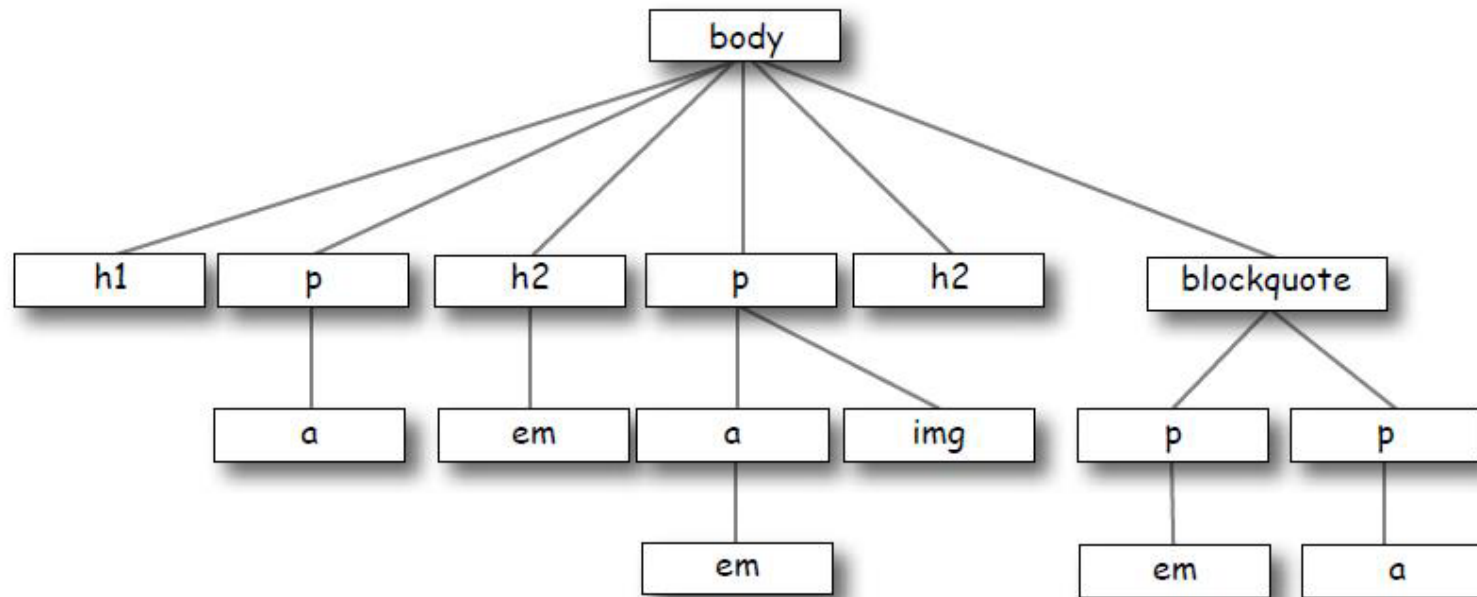
Save and reload. What color is the Green Tea Cooler paragraph now? _____

After you've finished, rewrite the green tea element to look like it did originally:

```
<p class="greentea">
```

Save and reload. What color is the Green Tea Cooler paragraph now? _____

Exercise: Who gets the inheritance?



```
body {  
    color: green;  
}  
  
p {  
    color: black;  
}
```

←
Here's the CSS. Use this to determine which of the above elements hit the jackpot and get the green (color).

Exercise: Find mistakes

```
<style>

body {
    background-color: white

h1, {
    gray;
    font-family: sans-serif;
}

h2, p {
    color:

<em> {
    font-style: italic;
}

</style>
```

How do we validate CSS

- <http://jigsaw.w3.org/css-validator/>

Deutsch English Español Français 한국어 Italiano Nederlands 日本語 Polski Português Русский فارسی Svenska Български Українська Čeština Romanian 简体中文

W3C® CSS Validation Service
Verifica Hojas de Estilo en Cascada (CSS) y documentos (X)HTML con hojas de estilo

mediante URI mediante Carga de Archivo mediante Entrada directa


Validar mediante URI

Introduce la URI de un documento (HTML con CSS o sólo CSS) que desees validar :

Dirección :

► Más opciones

Check

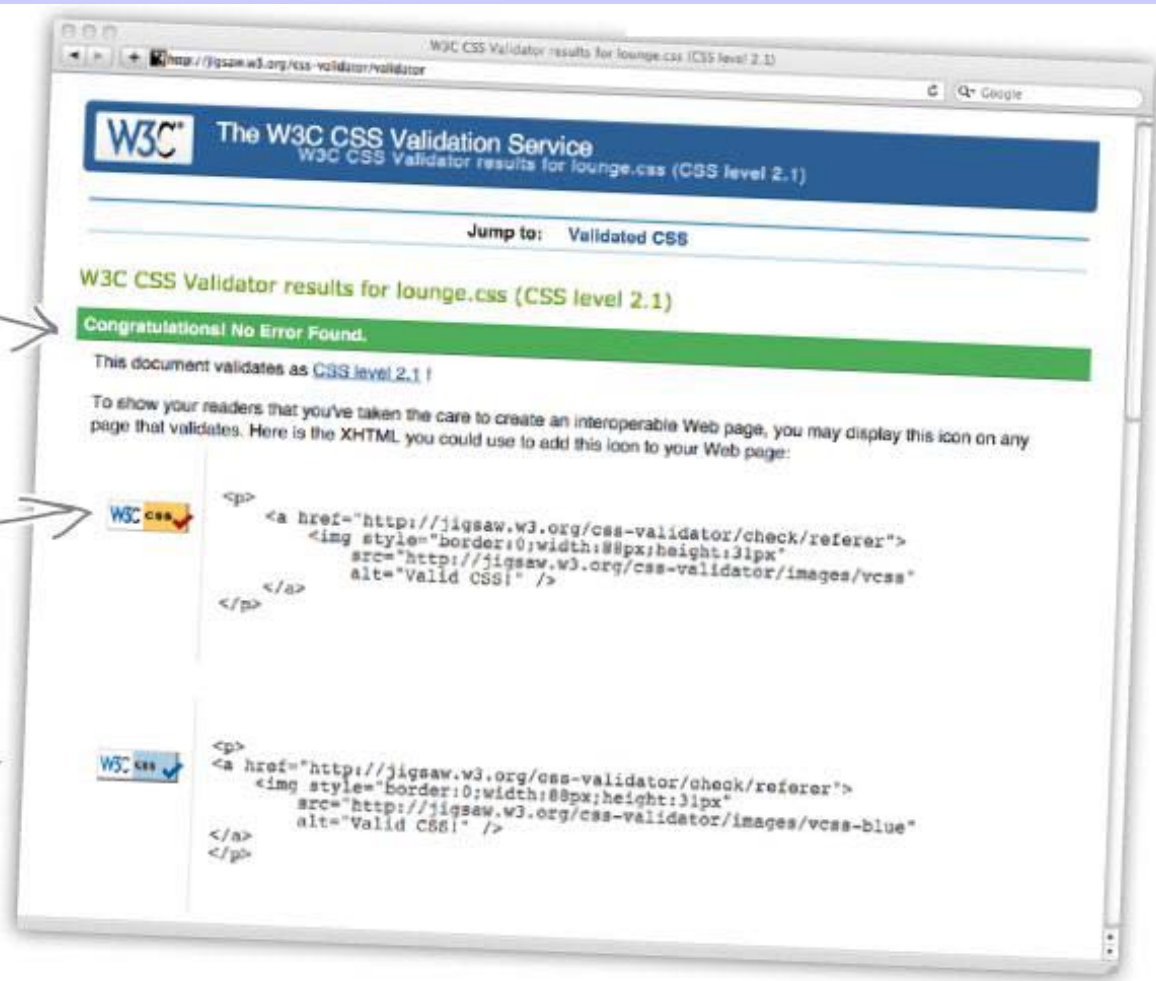
 The W3C CSS validator is developed with assistance from the Mozilla Foundation, and supported by community donations. [Donate](#) and help us build better tools for a better web. 

Nota: Si desees validar tu hoja de estilo CSS incrustada en un documento (X)HTML, deberías antes [comprobar que el \(X\)HTML utilizado es válido](#).

CSS Validator

Yay! Our CSS validates as CSS 2.1 (the validator hasn't upgraded to CSS 3 yet, but if it has by the time you read this, it should still validate).

Here are some icons you can put on your web page if you want to show off that your CSS validates. (You can get similar icons for validated HTML, too.)



Sopa de propiedades

Use color to set the font color of text elements.

↪ **color**

This property controls the weight of text. Use it to make text bold.

↪ **font-weight**

↪ **left**
This is how you tell an element how to position its left side.

This property sets the space between lines in a text element.

↪ **line-height**

top
Controls the position of the top of the element.

text-align
Use this property to align your text to the left, center, or right.

↪ **letter-spacing**
This lets you set the spacing between letters. Like this

background-color

↪ This property controls the background color of an element.

Use this property for italic or oblique text.

↪ **font-style**

↪ **border**

This property puts a border around an element. You can have a solid border, a ridged border, a dotted border...

↪ **margin**

If you need space between the edge of an element and its content, use margin.

Makes text bigger or smaller.

↪ **font-size**

↪ **list-style**
This property lets you change how list items look in a list.

Use this property to put an image behind an element.

↪ **background-image**

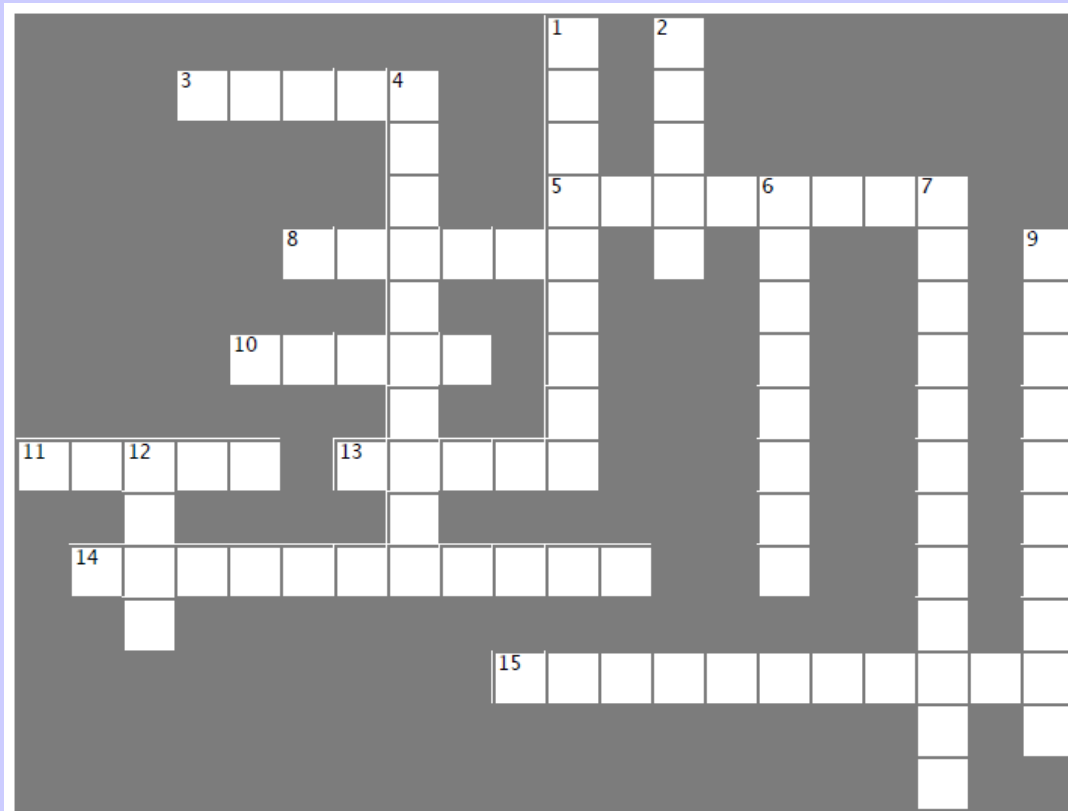
Aquí tenemos algunas de las propiedades que se puede dar estilo con CSS.

Across

3. Styles are defined in these.
5. Selects an element.
8. Each rule defines a set of properties and _____.
10. Defines a group of elements.
11. Property that represents font color.
13. Ornamental part of some fonts.
14. How elements get properties from their parents.
15. Property for font type.

Down

1. Fonts without serifs.
2. You can place your CSS inside these tags in an HTML file.
4. An external style file is called this.
6. With inheritance, a property set on one element is also passed down to its _____.
7. Won this time because they used external stylesheets.
9. They really wanted some style.
12. Use this element to include an external stylesheet.





BULLET POINTS

- CSS contains simple statements, called rules.
- Each rule provides the style for a selection of HTML elements.
- A typical rule consists of a selector along with one or more properties and values.
- The selector specifies which elements the rule applies to.
- Each property declaration ends with a semicolon.
- All properties and values in a rule go between { } braces.
- You can select any element using its name as the selector.
- By separating element names with commas, you can select multiple elements at once.
- One of the easiest ways to include a style in HTML is the `<style>` tag.
- For HTML and for sites of any complexity, you should link to an external stylesheet.
- The `<link>` element is used to include an external stylesheet.
- Many properties are inherited. For instance, if a property that is inherited is set for the `<body>` element, all the `<body>`'s child elements will inherit it.
- You can always override properties that are inherited by creating a more specific rule for the element you'd like to change.
- Use the class attribute to add elements to a class.
- Use a "." between the element name and the class name to select a specific element in that class.
- Use ".classname" to select any elements that belong to the class.
- You can specify that an element belongs to more than one class by placing multiple class names in the class attribute with spaces between the names.
- You can validate your CSS using the W3C validator, at <http://jigsaw.w3.org/css-validator>.