

Cap 10

div and span

Div



Div

Here's the handout menu with the elixir specials. Wow, the design is a lot different than the rest of the page: it's thin, the text is centered, there are red headings, an aquamarine border around the whole thing, and even some cocktail graphics at the top.



Weekly Elixir Specials



Lemon Breeze

The ultimate healthy drink, this elixir combines herbal botanicals, minerals, and vitamins with a twist of lemon into a smooth citrus wonder that will keep your immune system going all day and all night.



Chai Chiller

Not your traditional chai, this elixir mixes maté with chai spices and adds an extra chocolate kick for a caffeinated taste sensation on ice.



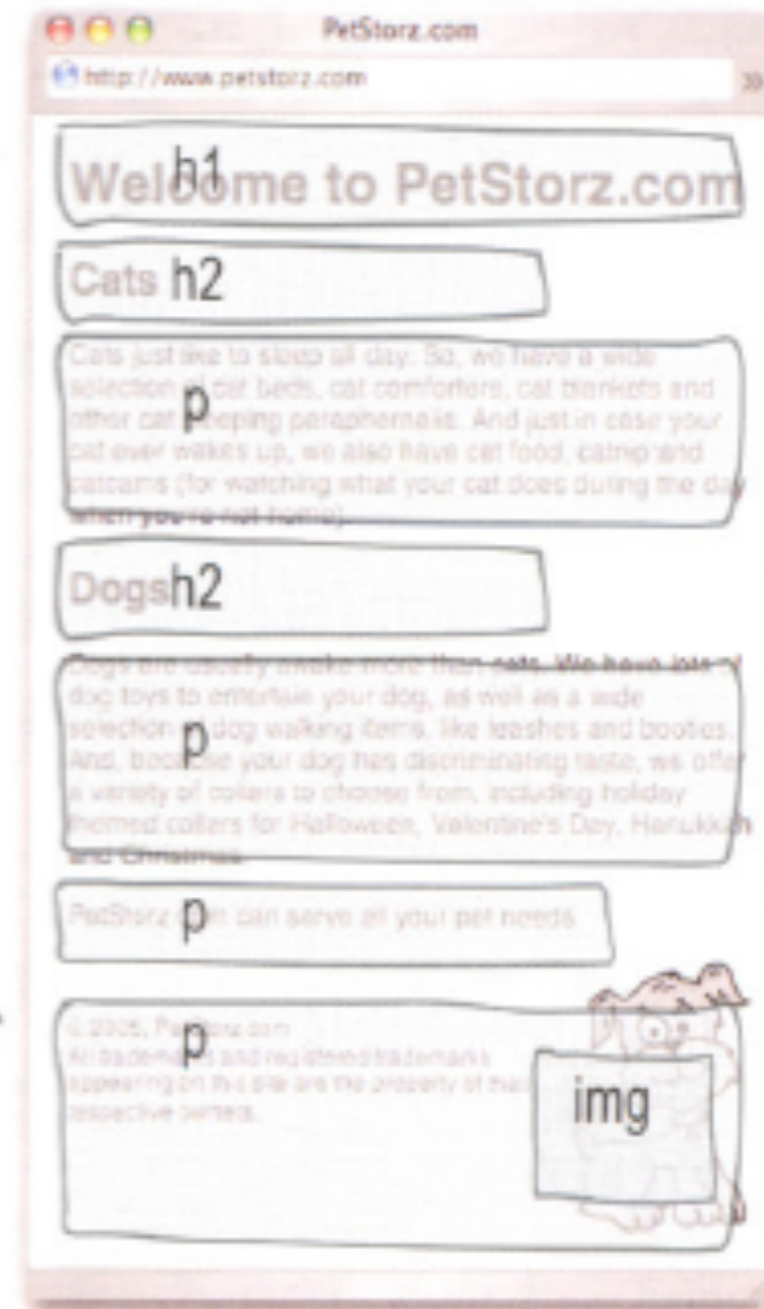
Black Brain Brew

Want to boost your memory? Try our Black Brain Brew elixir, made with black oolong tea and just a touch of espresso. Your brain will thank you for the boost.

Join us any evening for these and all our wonderful elixirs.

How we can divide a page into logical section

We've drawn an outline of the PetStorz page.



This is a pretty normal looking page: lots of headings, paragraphs, and an image in there.

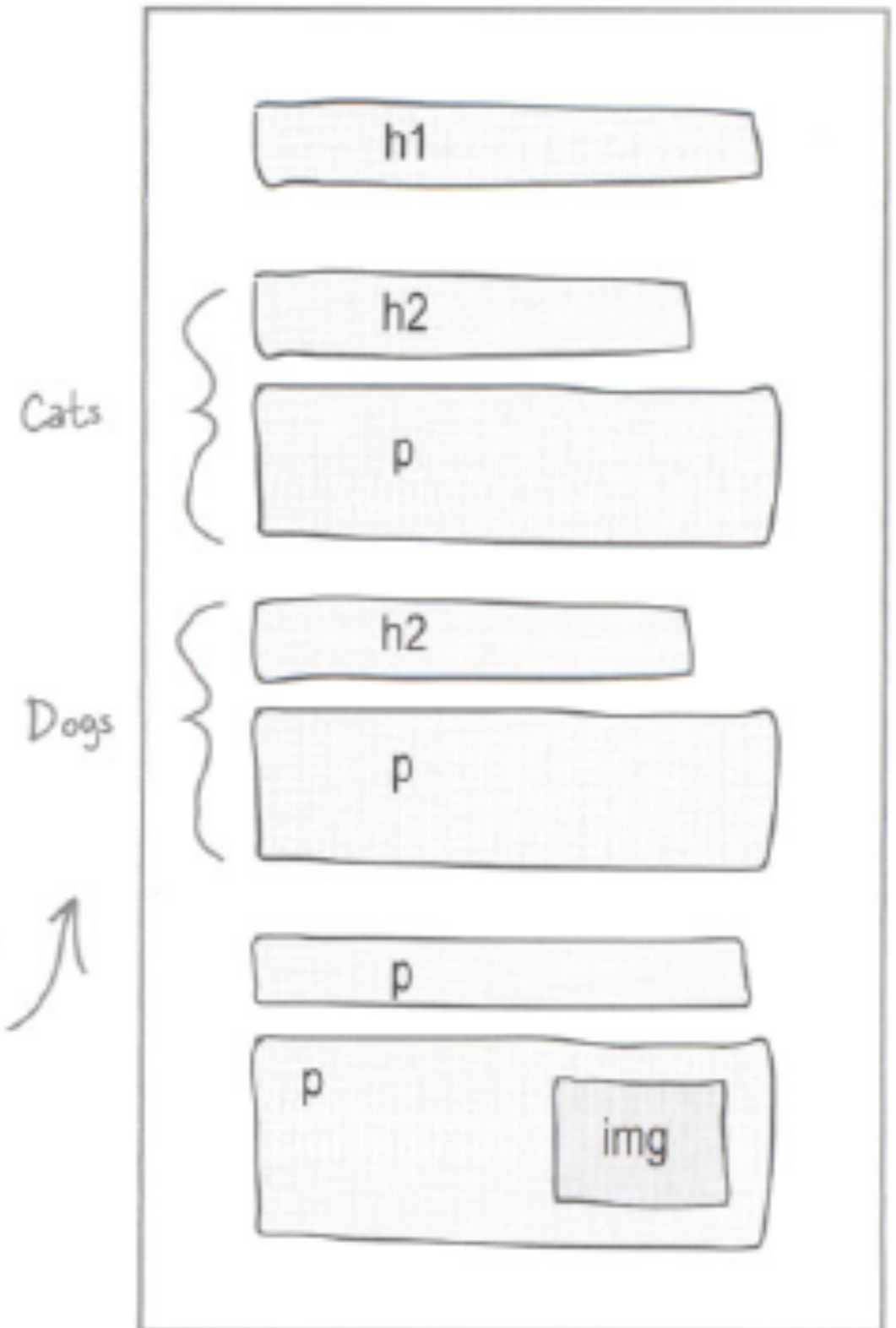


But by just focussing on the structure of the page, you can't really tell a whole lot about the page. What elements make up the header? Is there a footer on the page? What are the content areas?

Identify your logical sections

The PetStorcz page has two main content areas, one for cats, and one for dogs. It has some other areas too, but we'll come back to those.

In this case, both the cats and dogs sections consist of two elements, a heading and a paragraph. But often these groupings can contain many more elements.

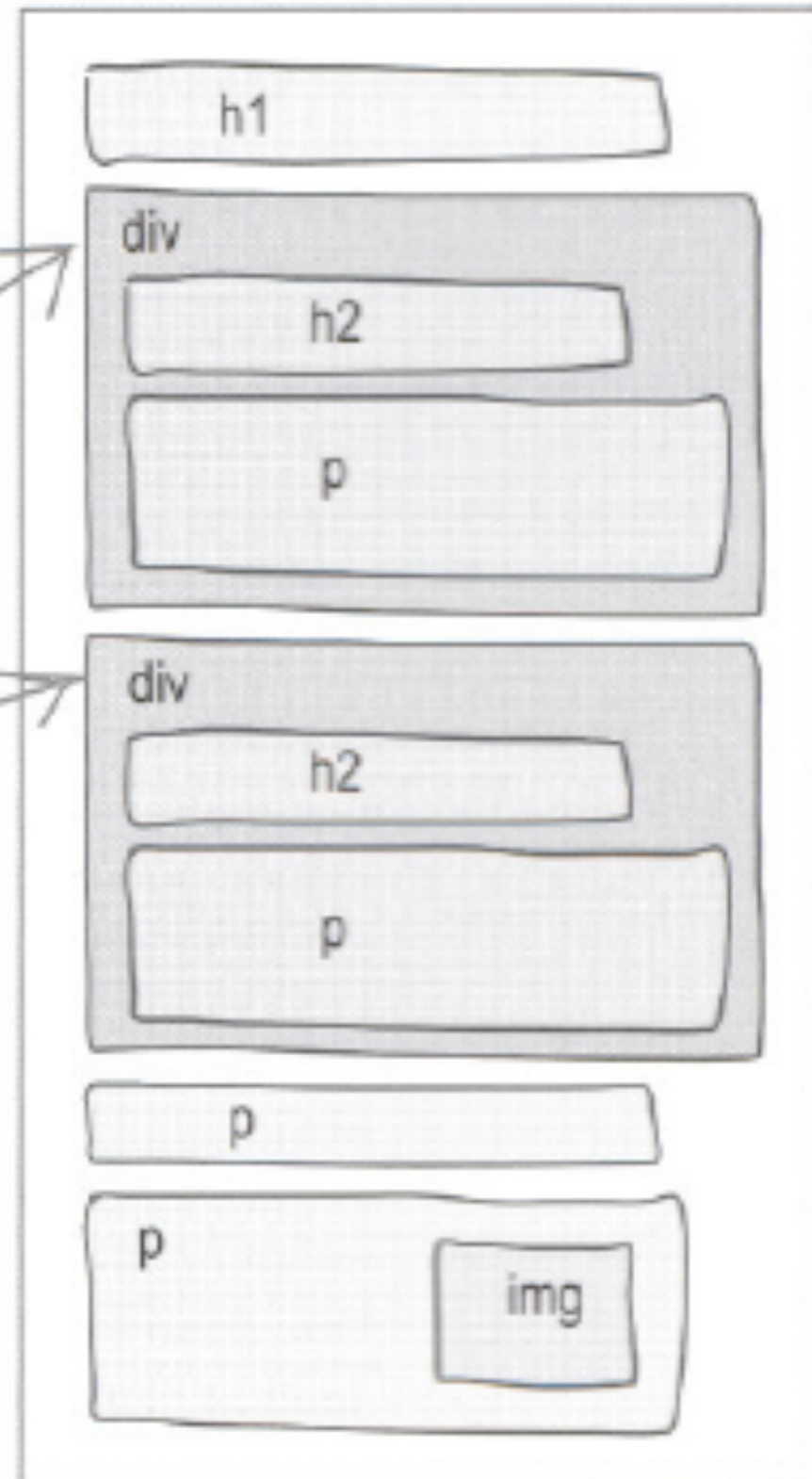


Using `<div>`s to mark sections

Let's nest the elements in each grouping in a `<div>` element.

Here's our cat group.

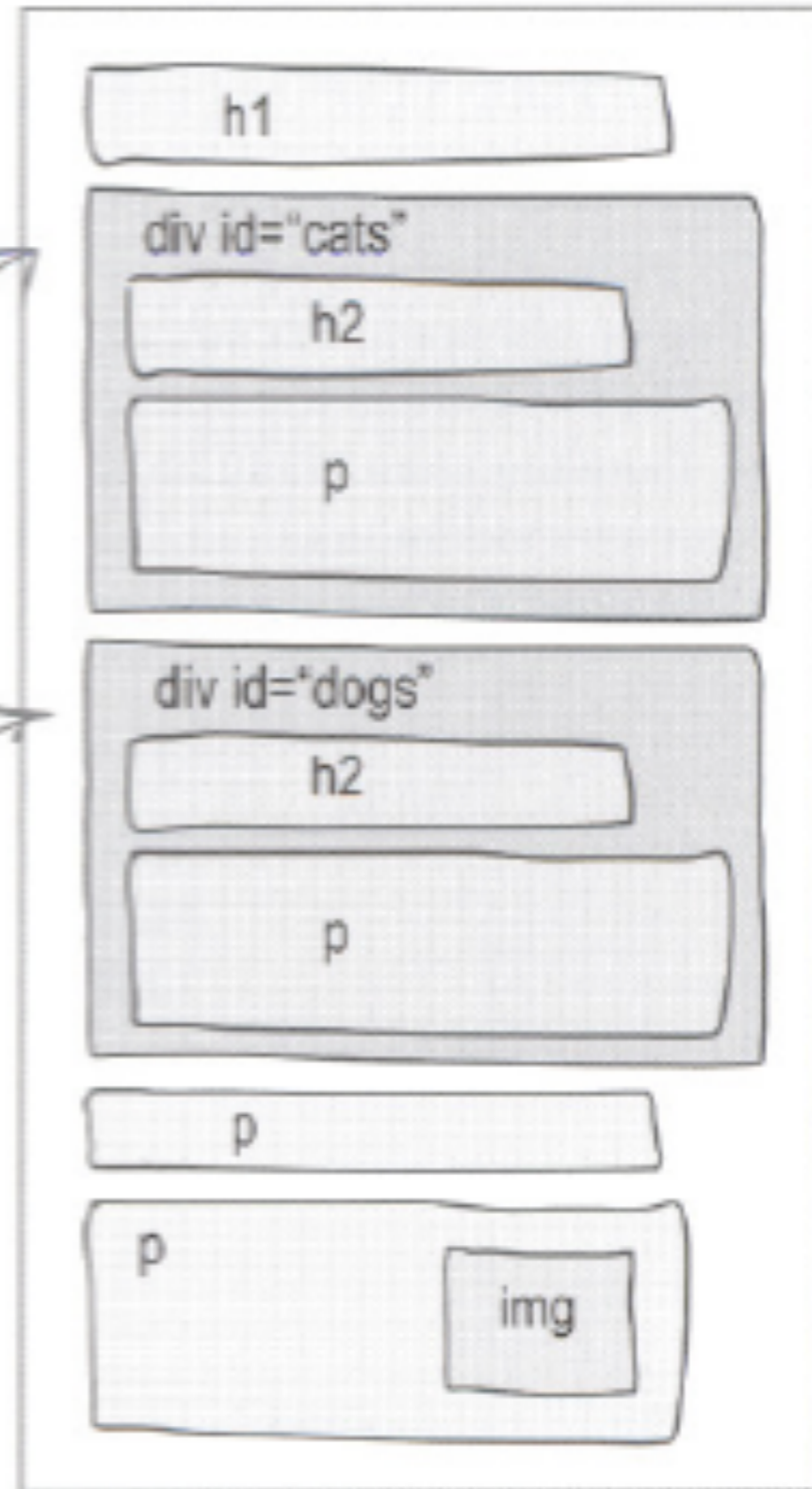
And here's our dog group.



Labelling the <div>s

Here we've added an id of "cats" to the first <div> to indicate what the logical section is for.

And likewise for dogs.

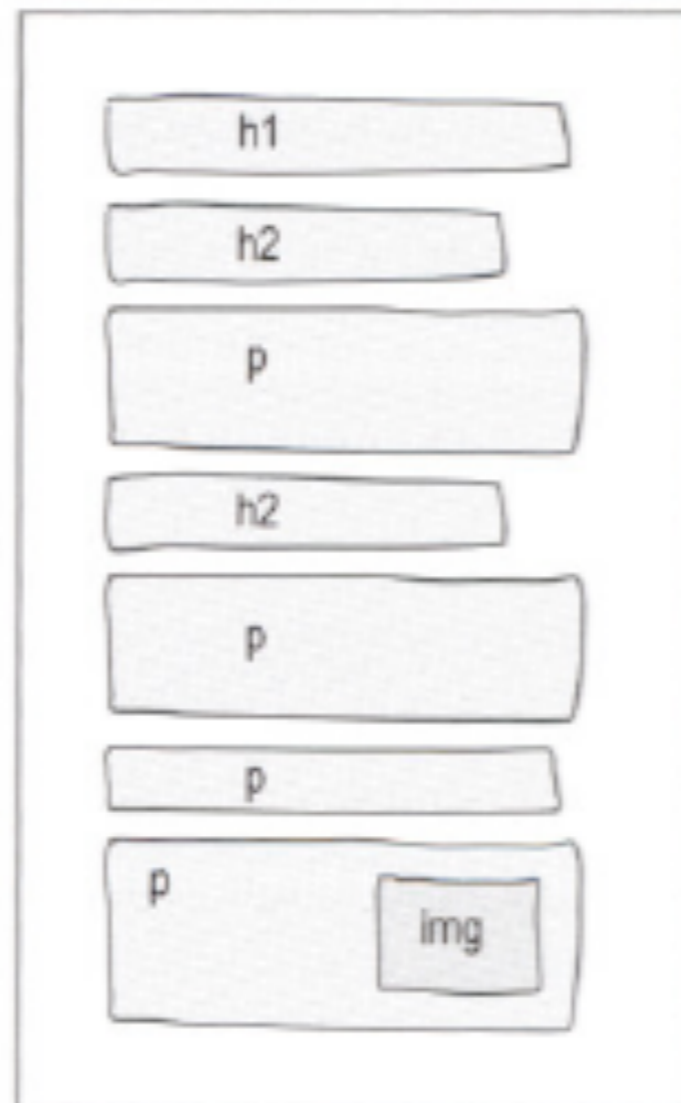




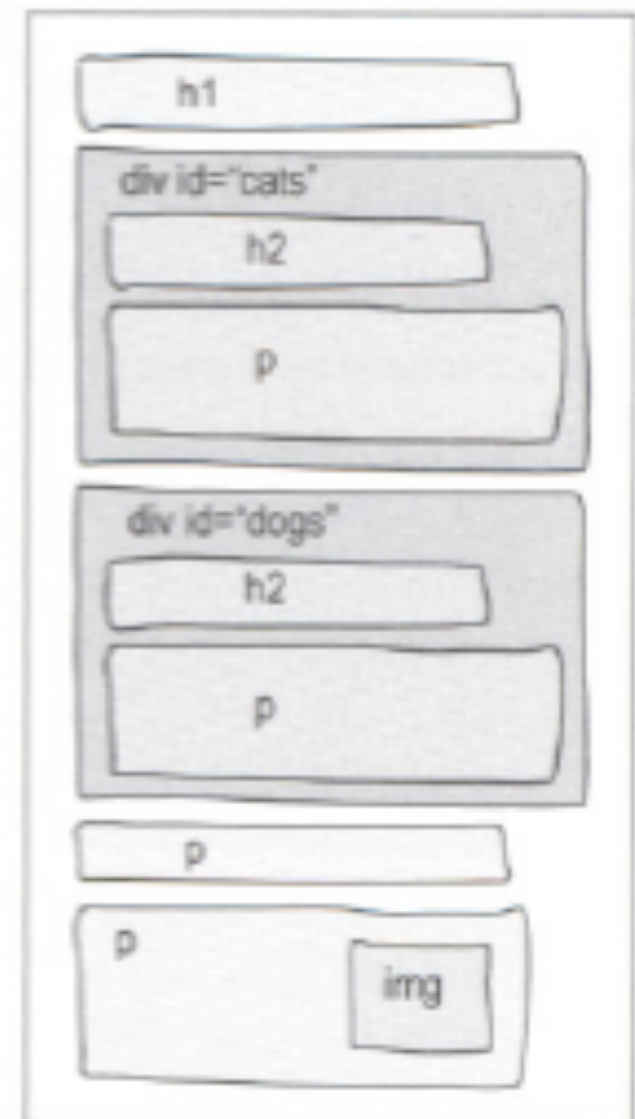
On a referral from the Starbuzz CEO, you've been asked to come in and consult on style changes to PetStorz main page. How quickly would you understand the PetStorz Web page if you were shown Page One?

What about Page Two?

Page One



Page Two



Adding some style

Here we have two rules, one for each <div>. Each <div> is selected by an id selector.

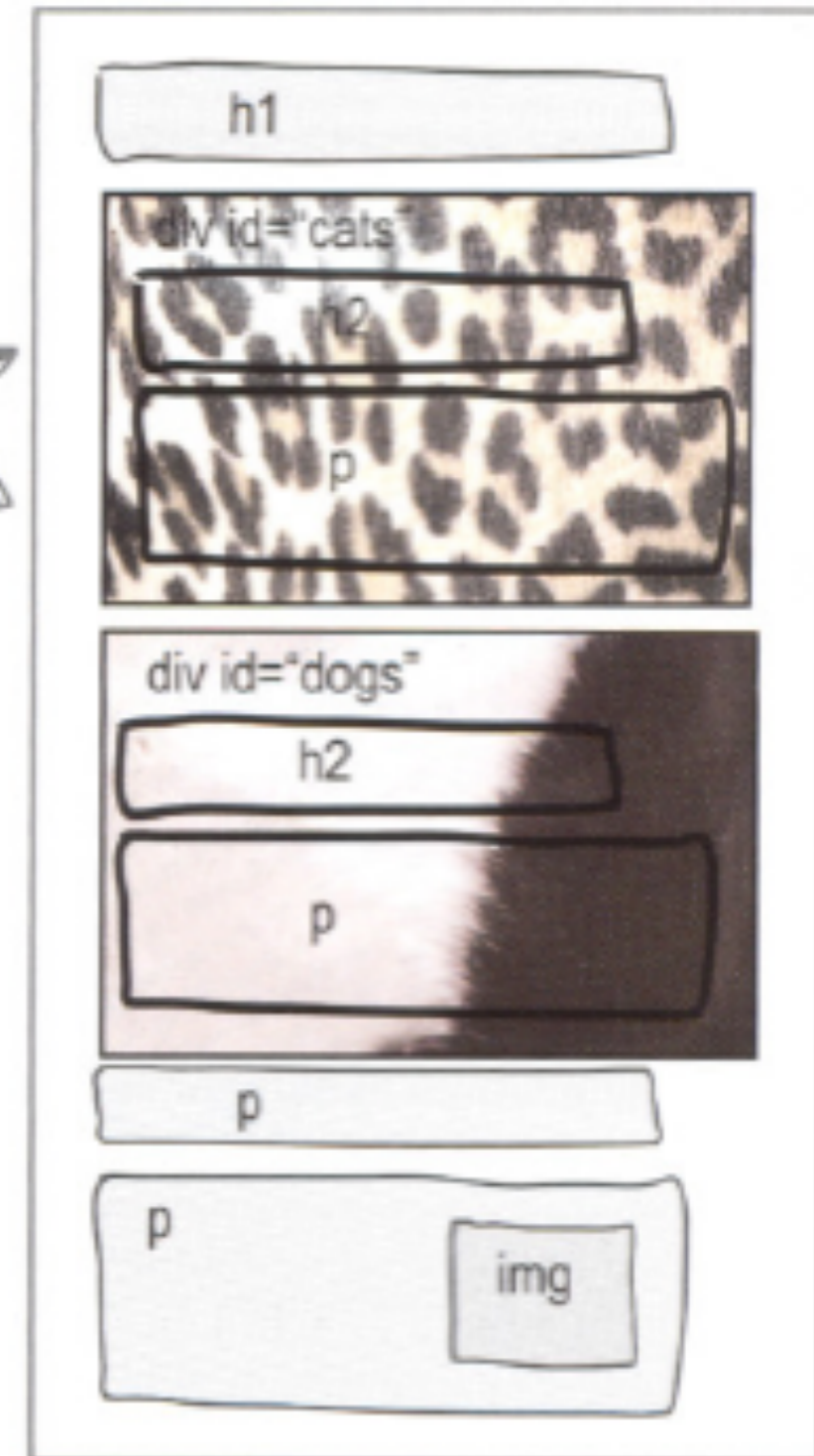
```
#cats {  
  background-image: url(leopard.jpg);  
}  
  
#dogs {  
  background-image: url(mutt.jpg);  
}
```

Each rule sets the background-image property. For cats we have a leopard image, and for dogs we have a mutt image.

Now the <div>s have a little style.

By setting the background on the <div>, it is also shown through the elements contained in the <div>.

The elements in the <div> will also inherit some properties from the <div>, just as any child element does (like font-size, color, etc).

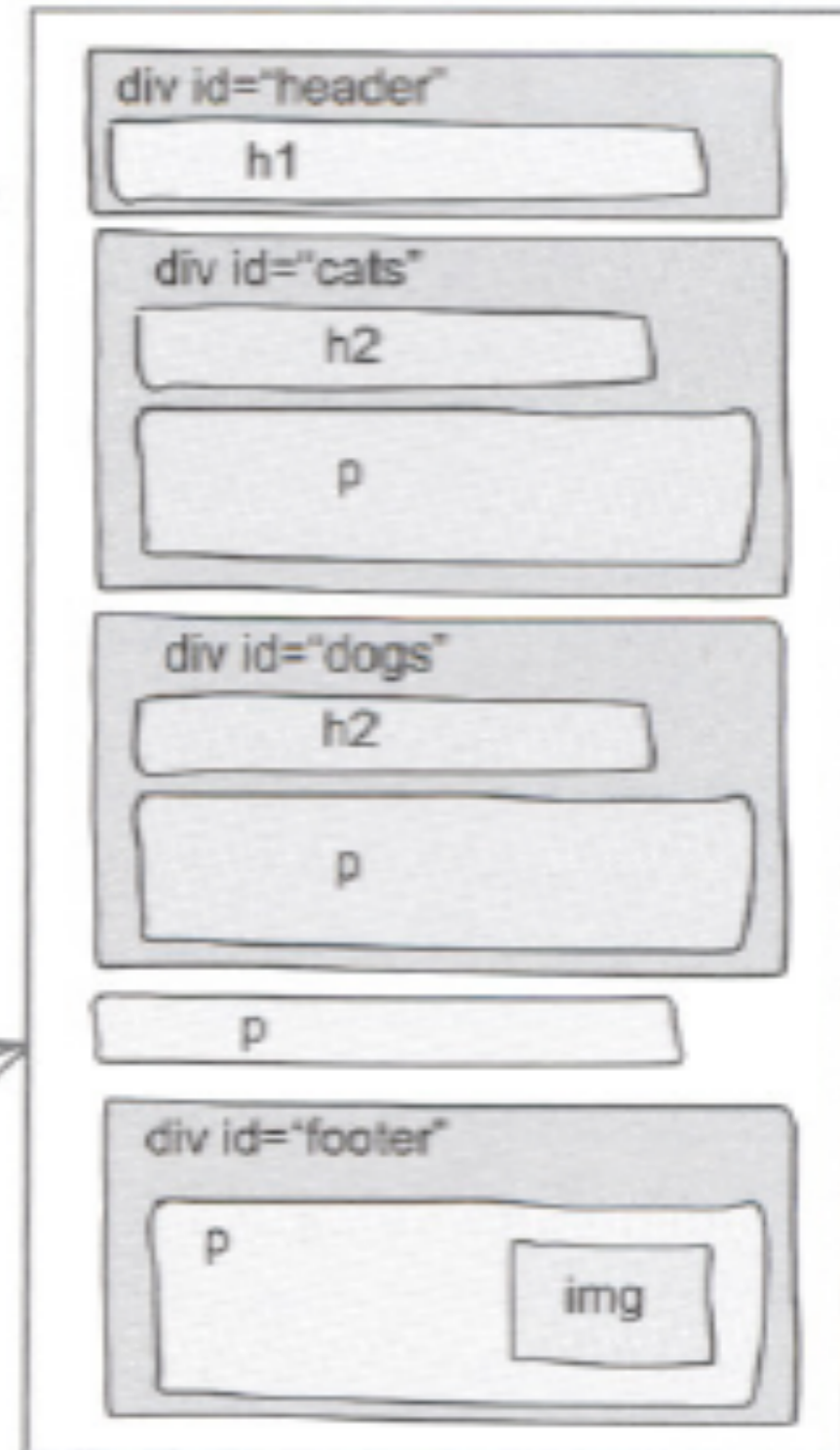


Exposing even more structure

Now we've added another `<div>` with an id indicating this is the header of the page.

And another indicating the footer of the page.

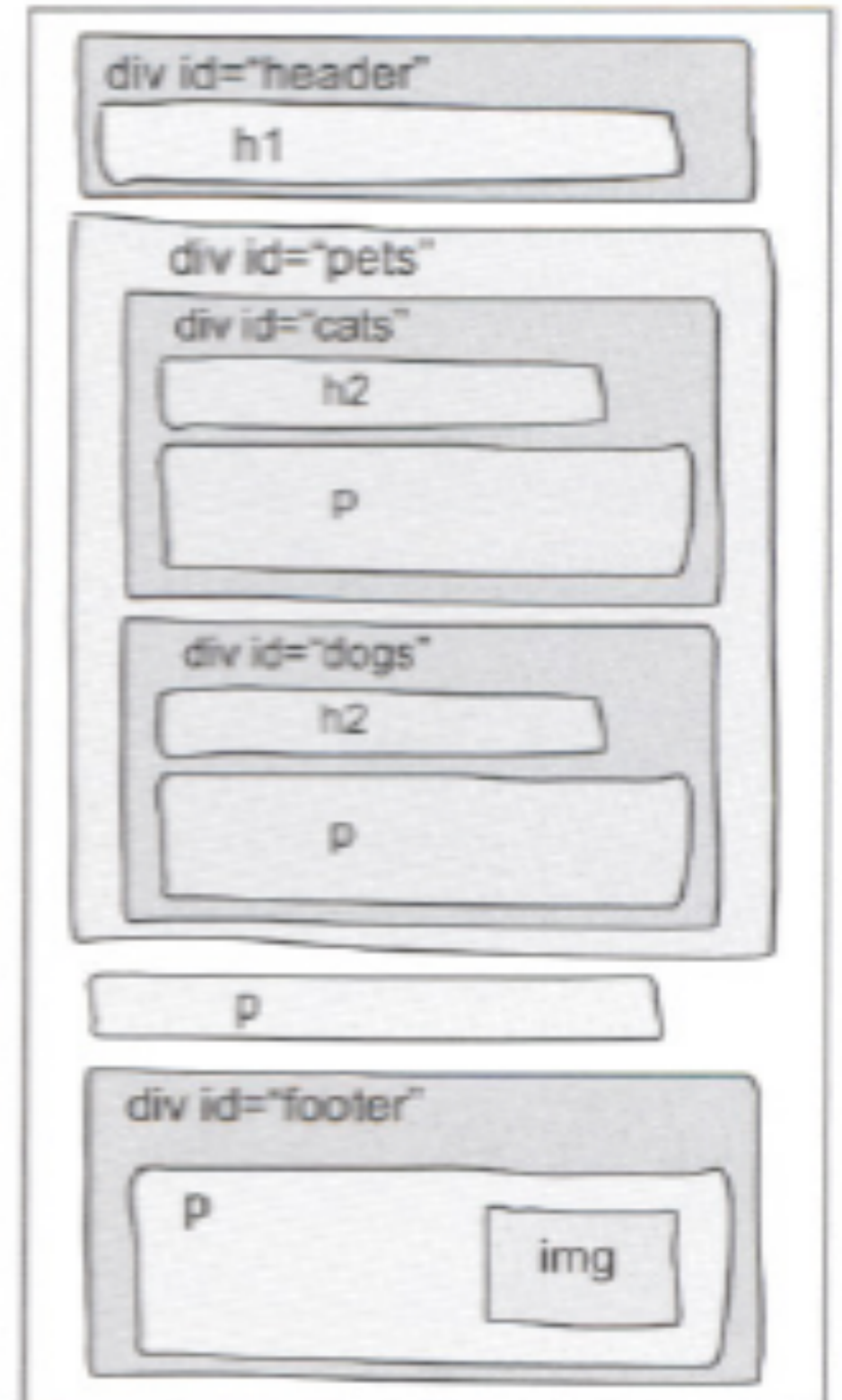
Adding this structure through `<div>`s can even help you think through your page design. For instance, does this lone `<p>` really need to be here?



Adding structure on structure



Now we've marked up this XHTML so that we know there is a logical section in the page with "pets" content in it. Further, that "pets" section has two logical subsections, one for "cats" and one for "dogs".



Div in the Elixir

```
<div id="elixirs">  
  <h2>Weekly Elixir Specials</h2>  
  
  <p>  
      
  </p>  
  <h3>Lemon Breeze</h3>  
  <p>  
    The ultimate healthy drink, this elixir combines  
    herbal botanicals, minerals, and vitamins with  
    a twist of lemon into a smooth citrus wonder  
    that will keep your immune system going all  
    day and all night.  
  </p>  
  
  <p>  
      
  </p>  
  
  <h3>Chai Chiller</h3>  
  <p>  
    Not your traditional chai, this elixir mixes mat&eacute;  
    with chai spices and adds an extra chocolate kick for  
    a caffeinated taste sensation on ice.  
  </p>  
  
  <p>  
      
  </p>  
  
  <h3>Black Brain Brew</h3>  
  <p>  
    Want to boost your memory? Try our Black Brain Brew  
    elixir, made with black oolong tea and just a touch  
    of espresso. Your brain will thank you for the boost.  
  </p>  
  
  <p>  
    Join us any evening for these and all our  
    other wonderful  
    <a href="beverages/elixir.html"  
      title="Head First Lounge Elixirs">elixirs</a>.  
  </p>  
</div>
```

Here's the opening tag, and we've given it an id of "elixirs" to identify it.

Remember we're just showing a snippet of XHTML from the entire file. When you open "lounge.html", you'll see all the markup for the page.

And, here's the closing tag.

Taking the `<div>` for the test drive

Hmmm... no change at all!
But that's okay: the `<div>`
is pure structure, and it
doesn't have any "look" or
default style in the page.

That said, a `<div>` is just a block
element, and you can apply any styles
you want to it. So, once you know how
to style a block element (and you do),
you know how to style a `<div>`.



BRAIN POWER

Remember, the goal here is to restyle the elixir content on the page so it looks like the handout.

Before we took a detour to learn about `<div>`s, we were trying to figure out how to get a border around the entire set of elixirs. Now that you've got a `<div>` in "lounge.html", how would you go about adding a border?



Adding a border

```
#elixirs {  
  border-width: thin;  
  border-style: solid;  
  border-color: #007e7e;  
}
```

↖ Add this at the end of your CSS file. It selects the elixirs <div> element using its id, and adds a thin, solid border in our favorite aquamarine color.

An over-the-border test drive

Here's the border that you just added to the elixirs <div> element.

You added a visible border to this <div>, but it still has no padding and no margin. We'll need to add that too.

Notice that the border goes around all the elements inside the <div> element. The <div> is a box like every other element, so, when you add a border, the border goes around the content, which is all the elements in the <div>.



Adding some real style to elixir section

The width of the elixirs handout is narrower than the rest of the page.

There's a background image at the top.

The main heading and the paragraph text are black, while the drink names are a red color that matches the red in the logo.


The text and images are centered, and there's padding on the sides to add space between the text and the border.

The line-height of the paragraphs looks a lot more like the default line height for the page (before we changed it in the last chapter).

The font family is a sans-serif font, just like the body font, so we don't have to change that. Remember that the `<div>` element and all the elements nested in it inherit the font family from the body.


This link is aquamarine.

Weekly Elixir Specials




Lemon Breeze

The ultimate healthy drink, this elixir combines herbal botanicals, minerals, and vitamins with a twist of lemon into a smooth citrus wonder that will keep your immune system going all day and all night.



Chai Chiller

Not your traditional chai, this elixir mixes maté with chai spices and adds an extra chocolate kick for a caffeinated taste sensation on ice.




Black Brain Brew

Want to boost your memory? Try our Black Brain Brew elixir, made with black oolong tea and just a touch of espresso. Your brain will thank you for the boost.

Join us any evening for these and all our other wonderful [elixirs](#).

Working on Elixir width

```
#elixirs {  
  border-width: thin;  
  border-style: solid;  
  border-color: #007e7e;  
  width: 200px;  
}
```



The width property lets you specify the width of the element's content area. Here we're specifying that the content width be 200 pixels.

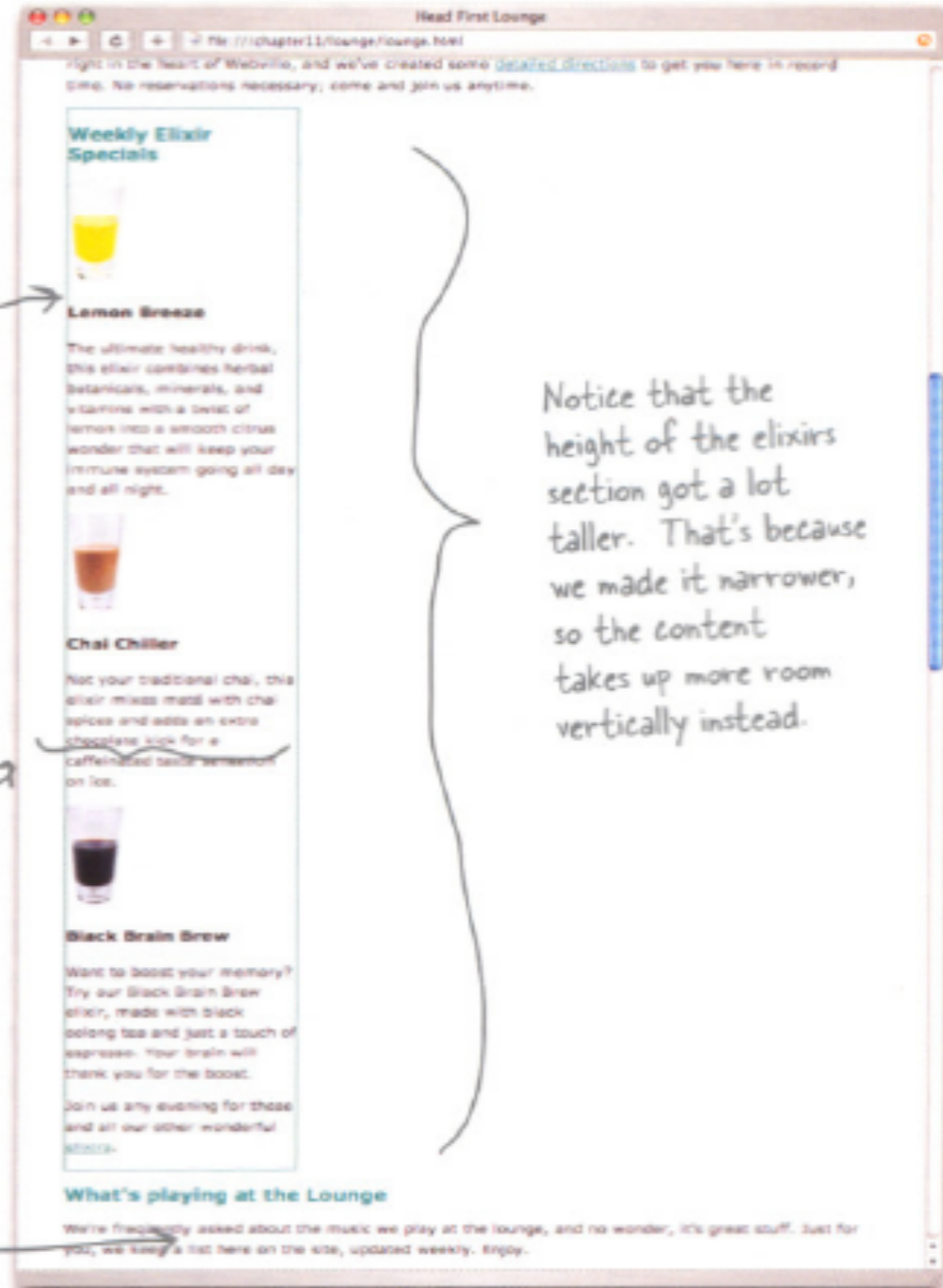
We're setting this on the elixirs <div>. So the content in the elixirs <div> will be 200 pixels wide, and the browser's layout rules will work to fit all the elements nested in the <div> within that width.

Test driving the width

Now all the content in the elixirs <div> fits into a space that is 200 pixels wide. It doesn't change, even if you make your browser window really wide, or really narrow. Try it!

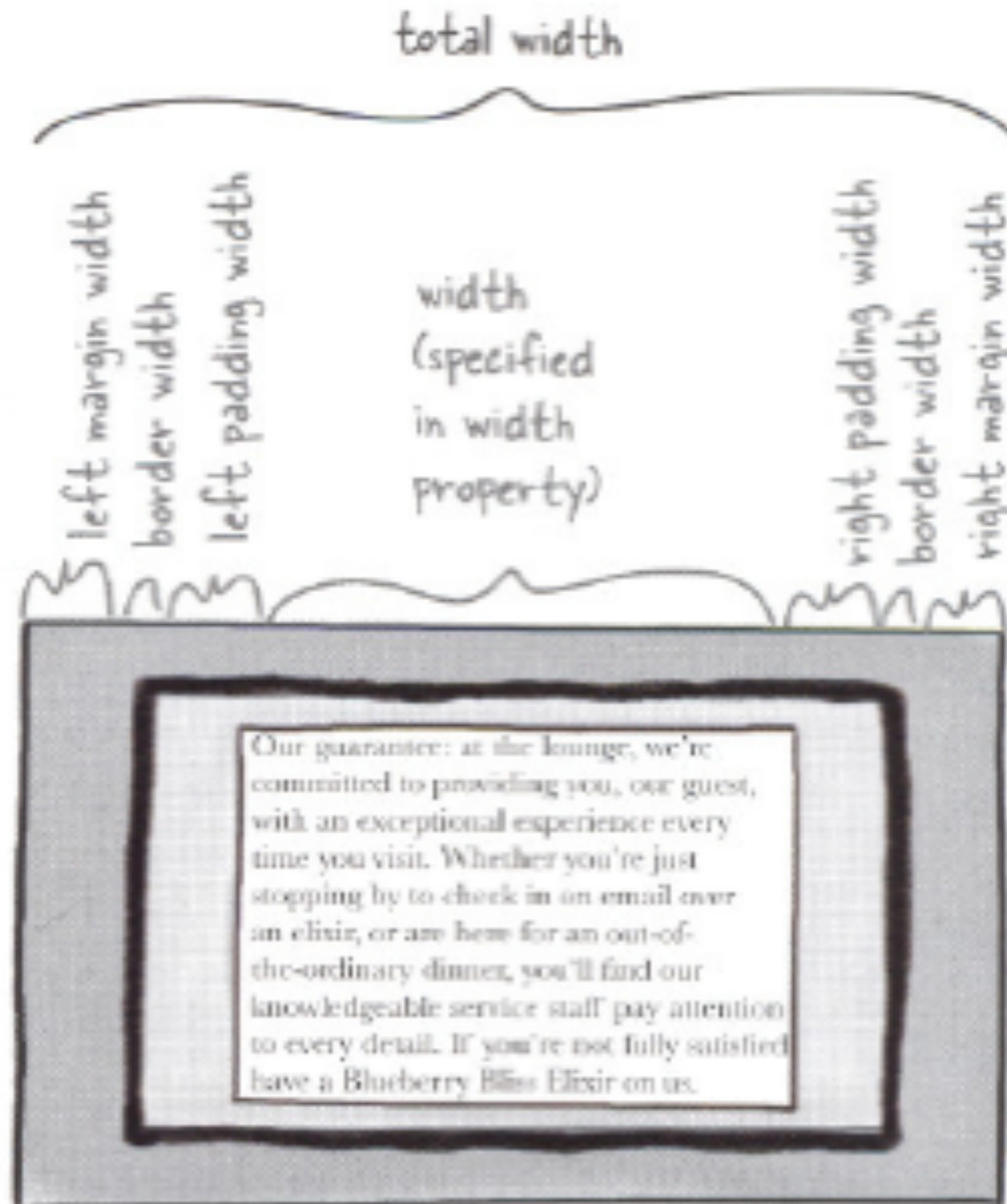
200 pixels

Compare the behavior of the <div> to that of the other elements when you make your browser window wide. The paragraphs automatically expand to fill the width of the browser. We'll talk about that more in a sec...

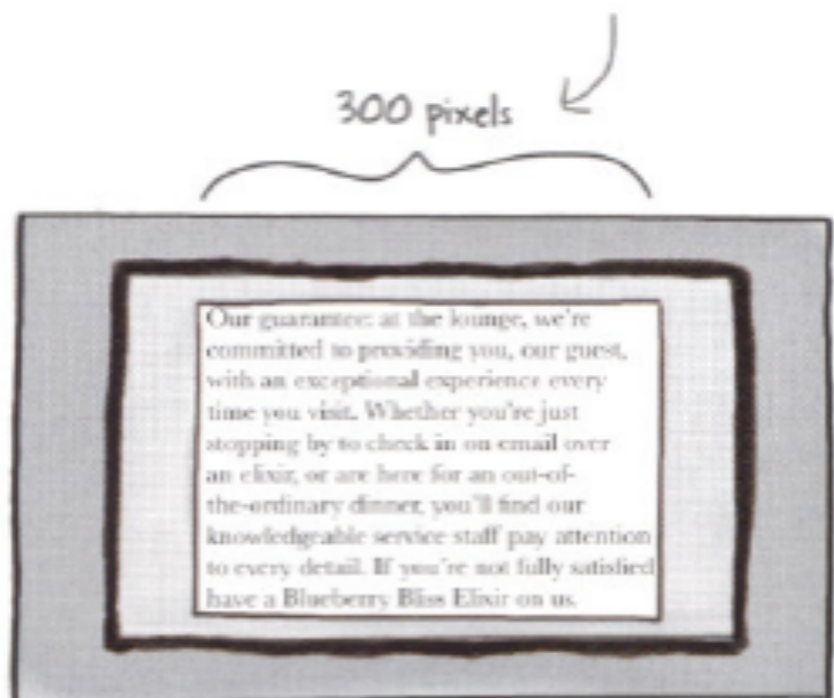


Notice that the height of the elixirs section got a lot taller. That's because we made it narrower, so the content takes up more room vertically instead.

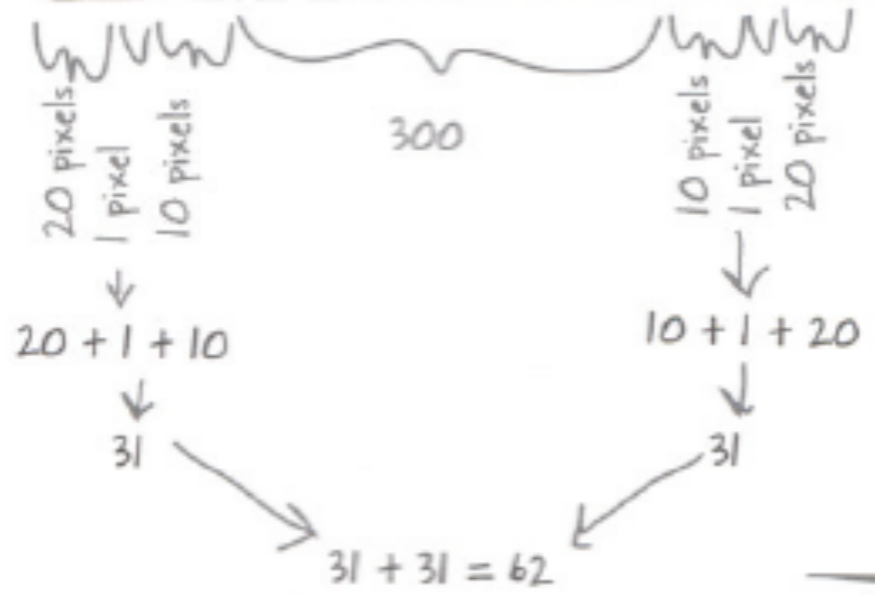
The width property specifies the width for the content area only



(1) The content area is 300 pixels.



(2) Figure out how much is taken up by the margins, padding, and border.



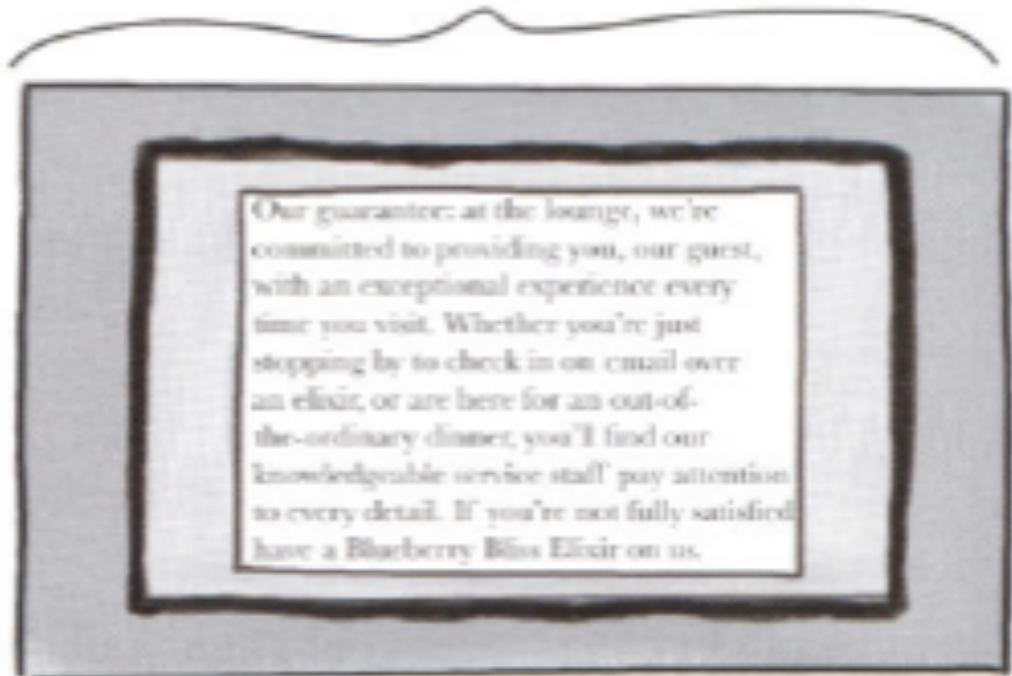
(3) It looks like 62 pixels are taken up, so add that to the content area's width of 300 pixels, and we have $300 + 62 = 362$ pixels for the entire box.

Sharpen your pencil



Here's a box that has all the widths labelled. What is the width of the entire box?

Your answer here



30 pixels
2 pixels
5 pixels

200 pixels

10 pixels
2 pixels
20 pixels

Adding the basic style to elixirs

Remember we're going to apply all this style to the elixirs `<div>` so that it only affects the `<div>` and the elements it contains, not the entire page.

```
#elixirs {  
  border-width:    thin;  
  border-style:   solid;  
  border-color:   #007e7e;  
  width:          200px;
```

```
  padding-right:  20px;  
  padding-bottom: 20px;  
  padding-left:   20px;
```

```
  margin-left:    20px;
```

```
  text-align:     center;
```

```
  background-image: url(images/cocktail.gif);  
  background-repeat: repeat-x;
```

```
}
```

The default padding on a `<div>` is 0 pixels, so we're going to add some padding to provide a bit of space for the content. Notice that we're not adding any padding at the top because there's already plenty of room there thanks to the default margin on the `<h2>` heading (look back at the last test drive and you'll see there's plenty of room above the `<h2>`). But we do need it on the right, bottom, and left.

We're adding some margin on the left to indent the elixirs from the rest of the page a bit. This is going to come in handy later...

Use `text-align` on block elements to align the text they contain. Here we're going to center-align the text.

And finally we're specifying an image to use in the background, in this case the cocktail image. We're setting the `background-repeat` property to `repeat-x`, which will tile the image only in the horizontal direction.

Test driving the new style

Head First Lounge
file:///chapter11/lounge/lounge.html

created some [detailed directions](#) to get you here in record time. No reservations necessary; come and join us anytime.

Weekly Elixir Specials

Lemon Breeze
The ultimate healthy drink, this elixir combines herbal botanicals, minerals, and vitamins with a twist of lemon into a smooth citrus wonder that will keep your immune system going all day and all night.

Chai Chiller
Not your traditional chai, this elixir mixes maté with chai spices and adds an extra chocolate kick for a caffeinated taste sensation on ice.

Black Brain Brew
Want to boost your memory? Try our Black Brain Brew elixir, made with black oolong tea and just a touch of espresso. Your brain will thank you for the boost.
Join us any evening for these and all our other wonderful [elixirs](#).

What's playing at the Lounge
We're frequently asked about the music we play at the lounge, and no wonder, it's great stuff. Just for you, we keep a list here on the site, updated weekly. Enjoy.

Handwritten annotations:

- ← The tiled image looks nice, and it only tiles horizontally.
- ← We've got some padding here, and at the bottom and left...
- ← ... and everything's centered nicely.

That's interesting because I noticed the text inside the `<div>` is all inside other block elements, like `<h2>`, `<h3>`, and `<p>`. So, if `text-align` is aligning *inline* elements in the `<div>` block element, how is the text in these nested block elements getting aligned?



Sharpen your pencil



So now that you understand widths, what's the total width of the elixirs box? To start with, we know the content area is 200 pixels. We've also set some left and right padding that affects the width, as well as a border that's set to "thin". Just assume a thin border is 1 pixel thick, like it is on most browsers. And what about margins? We set a left margin value, but no right margin value, so the right margin is 0 pixels by default.

Here are all the properties that relate to width. Your job is to figure out the total width of the elixirs <div>.

```
border-width: thin;
```

```
width: 200px;
```

```
padding-right: 20px;
```

```
padding-bottom: 20px;
```

```
padding-left: 20px;
```

```
margin-left: 20px;
```



Weekly Elixir Specials



Lemon Breeze

The ultimate healthy drink, this elixir combines herbal botanicals, minerals, and vitamins with a twist of lemon into a smooth citrus wonder that will keep your immune system going all day and all night.



Chai Chiller

Not your traditional chai, this elixir mixes masala with chai spices and adds an extra chocolate kick for a caffeinated taste sensation on ice.

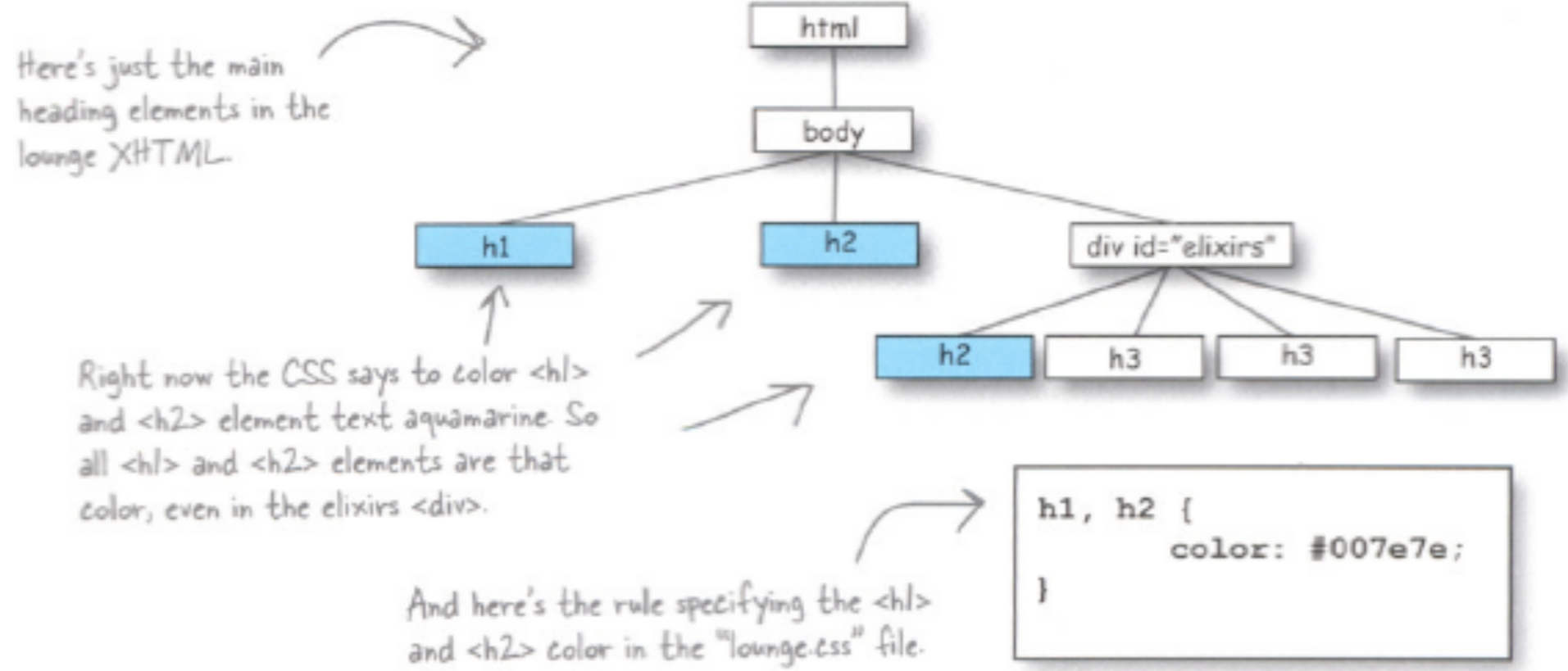


Black Brain Brew

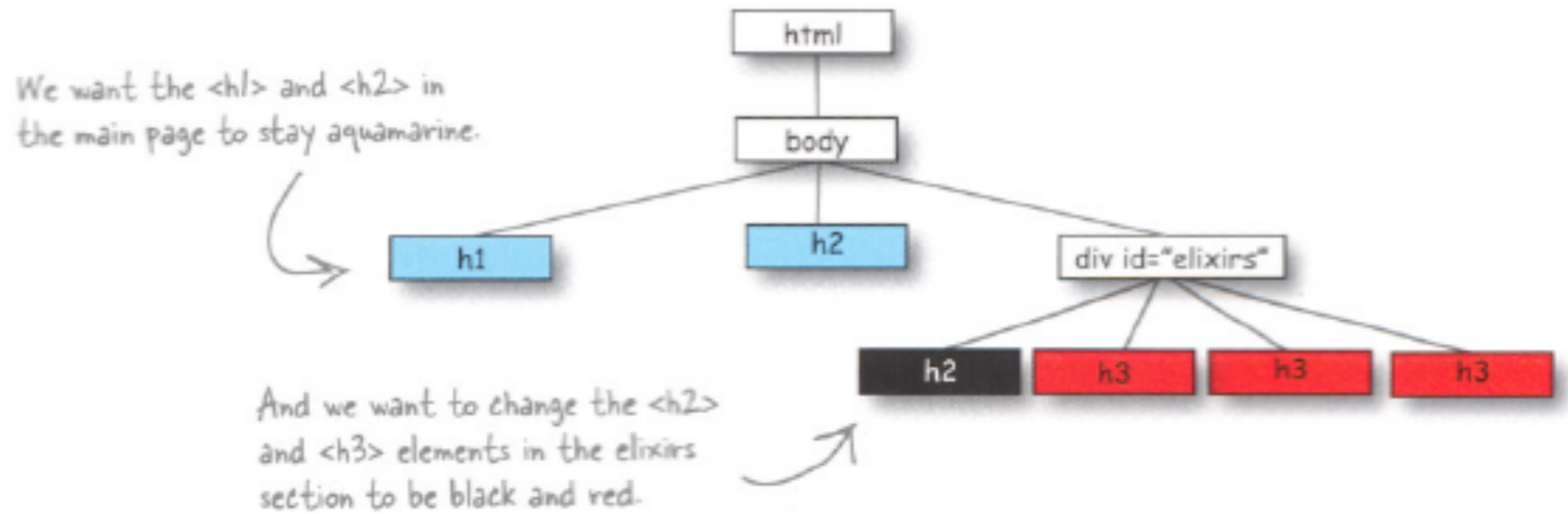
Want to boost your memory? Try our Black Brain Brew elixir, made with black oolong tea and just a touch of espresso. Your brain will thank you for the boost.

Join us any evening for these and all our wonderful elixirs.

What we have now



What we want

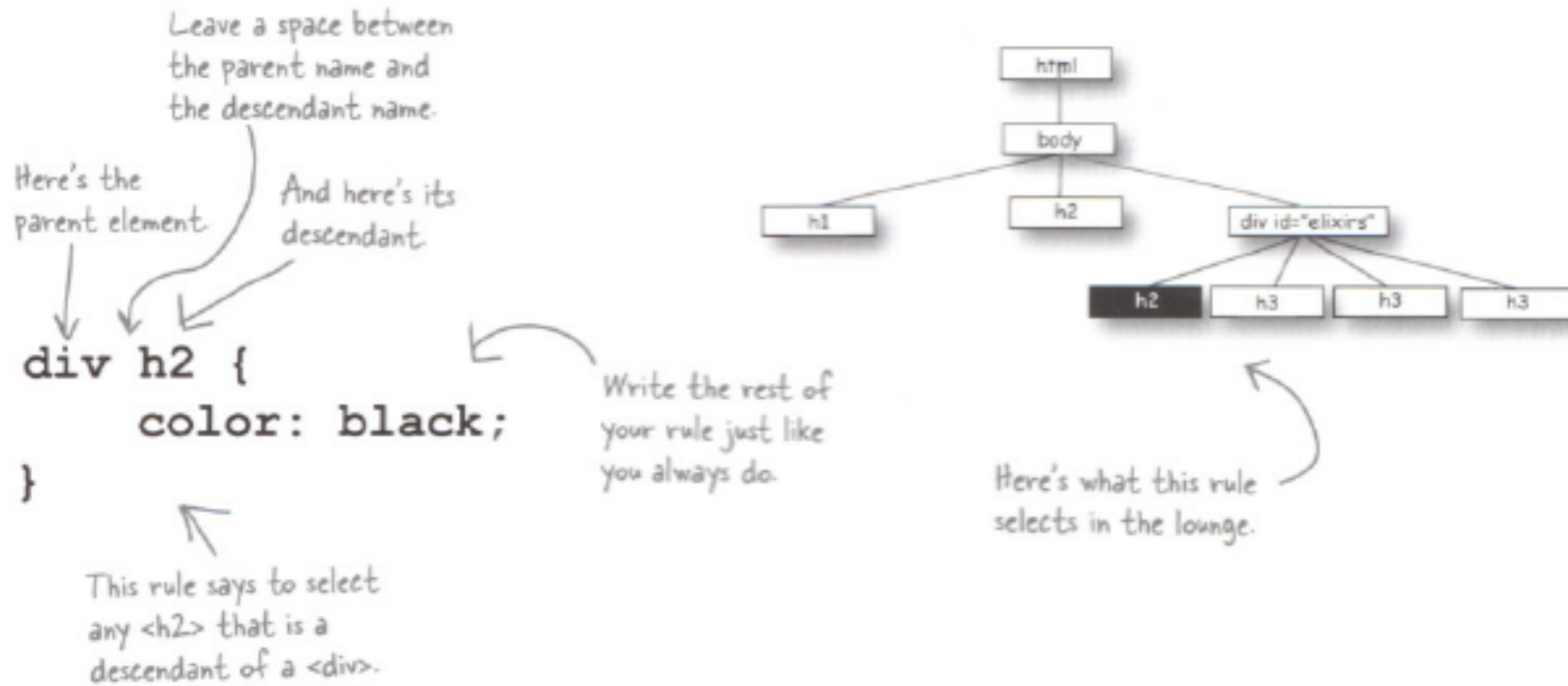


But if we change the existing rule for `<h2>`, we'll affect the font color of every `<h2>` in the main page. And, if we add a new rule for `<h3>`, then any `<h3>`s that get added to the main page later will be red, which is not what we want. Now, we could use a class like Jim suggested, but we're going to give Frank's idea a try first.

```
h1, h2 {  
    color: #007e7e;  
}
```

?

What we need is the way to select descendants



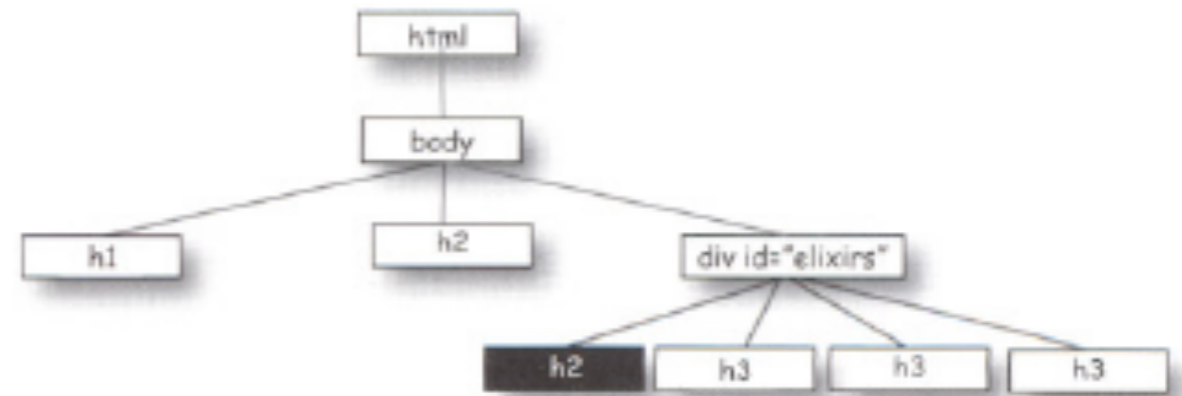
What we need is the way to select descendants

Now the parent element is the element with the id elixirs.

And here's its descendant.

```
#elixirs h2 {  
  color: black;  
}
```

This rule says to select any `<h2>` that is a descendant of an element with the id "elixirs".

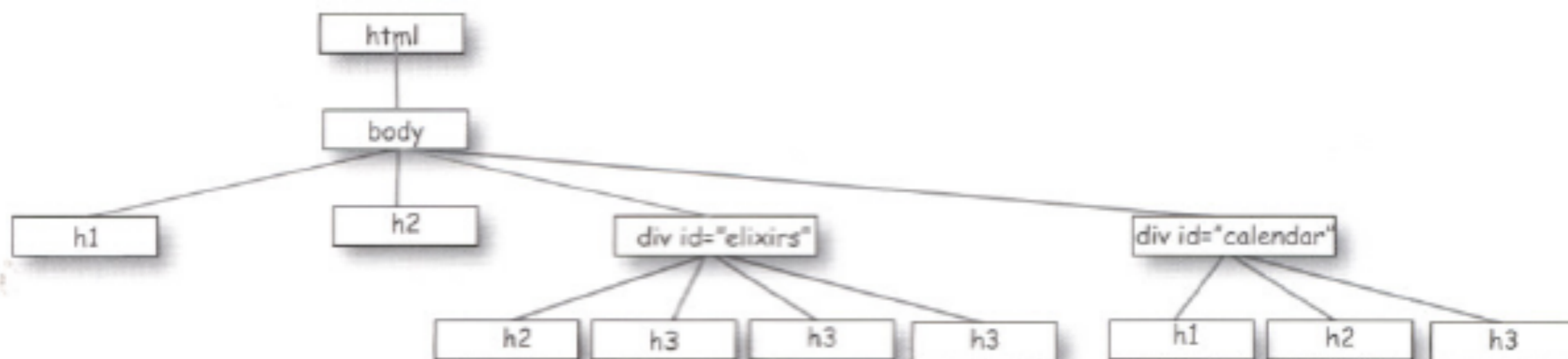


This rule selects the same element. But it's more specific, so if we added another `<div>` with an `<h2>` to the page, that's okay because this rule selects only `<h2>`s in the elixirs `<div>`.

Sharpen your pencil



Your turn. Write the selector that selects only `<h3>` elements inside the elixirs `<div>`. In your rule, set the color property to `#d12c47`. Also label the elements in the graph below that are selected.



Changing the color of elixirs headers

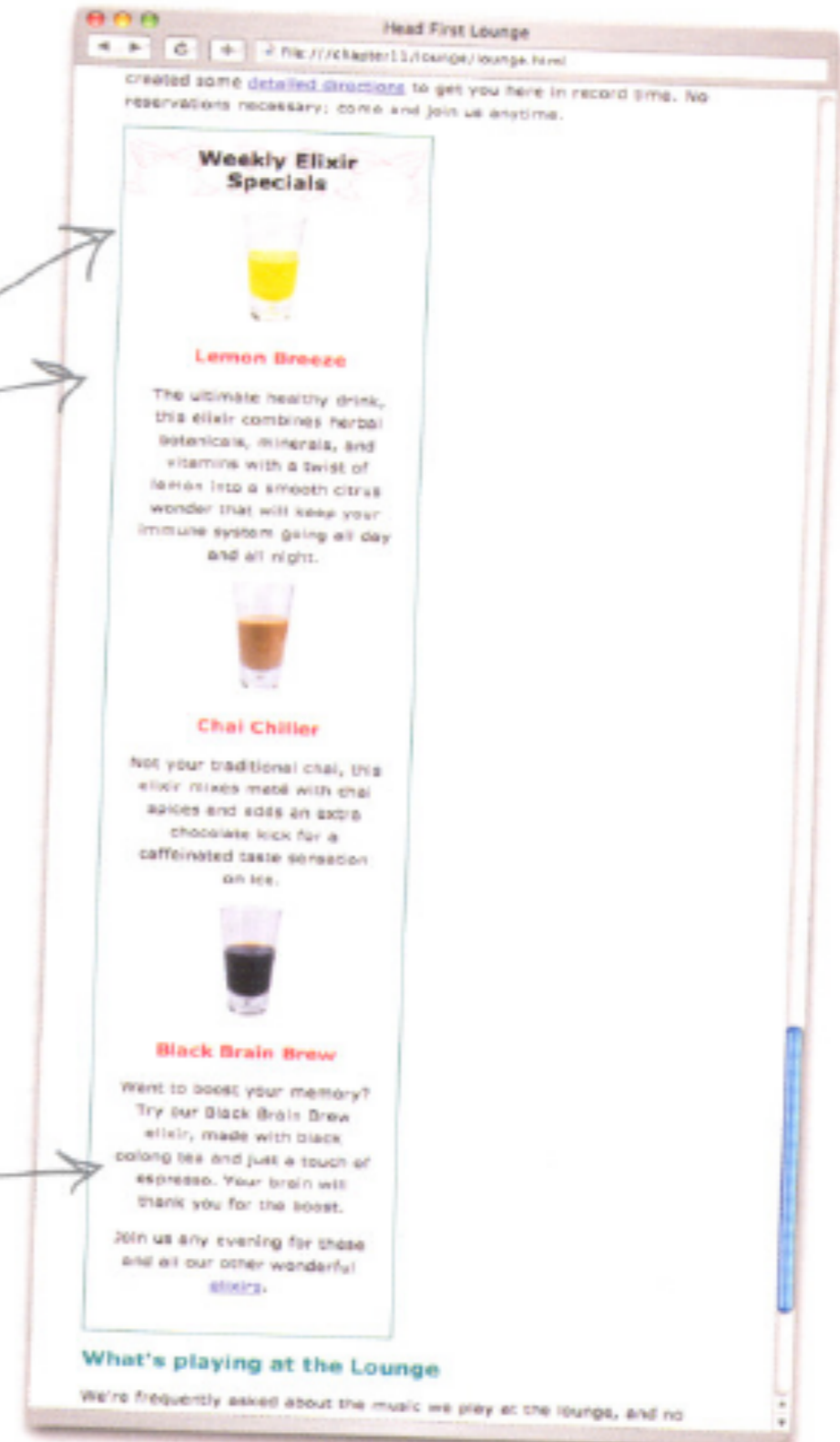
```
#elixirs h2 {  
  color: black;  
}  
  
#elixirs h3 {  
  color: #d12c47;  
}
```

← Here we're using the descendant selectors to target just the `<h2>` and `<h3>` elements in the elixirs `<div>`. We're setting `<h2>` to black, and `<h3>` to a red color, using a hex code.

A quick test drive

We've got black and red headings in the elixirs section, and we haven't affected the aquamarine color being used for <h2> headings in the main page.

Now all we need to do is fix the line height.



Fixing line height

```
#elixirs {  
  line-height: 1em;  
}
```

If you set the line-height property on the entire <div> then it will be inherited by all elements in the <div>, including the headings. Notice that the line height in the heading is too small and the two lines are starting to run together.



Here are the font sizes of the elements. We set body to "small", so that's inherited by elixirs.

body size is "small"

div id="elixirs" size is "small"

h2 is 120% of "small"

The line-height of <h2> is set to 1 times the font size of elixirs, which is "small", or about 12 pixels.

body line-height is 1.6 times "small"

div id="elixirs" line-height is 1 times "small", or about 12 pixels

h2 is 120% of "small" line-height is 1 times 120% of small, or about 14 pixels

We want <h2> to have a line-height that is 1 times its own font size, that is, 14 pixels (120% of small).

```
#elixirs {  
  line-height: 1;  
}
```

Add a line-height of 1 to the elixirs <div> to change the line-height of each element in it.

The font-size of the p element is "small" (p inherits its font-size from the elixirs <div>) so it will have a line-height of 12 pixels, which is what we want.

It's time to take little shortcut

```
padding-top: 0px;  
padding-right: 20px;  
padding-bottom: 30px;  
padding-left: 10px;
```

That's a lot of typing just to specify four numbers.

Here's the old school way of specifying your padding.

```
padding-top: 0px;  
padding-right: 20px;  
padding-bottom: 30px;  
padding-left: 10px;
```

And here's the new and improved way to write them as a shorthand.

```
padding: 0px 20px 30px 10px;
```

top right bottom left

```
margin-top: 0px;  
margin-right: 20px;  
margin-bottom: 30px;  
margin-left: 10px;
```

```
margin: 0px 20px 30px 10px;
```

top right bottom left

Just like padding, you can use a shorthand to specify all your margin values with one property.

```
padding-top: 20px;  
padding-right: 20px;  
padding-bottom: 20px;  
padding-left: 20px;
```

```
padding: 20px;
```

If all your padding values are the same, then you can write it like this.

This says that the padding should be 20 pixels on every side of the box.

Shortcut...

```
margin-top: 0px;
margin-right: 20px;
margin-bottom: 0px;
margin-left: 20px;
```

top and bottom are the same.

right and left are the same.

If the top and bottom, as well as the right and left margins are the same, then you can use a shorthand.

```
margin: 0px 20px;
```

top and bottom

right and left

```
border-width: thin;
border-style: solid;
border-color: #007e7e;
```

Rewrite border properties as one property. These can be in any order you like.

```
border: thin solid #007e7e;
```

```
border: solid thin #007e7e;
```

```
border: #007e7e solid thin;
```

```
border: solid thin;
```

```
border: #007e7e solid;
```

```
border: solid;
```

These are all perfectly valid border shorthands.

```
background-color: white;
background-image: url(images/cocktail.gif);
background-repeat: repeat-x;
```

Like border, values can go in any order in this shorthand. There are also a few other values you can specify in the shorthand, like background-position.

```
background: white url(images/cocktail.gif) repeat-x;
```

Shortcut...

Here are the properties that go into the font shorthand. Ordering matters here unless we say otherwise...

Finally you need to add your font families. You only need to specify one font, but alternatives are highly encouraged.

You must specify font size.

`font: font-style font-variant font-weight font-size/line-height font-family`

These values are all optional. You can specify any combination of them, but they need to come before font-size.

The line-height is optional. If you want to specify one, just put a / right after the font-size and add your line height.

Use commas between your font family names.

```
font-size: small;
font-family: Verdana, Helvetica, Arial, sans-serif;
line-height: 1.6em;
```

We're not using any of these, but that's okay, they're all optional.

`font: font-style font-variant font-weight font-size/line-height font-family`

```
font: small/1.6em Verdana, Helvetica, Arial, sans-serif;
```

And here's the shorthand version. Wow, that's quite a shorthand, huh? You're going to be able to double your time at the slopes (or on the beach) now.

Make It Stick

HTML is
for
structure

CSS is
for
presentation

Wash
Cat



To remember the ordering of the padding and margin shorthand values, think of a clock labeled with top, right, bottom, and left. Then, always go in a clockwise direction: top to right to bottom to left.

```
margin: 0px 20px 30px 10px;
```

↑ ↑ ↖ ↖
top right bottom left



I saw the nice job you did on the elixirs. Can you give us a hand with the music recommendations on the site? We don't need much, just some simple styling.

What's playing at the Lounge

We're frequently asked about the music we play at the lounge, and no wonder, it's great stuff. Just for you, we keep a list here on the site, updated weekly. Enjoy.

- *Buddha Bar*, **Claude Challe**
- *When It Falls*, **Zero 7**
- *Earth 7*, **L.T.J. Bukem**
- *Le Roi Est Mort, Vive Le Roi!*, **Enigma**
- *Music for Airports*, **Brian Eno**

The lounge's resident DJ.

All the CD titles are in an italic font style.

And all the artists are in bold.

 **BRAIN
POWER**

What do you think is the best way to style the CD and artists in the
"What's playing at the Lounge" section?

Adding ``s in three easy way

- 1 You're going to nest the CDs and artists in separate `` elements.
- 2 You're going to add one `` to the "cd" class and the other to the "artist" class.
- 3 You're going to create a rule to style the "cd" class with italic, and the "artist" class with bold.

Step one and two: add the s

Each list item consists of a CD title, a comma, and then the music artist.

```
<ul>
<li>Buddha Bar, Claude Challe</li>
<li>When It Falls, Zero 7</li>
<li>Earth 7, L.T.J. Bukem</li>
<li>Le Roi Est Mort, Vive Le Roi!, Enigma</li>
<li>Music for Airports, Brian Eno</li>
</ul>
```

Just add a opening tag along with the class attribute and a value of "cd".

Next, add a closing tag after the CD title.

Do the same for the artist. Nest it in a element, only this time put the in the "artist" class.

```
<ul>
<li><span class="cd">Buddha Bar</span>, <span class="artist">Claude Challe</span></li>
<li>When It Falls, Zero 7</li>
<li>Earth 7, L.T.J. Bukem</li>
<li>Le Roi Est Mort, Vive Le Roi!, Enigma</li>
<li>Music for Airports, Brian Eno</li>
</ul>
```

Step three: style the s

We're going to add a rule for each of the new classes, cd and artist.

```
.cd {  
  font-style: italic;  
}
```

For CDs we'll make the font style italic.

```
.artist {  
  font-weight: bold;  
}
```

And for artists we'll set the font-weight to bold.



Now the first music recommendation has the correct styling.



Sharpen your pencil

You need to finish the job. Add `` elements to the rest of the music recommendations and test your page. You'll find the solution in the back of the chapter.

```
<ul>
<li><span class="cd">Buddha Bar</span>, <span class="artist">Claude Challe</span></li>
<li>When It Falls, Zero 7</li>
<li>Earth 7, L.T.J. Bukem</li>
<li>Le Roi Est Mort, Vive Le Roi!, Enigma</li>
<li>Music for Airports, Brian Eno</li>
</ul>
```

<a> and style

Here's a link you've never clicked on before. This is called an "unvisited link", or just "link", and it's blue by default.



Join us any evening for these and all our other wonderful [elixirs](#).



Join us any evening for these and all our other wonderful [elixirs](#).

And here's a link you have clicked on before. We call these "visited links". Usually, visited links are displayed in a different color than unvisited links so that you can tell the difference. In most browsers, visited links are purple by default.

Join us any evening for these and all our other wonderful [elixirs](#).
Head First Lounge Elixirs



And if you hold your mouse over a link without clicking, this is called "hovering". On some browsers you'll see a tool tip that displays the text of the "title" attribute. And, if you pay close attention, on some Web pages, you'll see a different style as you hover.



<a> style

Notice we have the element <a>, followed by a ":" followed by the state we want to select.

↓
`a:link {
 color: green;
}`

This selector is applied to links when they are in an unvisited state.

`a:visited {
 color: red;
}`

← And this selector is applied to links when they are visited.

`a:hover {
 color: yellow;
}`

← And this selector is applied when you hover over a link.

Putting pseudo-classes to work

Okay, big change here. We're using a descendant selector combined with a pseudo-class. The first selector says to select any unvisited `<a>` element that is nested in an element with the id "elixirs". So we're JUST styling the links inside elixirs.

```
#elixirs a:link {  
  color: #007e7e;  
}
```

On these two we're setting the color. For unvisited links, a nice aquamarine...

```
#elixirs a:visited {  
  color: #333333;  
}
```

...and for visited links we're using a dark gray.

```
#elixirs a:hover {  
  background: #f88396;  
  color: #0d5353;  
}
```

Now for the really interesting rule. When the user is hovering over the link, we're changing the background to red. This makes the link look highlighted when you pass the mouse over it. Give it try!



Exercise

Open up your "lounge.css" and rework your `a:link`, `a:visited`, and `a:hover` rules to use the new descendant selector and the new style definitions. Save, reload, and turn the page.

Cascade style sheet



**The Author
(that's you!)**

First, there are all the style sheets you've written for your page.

Note that there is a way for a reader to actually override your styles. To do that they put "!important" at the end of a property declaration.



**The Reader
(your users)**

But some browsers also allow users to create their own styles for XHTML elements. If your style sheet doesn't define these styles, the user's style sheet is used instead.

And finally, you already know that the browser itself maintains a set of default styles that are used if you don't define the styles for an element. These are also the styles that are used if you don't have any author or reader style sheets.

When the browser needs to determine which style to apply to an element, it uses all these style sheets. Priority is given first to the author's styles (that is, your styles), then to the reader's styles, and then finally to the browser's default styles.



The Browser

So, to review, as the page authors, we can use multiple style sheets with our XHTML. And, the user might also supply their own styles, and then the browser has its default styles, too. And on top of all that we might have multiple selectors that apply to the same element. How do we figure out which styles an element gets?



The cascade

Step one:

Gather all your style sheets together.

Step two:

Find all the declarations that match.

Step three:

Now take all your matches, and sort them.

Step four:

Now sort all the declarations by how specific they are.

Step five:

Finally, sort any conflicting rules in the order they appear in their individual style sheets.



What's my specificity game

To calculate the specificity you start with a set of three numbers, like this:

0 0 0

In the old days we used four numbers, but that was before XHTML... aren't you glad you're learning this now?

And then we just tally up various things from the selector, like this:

Does the selector have any ids? One point each.

Does the selector have any classes or pseudo-classes? One point each.

Does the selector have any element names? One point for each.

0 0 0

For instance, the selector "h1" has one element in it, so you get:

Read this as the number one. → **0 0 1**

Both "h1" and "h1.blue" have one element, so they both get a "1" in the right most number column.

As another example, the selector "h1.blue" has one element and one class, so you'd get:

Read this as the number eleven. → **0 1 1**

"h1.blue" also has one class, so it gets a "1" in the middle number column.

Neither have ids in their selectors, so they both get a "0" in the left number column.

Step one

```
<h1 class="blueberry">Blueberry Bliss Elixir</h1>
```

```
h1 {  
  color: black;  
}
```



The Browser

Remember, you're the browser, because you're trying to figure out how display this <h1> element.

Let's take this through all the cascade steps:

Step one:

Gather all your style sheets together.

```
h1 {  
  color: #efefef;  
}  
  
h1.blueberry {  
  color: blue;  
}
```



The Author

```
body h1 {  
  color: #cccccc;  
}
```



The Reader

The person using the browser.

That's you (for now).

Usually, you're the author (the person writing the CSS). But right now, you're the browser.

Step two

Step two:

Find all the declarations that match.

Here are all the rules that could possibly match the `<h1>` element and that contain the `color` property.



Reader {
body h1 {
color: #cccccc;
}

Browser {
h1 {
color: black;
}

Author {
h1 {
color: #efefef;
}
h1.blueberry {
color: blue;
}

Step three

Step three:

Now take all your matches, and sort them by author, reader, browser.

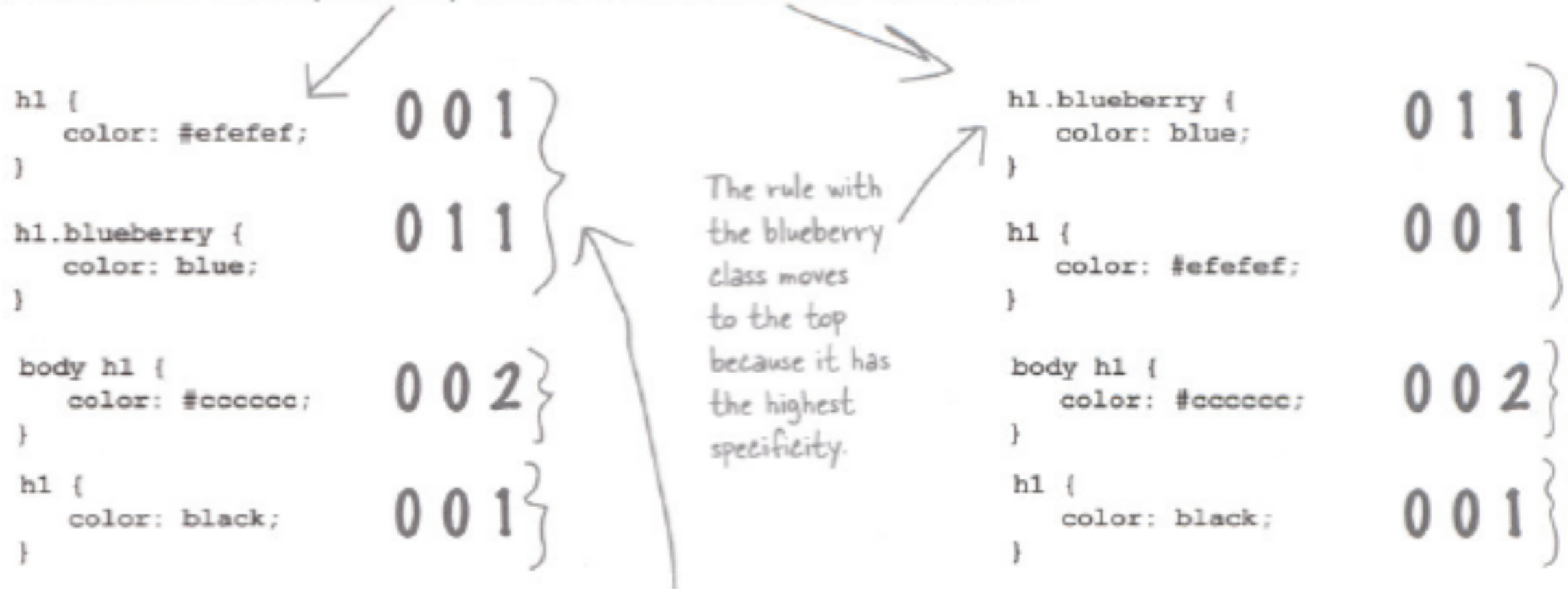
```
Author {  
  h1 {  
    color: #efefef;  
  }  
  h1.blueberry {  
    color: blue;  
  }  
Reader {  
  body h1 {  
    color: #cccccc;  
  }  
Browser {  
  h1 {  
    color: black;  
  }  
}
```

← Here we've just reordered the rules by author, then reader, and then browser.

Step four

Step four:

Now sort the declarations by how specific they are. To do that we need to first calculate each specificity score, and then reorder the rules.



Notice that we only sort within the author, reader, and browser categories. We don't re-sort the entire list, or else the "body h1" rule would move above the "h1" rule set by the author.

Step five

Step five:

Finally, sort any conflicting rules in the order that they appear in their individual style sheets.

We're okay here, because we don't have any conflicting rules at this point. The blueberry, with a score of 11, is the clear winner. If there had been two rules with a score of 011, then the rule appearing latest would be the winner.



```
h1.blueberry {  
  color: blue;  
}  
  
h1 {  
  color: #efefef;  
}  
  
body h1 {  
  color: #cccccc;  
}  
  
h1 {  
  color: black;  
}
```

} Author
}
} Reader
} Browser

Homework



This is a special brain power; so special that we're going to let you think about it between chapters. Here's what you need to do:

- 1 Open the file "lounge.html" and locate the elixirs `<div>`.
- 2 Move the entire elixirs `<div>` section to top the of the file so it's just below the paragraph that contains the lounge logo.
- 3 Save and reload your page. What changed?
- 4 Open the file "lounge.css".
- 5 Locate the "#elixirs" rule.
- 6 Add this declaration at the bottom of the rule:

```
float: right;
```

- 7 Save your file, and reload the page in your browser.

What changed? What do you think this declaration does?