



CORSO DI LAUREA IN INFORMATICA

Tecnologie Software per il Web

HTML 5 & CSS – Part 2

a.a. 2020-2021

Horizontal rule

- Il tag **<hr>** serve ad inserire una riga di separazione
- Attributi:
 - align = {left | center | right} Allineamento della riga rispetto a ciò che la circonda
 - size = pixels Altezza della riga
 - width = length Larghezza della riga in modo assoluto o in % delle dimensioni di ciò che la contiene
 - noshade Riga senza effetto di ombreggiatura

```
<hr width="50%" align="center">
<hr size="5" width="50%" align="center">
<hr noshade size="5" width="50%" align="center">
```

- Si possono applicare le regole CSS:



```
<html>
<head>
<style>
/* Dashed red border */
hr.new1 {
    border-top: 1px dashed red;
}

/* Dotted red border */
hr.new2 {
    border-top: 1px dotted red;
}

/* Large rounded green border */
hr.new3 {
    border: 10px solid green;
    border-radius: 5px;
}
</style>
</head>
<body>
<hr>
<hr class="new1">
<hr class="new2">
<hr class="new3">
</body>
</html>
```

div

- Se al posto di **<p>** si usa il tag **<div>** il blocco di testo va a capo, ma - a differenza del paragrafo - non lascia spazi prima e dopo la sua apertura
 - Attributi: **align, style, class, id**
- È l'elemento di tipo **block** per eccellenza
 - È un segnaposto dove verranno posizionati i risultati Ajax...
- Esempio (due **<div>**):

```
<div>Blocco di testo 1</div> <div>Blocco di testo 2</div>
```

- Vengono visualizzati così:

Blocco di testo 1

Blocco di testo 2

span

- Lo **** è un contenitore generico che può essere annidato (ad esempio) all'interno dei **<div>**
- È un elemento **inline**, e quindi non va a capo ma continua sulla stessa linea del tag che lo include
- Esempio (due ****):
Contenitore 1 Contenitore 2
- Visualizzazione: **Contenitore 1Contenitore 2**
- È un elemento molto utilizzato soprattutto insieme ai fogli di stile per dare un aspetto particolare ad un pezzo di testo in un blocco (per esempio, **per evidenziare**)
- Se non viene associato ad uno stile è *invisibile*

Contenitori di testo: riepilogo

- Ricapitolando: `<p>`, `<div>` e `` sono tre diversi tipi di contenitori di testo
- Si comportano in modo diverso:
 - `<p>` è un elemento di blocco e lascia spazio prima e dopo la propria chiusura
 - `<div>` è un elemento di blocco, non lascia spazio prima e dopo la propria chiusura, ma va a capo
 - `` è un elemento inline e quindi non va a capo
- `` può solo racchiudere elementi inline, `<div>` sia block che inline
- `` fluttua nel paragrafo esistente, `<div>` di default ha un linea vuota sopra e sotto

div/span and center

```
<div align="center">...</div>
```

- Is equivalent to

```
<center>...</center>
```

- *DIV is preferred in modern apps!*

<div>

Example: div & span

```
...
<body>
<p>Some text before the div</p>
<div style="border: inset; background-color: #cccccc; text-align="center">
    <h2>Using div and span</h2>
    <p>A list inside the div:</p>
    <ol>
        <li>List Item One</li>
        <li><span style="background-color: yellow;">
            List Item Two (inside span)
            </span>
        </li>
        <li>List Item Three</li>
    </ol>
</div>
<p>Some text after the div</p>
</body>
...
```

Example: div & span (2)

The screenshot shows a Mozilla Firefox window with the title "div and span - Mozilla Firefox". The address bar displays "div and span" and the URL "ew/WebContent/div.html". The page content is as follows:

Some text before the div

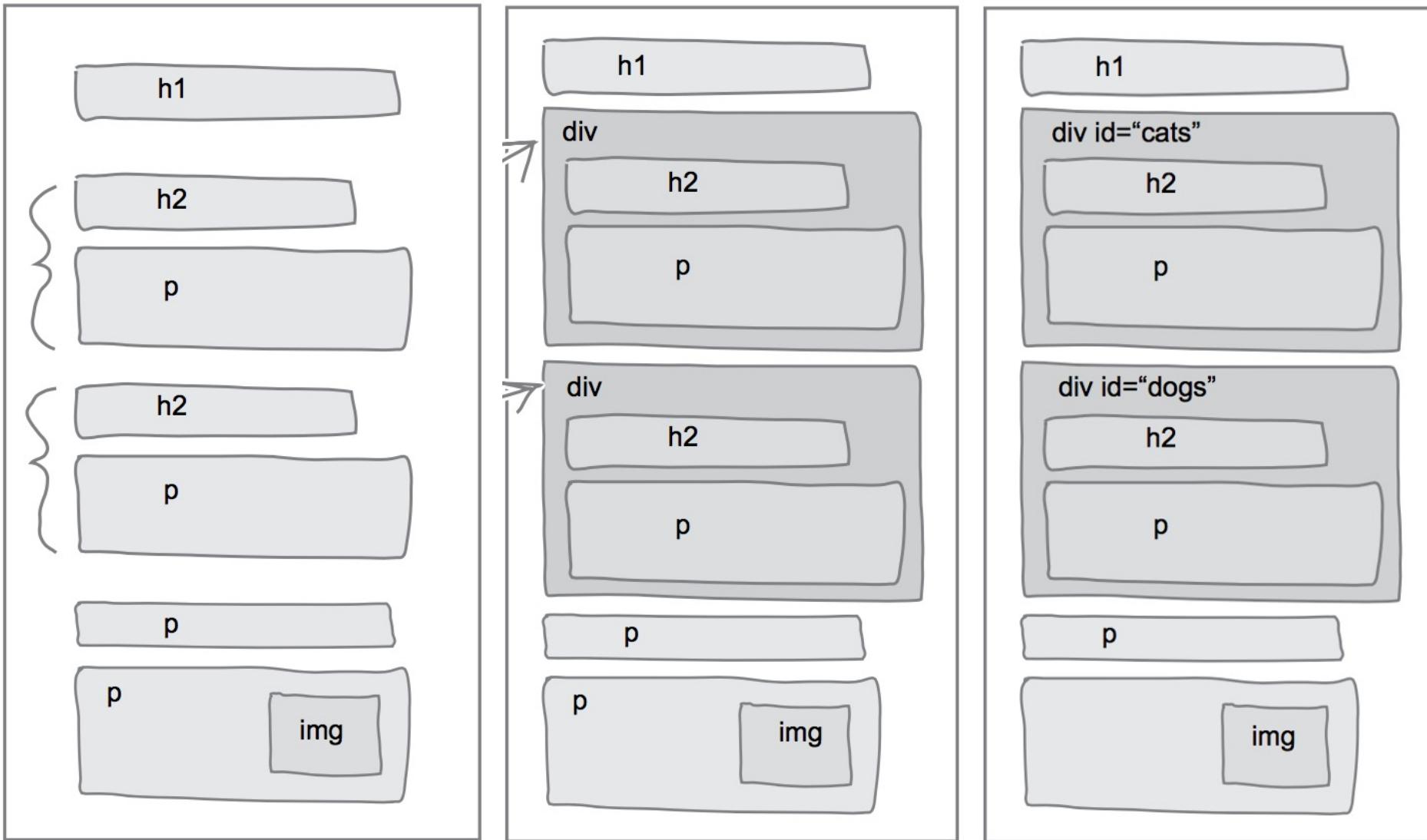
Using div and span

A list inside the div:

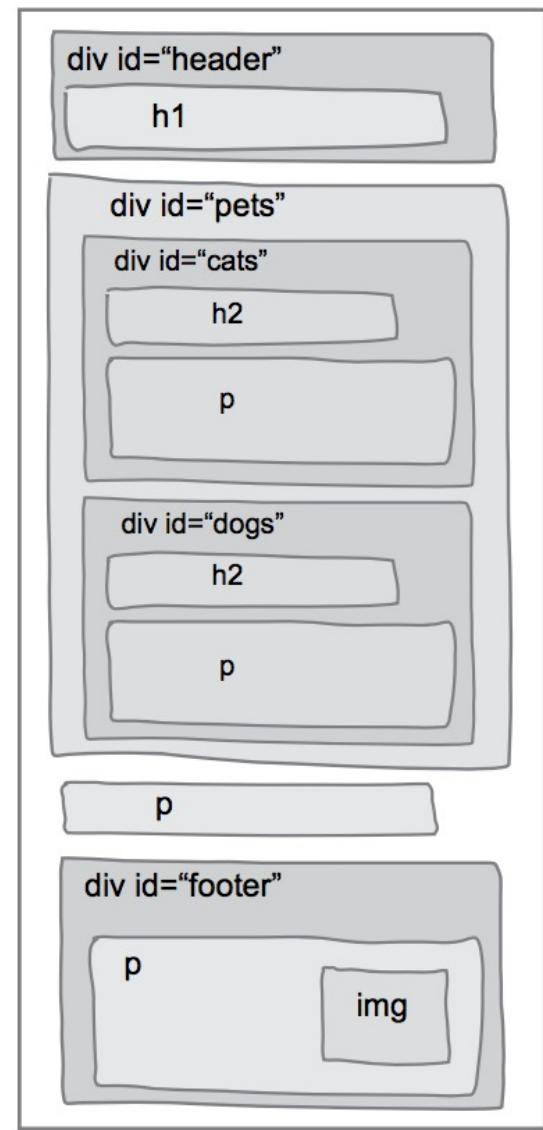
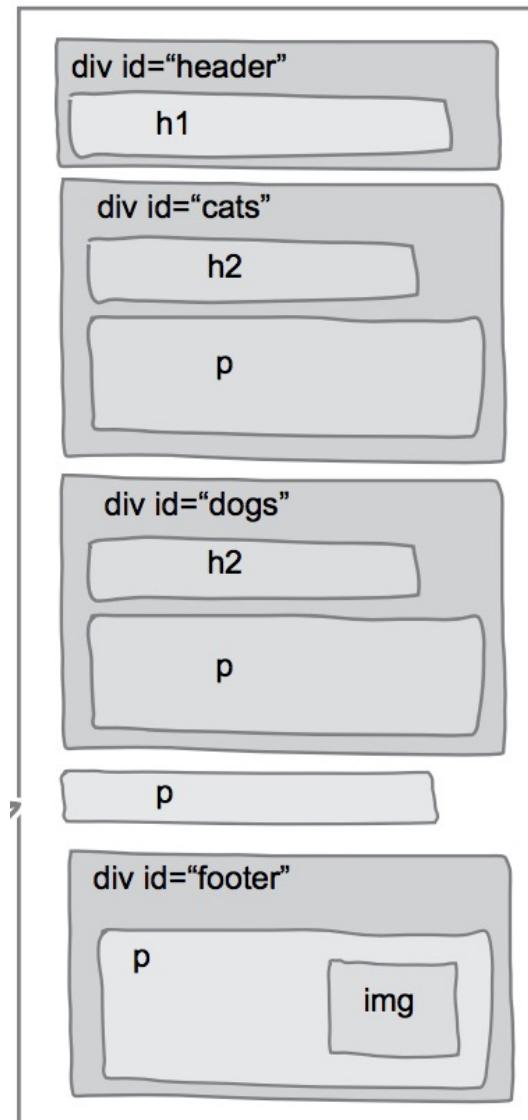
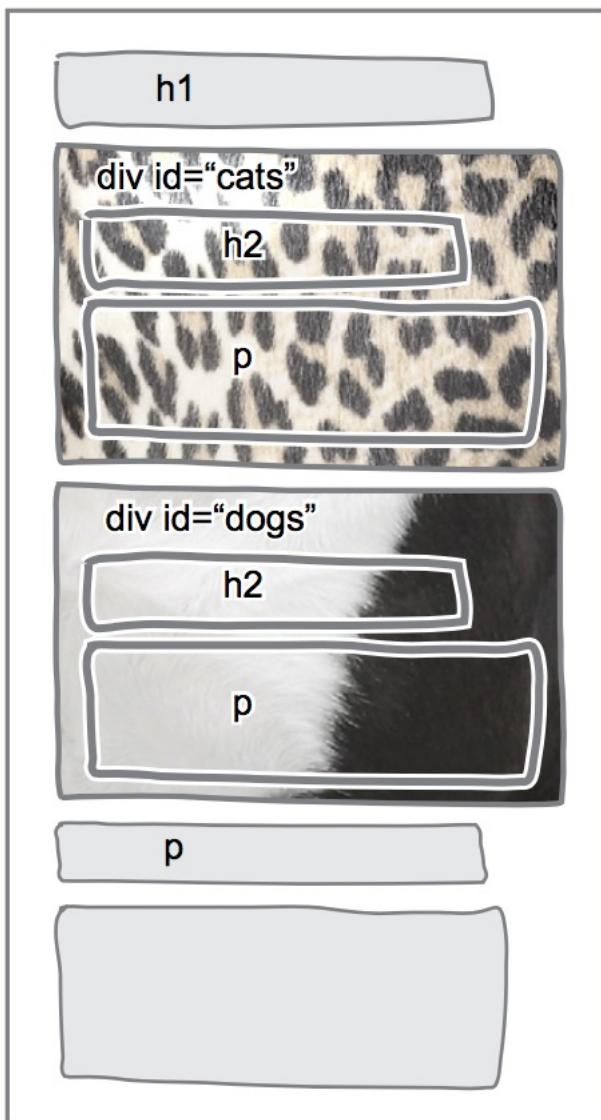
- 1. List Item One
- 2. List Item Two (inside span)
- 3. List Item Three

Some text after the div

Advanced Web Construction



Advanced Web Construction





Head First Lounge

Welcome to the Head First Lounge

The Head First Lounge is, no doubt, the biggest trendsetter in Webville. Stop in to sample the eclectic offering of elixirs, teas, and coffees, or stay a bit longer and enjoy the multicultural culinary menu that combines a harmony of taste, texture, and color with the best in fresh and healthy ingredients.

During your stay at the lounge, you'll enjoy a smooth mixture of ambient and mystic sounds, filling the lounge and adding an extra dimension to your dining experience. The decor surrounds you with the relaxing sentiments of sights from eras past. And, don't forget, the lounge offers free wireless access to the Internet, so bring your laptop.



Our guarantee: at the lounge, we're committed to providing you, our guest, with an exceptional experience every time you visit. Whether you're just stopping by to check in on email over an elixir, or are here for an out-of-the-ordinary dinner, you'll find our knowledgeable service staff pay attention to every detail. If you're not fully satisfied, have a Blueberry Bliss Elixir on us.

But that's not all; at night, join us in the backroom as our resident DJ spins a choice selection of trance and drum&bass beats across our spacious tiki-themed dance floor. Or just hang out in one of our comfy white vinyl booths at the dance bar. You can have your elixirs delivered from the main lounge right to the dance floor. If you've had enough of the beat, just head back to the lounge area to relax. And, no matter where you find yourself in the lounge, you'll always be connected with our wireless Internet access.

Now that you've experienced the lounge *virtually*, isn't it time to check us out *for real*? We're located right in the heart of Webville, and we've created some [detailed directions](#) to get you here in record time. No reservations necessary; come and join us anytime.

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**

© 2012, Head First Lounge

All trademarks and registered trademarks appearing on this site are the property of their respective owners.

Improvements...

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](#).

lounge (HTML CSS code.zip)

- Open up your “lounge.html” file in the HTML CSS code
- Locate the elixir elements, and then insert opening and closing `<div>` tags around them.

```
<div id="elixirs">
```

```
    <h2>Weekly Elixir Specials</h2>
```

```
    ...
```

```
    <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>
```

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.

Elixirs 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.

located right in the heart of Webville, and we've 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.

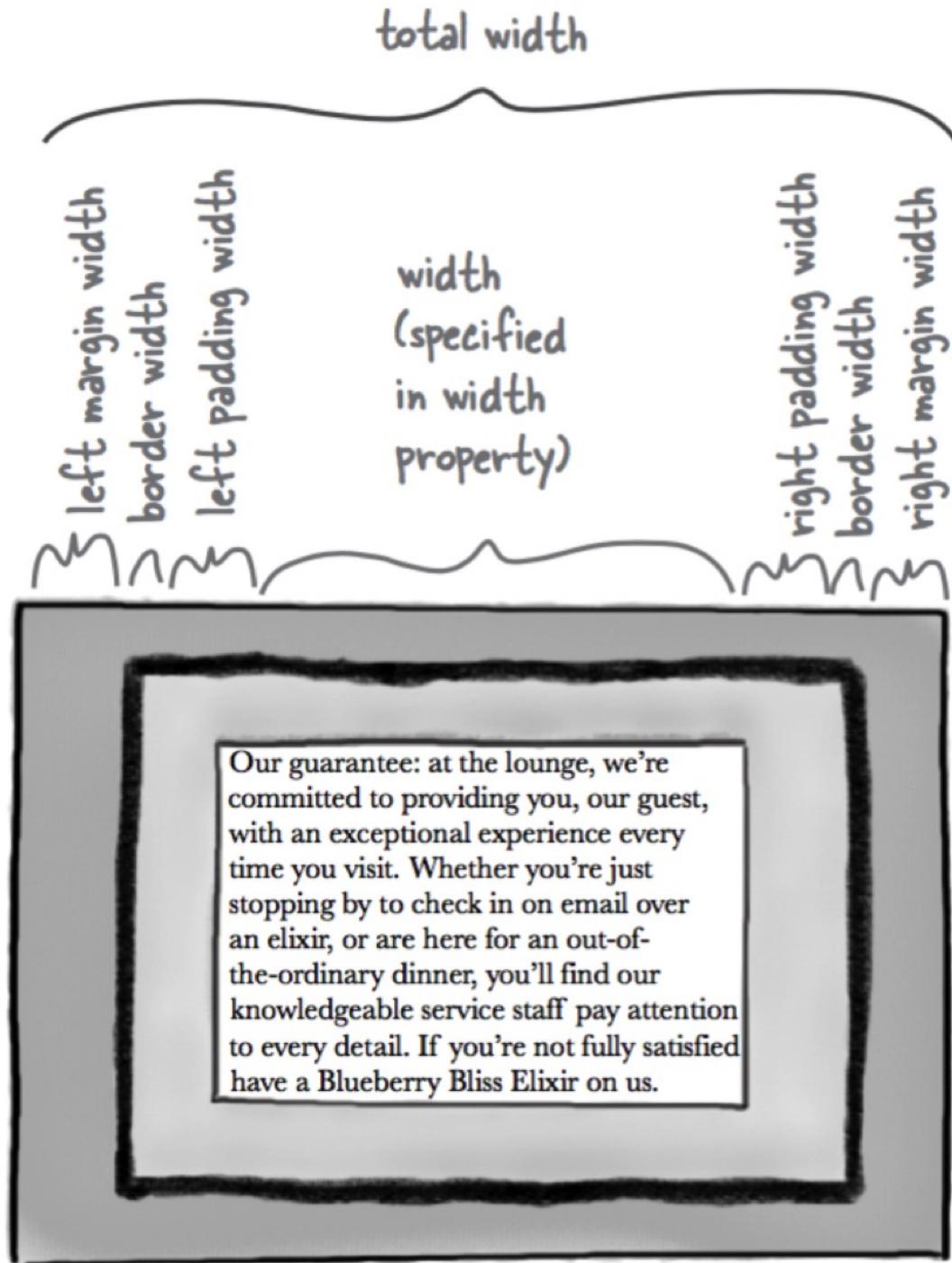
 **Chai Chiller**
Not your traditional chai, this elixir mixes maté with chai spices and adds a chocolate kick for a caffeinated taste sensation on ice.

 **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

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
- You specify the width of the content area, the padding, the border, and the margin
 - All of that added together is the width of the entire element



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;
```

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.

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

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

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...

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

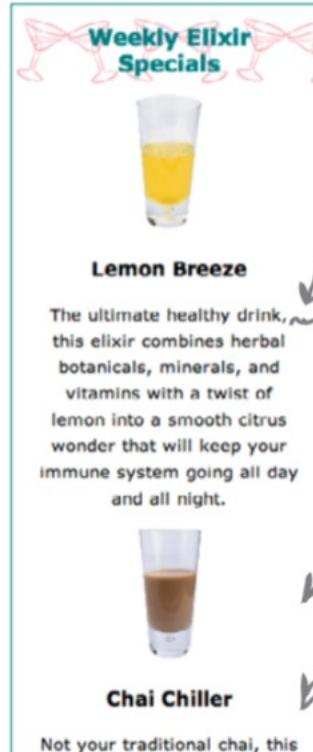
Use text-align on block elements to align the text they contain. Here, we're going to center-align the text.

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

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.

- All the text inside the `<div>` element is in nested block elements, but it is all aligned now
 - That's because these block elements *inherit* the **text-align** property from the `<div>`

We're located right in the heart of Webville, and we've created some [detailed directions](#) to get you here in record time. No reservations necessary; come and join us anytime.



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.

Select descendants

Leave a space between
the parent name and
the descendant name.

Here's the
parent element.

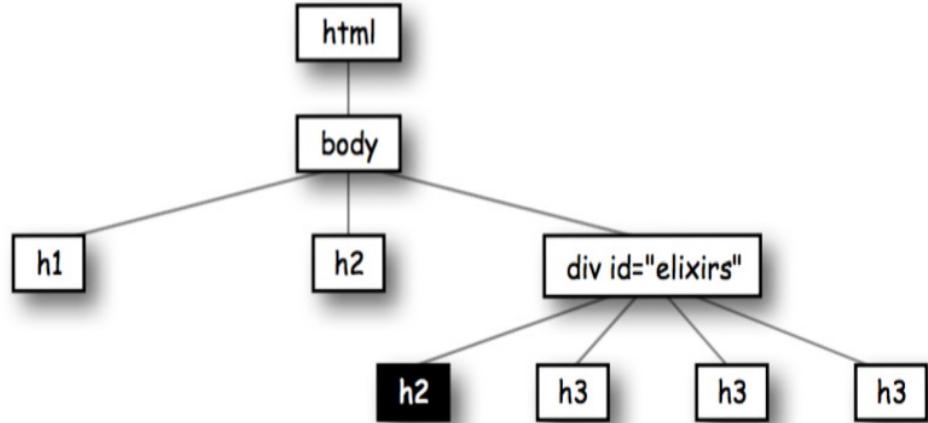
And here's its
descendant.

```
div h2 {  
    color: black;  
}
```

This rule says to select
any `<h2>` that is a
descendant of a `<div>`.

Write the rest of
your rule just like
you always do.

Here's what this rule
selects in the lounge.



**Problem: if someone created another `<div>` in
the “lounge.html” file, she’d get black `<h2>`
text, even if she didn’t want it**

#

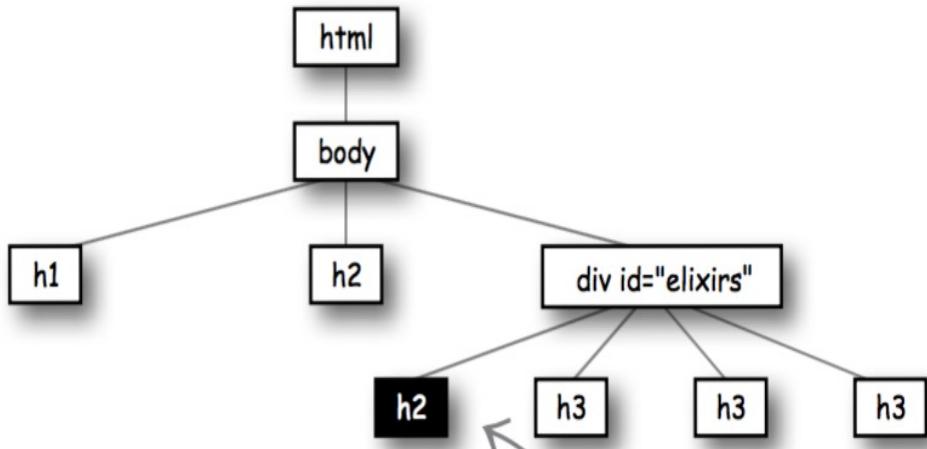
Now the parent element is the element with the id elixirs.

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".

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



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>`.

Adding the span

- Locate the “Who’s playing at the Lounge” heading

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>
```

Styling the span

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.

What's playing at the Lounge

We're frequently asked about the music we play at the lounge, and no wonder,

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

The <a> element

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.

Not your trad
elixir mixes
spices and
chocolat
caffinated
o

Chai

Black B

Want to boos
Try our Bla
elixir, ma
oolong tea an
espresso.
thank you

Join us any e
and all our
e

What's playin

We're frequently as
wonder, it's great s
updated weekly. Er

<a> styles

Notice we have the element <a>, followed by a : (colon), followed by the state we want to select. Make sure you don't have any spaces in these selectors (e.g., a : link won't work!)

```
a:link {  
    color: green;  
}  
  
a:visited {  
    color: red;  
}  
  
a:hover {  
    color: yellow;  
}
```

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

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

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

Pseudo-classes

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 styling JUST 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 a try!

Float

- Move the div elixir element under the logo of lounge.html
- Add to elixirs in the css file
float: right;



Welcome to the Head First Lounge

The Head First Lounge is, no doubt, the biggest trendsetter in Webville. Stop in to sample the eclectic offering of elixirs, teas, and coffees, or, stay a bit longer and enjoy the multicultural culinary menu that combines a harmony of taste, texture, and color with the best in fresh and healthy ingredients.

During your stay at the lounge, you'll enjoy a smooth mixture of ambient and mystic sounds, filling the lounge and adding an extra dimension to your dining experience. The decor surrounds you with the relaxing sentiments of sights from eras past. And, don't forget, the lounge offers free wireless access to the Internet, so bring your laptop.



Our guarantee: at the lounge, we're committed to providing you, our guest, with an exceptional experience every time you visit. Whether you're just stopping by to check in on email over an elixir, or are here for an out-of-the-ordinary dinner, you'll find our knowledgeable service staff pay attention to every detail. If you're not fully satisfied, have a Blueberry Bliss Elixir on us.

But that's not all; at night, join us in the backroom as our resident DJ spins a choice selection of trance and drum&bass beats across our spacious tiki-themed dance floor. Or just hang out in one of our comfy white vinyl booths at the dance bar. You can have your elixirs delivered from the main lounge right to the dance floor. If you've had enough of the beat, just head back to the lounge area to relax. And, no matter where you find yourself in the lounge, you'll always be connected with our wireless Internet access.

Now that you've experienced the lounge *virtually*, isn't it time to check us out *for real*? We're located right in the heart of Webville, and we've created some [detailed directions](#) to get you here in record time. No reservations necessary; come and join us anytime.

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*



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 mate with chai spices and adds an extra chocolate kick for a caffeinated taste sensation on ice.



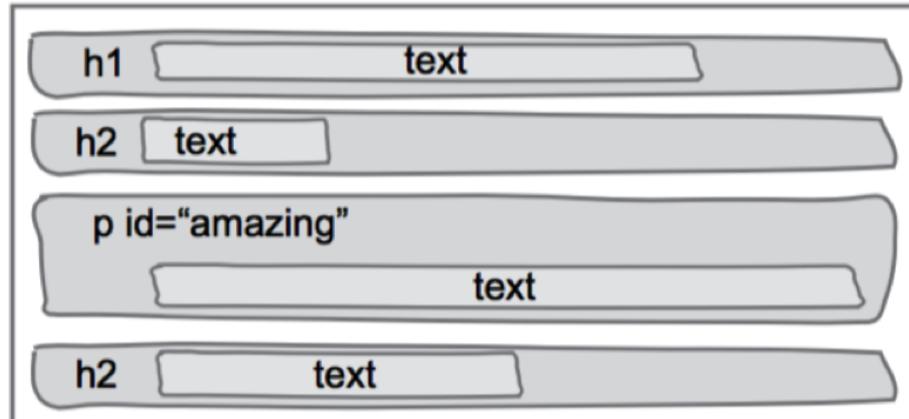
Black Brain Brew

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](#).

Remind: How to float an element

- give it an identity



- give it a width

```
#amazing {  
    width: 200px;  
}
```

- float it

```
#amazing {  
    width: 200px;  
    float: right;  
}
```

The footer

```
<div id="footer">
  <small>
    © 2012, Head First Lounge<br>
    All trademarks and registered trademarks appearing on this site are
    the property of their respective owners.
  </small>
</div>
```

```
#footer {
  font-size: 50%;
  text-align: center;
  line-height: normal;
  margin-top: 30px;
}
```

Now that you've experienced the lounge *virtually*, isn't it time to check us out *for real*? We're located right in the heart of Webville, and we've created some [detailed directions](#) to get you here in record time. No reservations necessary; come and join us anytime.

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*



Head First Lounge

Welcome to the Head First Lounge

The Head First Lounge is, no doubt, the biggest trendsetter in Webville. Stop in to sample the eclectic offering of elixirs, teas, and coffees, or, stay a bit longer and enjoy the multicultural culinary menu that combines a harmony of taste, texture, and color with the best in fresh and healthy ingredients.

During your stay at the lounge, you'll enjoy a smooth mixture of ambient and mystic sounds, filling the lounge and adding an extra dimension to your dining experience. The decor surrounds you with the relaxing sentiments of sights from eras past. And, don't forget, the lounge offers free wireless access to the Internet, so bring your laptop.



Our guarantee: at the lounge, we're committed to providing you, our guest, with an exceptional experience every time you visit. Whether you're just stopping by to check in on email over an elixir, or are here for an out-of-the-ordinary dinner, you'll find our knowledgeable service staff pay attention to every detail. If you're not fully satisfied, have a Blueberry Bliss Elixir on us.

But that's not all; at night, join us in the backroom as our resident DJ spins a choice selection of trance and drum&bass beats across our spacious tiki-themed dance floor. Or just hang out in one of our comfy white vinyl booths at the dance bar. You can have your elixirs delivered from the main lounge right to the dance floor. If you've had enough of the beat, just head back to the lounge area to relax. And, no matter where you find yourself in the lounge, you'll always be connected with our wireless Internet access.

Now that you've experienced the lounge *virtually*, isn't it time to check us out *for real*? We're located right in the heart of Webville, and we've created some [detailed directions](#) to get you here in record time. No reservations necessary; come and join us anytime.

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**

© 2012, Head First Lounge

All trademarks and registered trademarks appearing on this site are the property of their respective owners.

Congratulations!

The example is complete



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](#).

Modern HTML

- Much of the new HTML5 markup is aimed at recognizing how people structure their pages with <div>s and providing markup that is more specific, and better suited for certain kinds of structure



- Esempio starbuzz in HTML CSS code

Starbuzz Coffee

file:///chapter12/starbuzz/ir/index.html

Starbuzz Coffee



...providing all the caffeine you need to power your life. just drink it.

BEVERAGES

House Blend, \$1.49
Mocha Cafe Latte, \$2.35
Cappuccino, \$1.89
Chai Tea, \$1.85

ELIXIRS

We proudly serve elixirs brewed by our friends at the Head First Lounge.

Green Tea Cooler, \$2.99
Raspberry Iced Concentration, \$2.99
Blueberry Bliss Elixir, \$2.99
Cranberry Antioxidant Blast, \$2.99
Chai Chiller, \$2.99
Black Brain Brew, \$2.99

QUALITY COFFEE, QUALITY CAFFEINE

At Starbuzz Coffee, we are dedicated to filling all your caffeine needs through our quality coffees and teas. Sure, we want you to have a great cup of coffee and a great coffee experience as well, but we're the only company that actively monitors and optimizes caffeine levels. So stop by and fill your cup, or order online with our new Bean Machine online order form, and get that quality Starbuzz coffee that you know will meet your caffeine standards.

And, did we mention *caffeine*? We've just started funding the guys doing all the wonderful research at the [Caffeine Buzz](#). If you want the latest on coffee and other caffeine products, stop by and pay them a visit.

OUR STORY

"A man, a plan, a coffee bean". Okay, that doesn't make a palindrome, but it resulted in a damn good cup of coffee. Starbuzz's CEO is that man, and you already know his plan: a Starbuzz on every corner.

In only a few years he's executed that plan and today you can enjoy Starbuzz just about anywhere. And, of course, the big news this year is that Starbuzz teamed up with Head First readers to create Starbuzz's Web presence, which is growing rapidly and helping to meet the caffeine needs of a whole new set of customers.

STARBUZZ COFFEE BEVERAGES

We've got a variety of caffeinated beverages to choose from at Starbuzz, including our [House Blend](#), [Mocha Cafe Latte](#), [Cappuccino](#), and a favorite of our customers, [Chai Tea](#).

We also offer a variety of coffee beans, whole or ground, for you to take home with you. Order your coffee today using our online [Bean Machine](#), and take the Starbuzz Coffee experience home.



ORDER ONLINE with the [BEAN MACHINE](#)
[FAST](#)
[FRESH](#)
[TO YOUR DOOR](#)

Why wait? You can order all our fine coffees right from the Internet with our new, automated Bean Machine. How does it work? Just click on the Bean Machine link, enter your order, and behind the scenes, your coffee is roasted, ground (if you want), packaged, and shipped to your door.

© 2012, Starbuzz Coffee
All trademarks and registered trademarks appearing on this site are the property of their respective owners.

```
<div id="header">
```

```
<div id="drinks">
```

```
<div id="main">
```

```
<div id="sidebar">
```

```
<div id="footer">
```



We're not showing the super-detailed structure of the page,
so for now just focus on this large-grained structure.

```
<div class="header">
  <div class="navigation">
    <ul class="nav-list">
      <li><a href="#" title="Home">Home</a></li>
      <li><a href="#" title="About">About</a></li>
    </ul>
  </div> <!--end of navigation -->
</div> <!--end of header -->
```

we do it with HTML5

```
<header>
  <nav>
    <ul id="nav-list">
      <li><a href="#" title="Home">Home</a></li>
      <li><a href="#" title="About">About</a></li>
    </ul>
  </nav>
</header>
```

<header>

<section id="drinks">



Each of these "sections" groups together a set of related content; that's just what the <section> element is for.

<section id="main">

<aside>

The sidebar is really peripheral content; we can place that in an aside element, given that's exactly what the <aside> is for.

<footer>

1

Add the <header> element

Start by replacing the <div id="header"> with a <header> element. Like this:

```
<div id="header">
  <header>           ← Remove the <div> tags and replace
    
    
  </header>
</div>
```

2

Add the <footer> element

Do the same for the <div id="footer">, only replace it with a <footer> element:

```
<div id="footer">
  <footer>
    © 2012, Starbuzz Coffee
    <br>
    All trademarks and registered trademarks appearing on
    this site are the property of their respective owners.
  </footer>
</div>
```

5 Change the sidebar to an <aside>

Now let's change the "sidebar" `<div>` to an `<aside>` element:

```
<div id="sidebar">
  <aside>
    <p class="beanheading">
      
      ...
    </p>
    <p> ...
    </p>
  </aside>
</div>
```

We decided to save a few trees (or bits) by abbreviating the content a little; just make sure you keep all the original content in the page and change the `<div>` tags to `<aside>` tags.

Update your CSS for the new elements

```
}

#header {
  header {
    background-color: #675c47;
    margin: 10px 10px 0px 10px;
    height: 108px;
  }
}

#header img#headerSlogan {
  header img#headerSlogan {
    float: right;
  }
}
```



First, remove the `#` mark from the header rules. We're going from targeting a `<div>` with an id of "header" to an element named header.



```
#sidebar {
  aside {
    display:           table-cell;
    background:       #efe5d0 url(images/backgroun
    font-size:        105%;
    padding:          15px;
    vertical-align:   top;
  }
}

#footer {
  footer {
    background-color: #675c47;
    color:            #efe5d0;
    text-align:       center;
    padding:          15px;
    margin:           0px 10px 10px 10px;
    font-size:        90%;
  }
}
```



Finally, we need to select the footer element.

<section> elements

The HTML without the
<section> element

```
<div id="tableContainer">
  <div id="tableRow">
    <div id="drinks">
      ...
    </div>
    <div id="main">
      ...
    </div>
    <aside>
      ...
    </aside>
  </div> <!-- tableRow -->
</div> <!-- tableContainer -->
```

The CSS as it is now for
#drinks and #main.

```
#drinks {
  display: table-cell;
  background-color: #efe5d0;
  width: 20%;
  padding: 15px;
  vertical-align: top;
}

#main {
  display: table-cell;
  background: url(images/background.gif) top left;
  font-size: 105%;
  padding: 15px;
  vertical-align: top;
}
```

The HTML with the
<section> element

```
<div id="tableContainer">
  <div id="tableRow">
    <section id="drinks">
      ...
    </section>
    <section id="main">
      ...
    </section>
    <aside>
      ...
    </aside>
  </div> <!-- tableRow -->
</div> <!-- tableContainer -->
```

The CSS updated for
the two sections

```
section#drinks {
  display: table-cell;
  background-color: #efe5d0;
  width: 20%;
  padding: 15px;
  vertical-align: top;
}

section#main {
  display: table-cell;
  background: #efe5d0 url(
  font-size: 105%;
  padding: 15px;
  vertical-align: top;
}
```

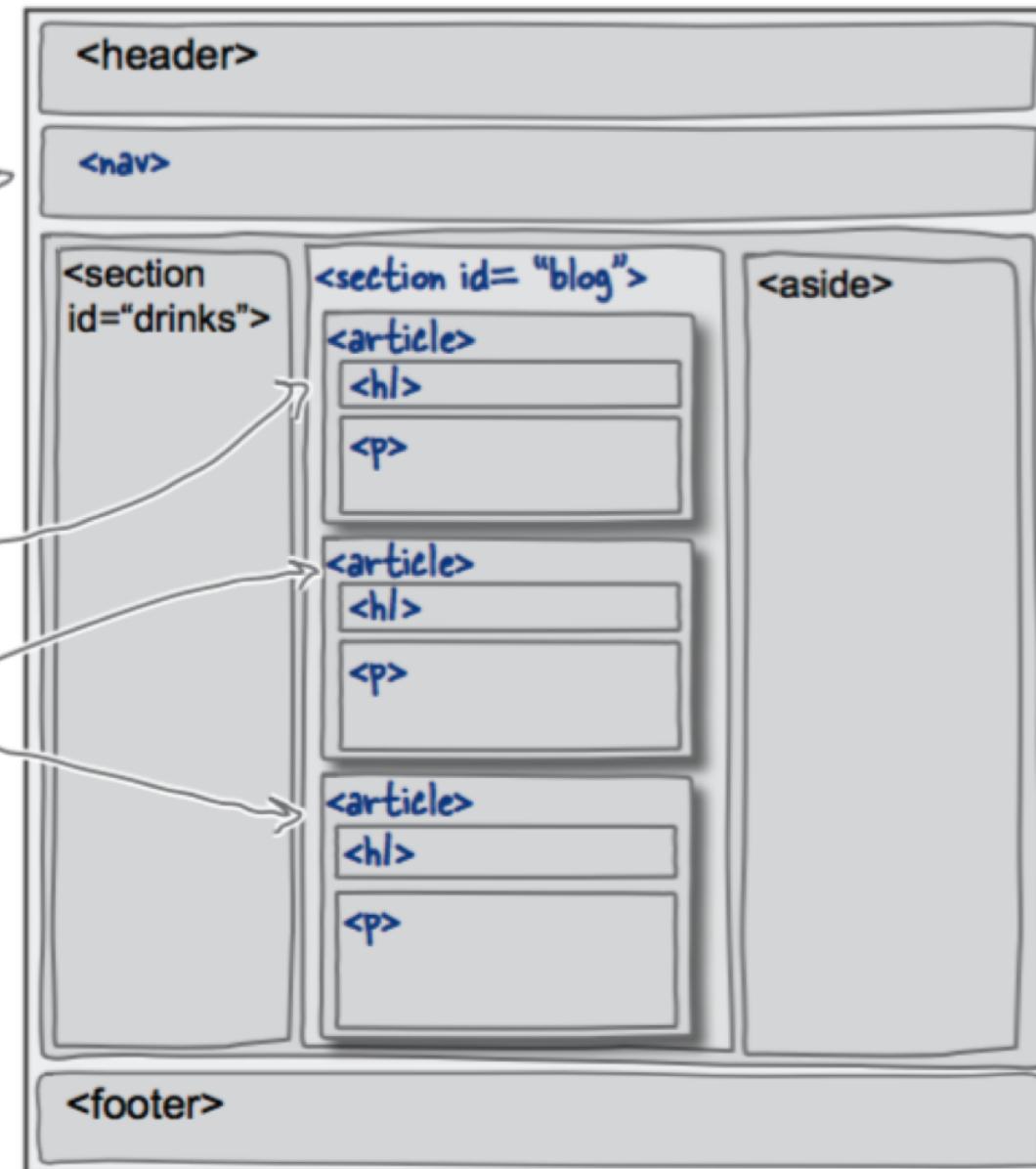
The new blog page. It's like the home page, except the middle section is now blog posts and we have a navigation menu below the header.

Navigation menu and articles

We used the `<nav>` element for the navigation menu.

We put the blog "section" of the page in a `<section>` element because `<section>` is used to group related content together.

We put each blog post in its own `<article>` element because each blog post is a self-contained item (that is, you could take articles away without affecting the readability of the ones you left behind).



Blog page

- Use “blog.html” in starbuzz HTML CSS code, use a **<section>** element for the blog section and an **<article>** element for each blog post

```
<section id="blog">
  <article>
    <h1>Starbuzz meets social media</h1>
    <p>
      Here at Starbuzz we're embracing the social media craze. In fact,
      we're going further than any of our competitors and we're very close...
    </p>
    <p>
      Sound like science fiction? It's not; I'm already testing our final
      prototype social network cup as I write this...
    </p>
    <p>
      So, keep your eyes out for this amazing new cup. And I'll be
      releasing a video teaser soon to tell you all about this new invention,
      straight from Starbuzz Coffee.
    </p>
  </article>
```

We're using a **<section>** element for the middle column, just like we did for “main” in the index.html file.

We're only showing part of each post here.

CSS for the blog page

```
<!DOCTYPE html>
<html>
  <head>
    <meta charset="utf-8">
    <title>Starbuzz Coffee - Blog</title>
    <link rel="stylesheet" type="text/css" href="starbuzz.css">
  </head>
  ...

```

Here's the link to the CSS...

...and while we're here, go ahead and update the title of the page.

```
section#main, section#blog {
  display: table-cell;
  background: #efe5d0 url(images/background.gif) top left;
  font-size: 105%;
  padding: 15px;
  vertical-align: top;
}
```

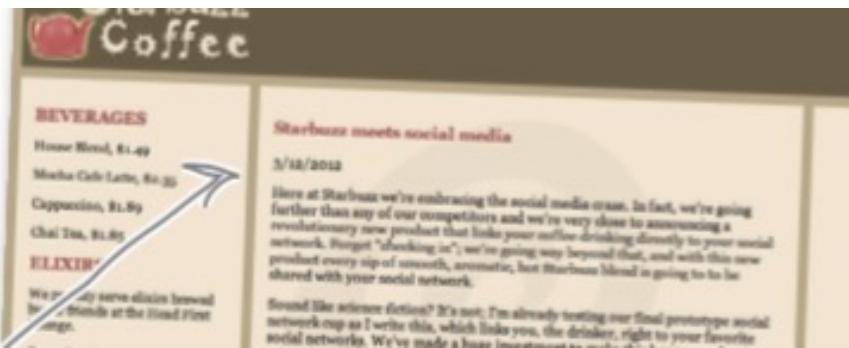
We can use the same rule for both `<section>` elements by using the two selectors separated by a comma. This says, apply all these properties to both of the selected elements.

Even though the two elements, the "main" `<section>` and the "blog" `<section>`, are on two different pages, this will work because both pages link to the same CSS file.

<time> element

- Add the following dates to blog.html

```
<article>
  <h1>Starbuzz meets social media</h1>
  <time datetime="2012-03-12">3/12/2012</time>
  ...
</article>
<article>
  <h1>Starbuzz uses computer science</h1>
  <time datetime="2012-03-10">3/10/2012</time>
  ...
</article>
<article>
  <h1>Most unique patron of the month</h1>
  <time datetime="2012-02-18">2/18/2012</time>
  ...
</article>
```



add more <header>

```
...  
<section id="blog">  
<article>  
  <header>  
    <h1>Starbuzz meets social media</h1>  
    <time datetime="2012-03-12">3/12/2012</time>  
  </header>  
  <p>...</p>  
</article>  
  
<article>  
  <header>  
    <h1>Starbuzz uses computer science</h1>  
    <time datetime="2012-03-10">3/10/2012</time>  
  </header>  
  <p>...</p>  
</article>
```

Place your <header> element here, around the heading and the time elements.



```
header {  
    background-color: #675c47;  
    margin: 10px 10px 0px 10px;  
    height: 108px;  
}
```

This header rule height property causes the background color to be set and space to be added to ALL headers in the page, not just the main header. And the margin isn't helping either.

- creating a class just for the `<header>` at the top of the page

```
<body>
  <header class="top">
    
    
  </header>
  ...

```

Add the class "top" to the first
`<header>` element in the page.

```
header.top {
  background-color: #675c47;
  margin:           10px 10px 0px 10px;
  height:          108px;
}
```

```
header.top img#headerSlogan {
  float:           right;
}
```

- Notice that now the <header> rules apply only to the <header> at the very top of the page, which is just what we want. Meanwhile, the article <header>s get the default style, which will work fine as well



Adding the navigation bar

- In the file “blog.html”

```
<body>
  <header class="top">
    
    
  </header>
  <ul>
    <li><a href="index.html">HOME</a></li>
    <li class="selected"><a href="blog.html">BLOG</a></li>
    <li><a href="">INVENTIONS</a></li>
    <li><a href="">RECIPES</a></li>
    <li><a href="">LOCATIONS</a></li>
  </ul>
  ...
</body>
```

Navigation CSS

```
ul {  
    background-color: #efe5d0; ← We're adding a background color, and some margins and padding.  
    margin: 10px 10px 0px 10px; Notice that the bottom margin is 0 because the table display  
    list-style-type: none; ← already has a 10px border-spacing at the top.  
    padding: 5px 0px 5px 0px;  
}  
  
ul li {  
    display: inline; ← Also notice that we've removed the bullets from the list items.  
    padding: 5px 10px 5px 10px;  
}
```

Here, we're changing the display of each list item from "block" to "inline", so now the list items won't have a carriage return before and after; they'll all flow into one line on the page like regular inline elements do.

Navigation CSS (2)

```
ul li a:link, ul li a:visited {  
    color: #954b4b;  
    border-bottom: none;  
    font-weight: bold;  
}  
  
ul li.selected {  
    background-color: #c8b99c;  
}
```



We want the links in the navigation list to look a bit different from the rest of the links in the page, so we override the other rules for <a> (above this rule in the CSS) with a rule that sets properties for both the links and the visited state of the links (so they look the same).



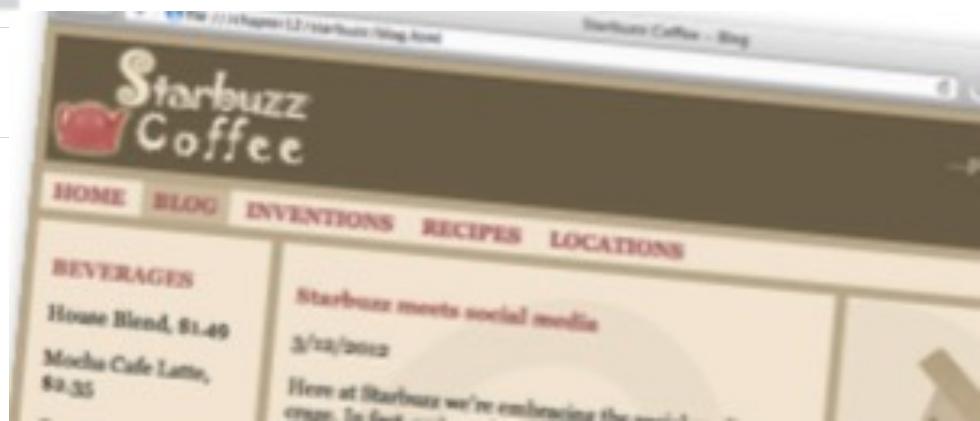
And finally, we're setting the background of the element with the class "selected" so the navigation item corresponding to the page we're on looks different from the rest.

<nav> element

Here's the <nav> starting tag, and we're enclosing the entire navigation list within a <nav> element.

```
<nav>
  <ul>
    <li><a href="index.html">HOME</a></li>
    <li class="selected"><a href="blog.html">BLOG</a></li>
    <li><a href="">INVENTIONS</a></li>
    <li><a href="">RECIPES</a></li>
    <li><a href="">LOCATIONS</a></li>
  </ul>
</nav>
```

```
nav {  
    background-color: #efe5d0;  
    margin: 10px 10px 0px 10px;  
}  
  
nav ul {  
    margin: 0px;  
    list-style-type: none;  
    padding: 5px 0px 5px 0px;  
}  
  
nav ul li {  
    display: inline;  
    padding: 5px 10px 5px 10px;  
}  
  
nav ul li a:link, nav ul li a:visited {  
    color: #954b4b;  
    border-bottom: none;  
    font-weight: bold;  
}  
  
nav ul li.selected {  
    background-color: #c8b99c;
```



The structure of a page in HTML 5



Add a video

The screenshot shows a web browser displaying the Starbuzz Coffee blog. The main content area features a video player with a thumbnail image of a man holding a coffee cup. Below the video, the text reads "Starbuzz meets social media". To the right of the video, there is a sidebar advertisement for "BEAN MACHINE" featuring a coffee bean and the text "ORDER ONLINE with the BEAN MACHINE". The sidebar also includes the words "FAST", "FRESH", and "TO YOUR DOOR". The top right corner of the page has a slogan: "...providing all the caffeine you need to power your life. just drink it."

Starbuzz Coffee - Blog

Re://Chapter12/starbuzz/BlogVideos.html

Starbuzz
Coffee

Home Blog Inventories Recipes Locations

REVERAGES

- Hawaiian Blend, \$1.99
- Mocha Caffe Latte, \$1.99
- Cappuccino, \$1.99
- Chai Tea, \$1.99

ELIXIRS

- We proudly serve elixirs brewed by our Monk at the Head First Lounge.
- Green Tea Cooler, \$1.99
- Raspberry Ice Concentration, \$1.99
- Blueberry Blue Haze, \$1.99
- Cherry Antioxidant Blast, \$1.99
- Chai Chiller, \$1.99
- Rockin' Roast Brew, \$1.99

Starbuzz inventories... Tweet Slip
8/13/2012

As promised, today I'm proud to announce that Starbuzz Coffee is launching the Tweet Slip app, a special Starbuzz Coffee app that tweets each time you take a sip! Check out my video of me taking a sip:

Starbuzz meets social media
8/13/2012

Here at Starbuzz we're embracing the social media craze. In fact, we're going further than any of our competitors and we're very close to announcing a revolutionary new product that links your coffee drinking directly to your social network. Stay tuned! "Thinking in..." we're going to launch that, and with the new product comes an app...

...providing all the caffeine you need to power your life. just drink it.

ORDER ONLINE with the BEAN MACHINE
FAST
FRESH
TO YOUR DOOR

- Go ahead and add this HTML just under the `<section>` element, above the other articles (in “blog.html”):

```
<article>
  <header>
    <h1>Starbuzz launches...Tweet Sip</h1>
    <time datetime="2012-05-03">5/3/2012</time>
  </header>
  <p>
    Check out my video of our new invention.
  </p>
  <!-- add the video here!! -->
</article>
```

<video> tag

Here we have the opening video tag, with quite a few attributes...

```
<video controls autoplay width="512" height="288" src="video/tweetsip.mp4">  
</video>
```

And here we have the closing tag.

We'll come back to the details of all these attributes in a sec, but for now notice we're setting the width and height of the element, along with specifying a src URL for the video.

We'll see what content we can place in here in a bit as well...

- For compatibility reason, in HTML 5 we can specify the **type** attribute in the <source> tag (multiple sources are allowed):

```
<video id="video" controls autoplay width="512" height="288">  
    <source type="video/mp4" src="video/tweetsip.mp4" >  
</video>
```

Video attributes

```
<video controls  
        autoplay  
        width="512" height="288"  
        src="video/tweetsip.mp4" ↗ The source location of the video  
        poster="images/poster.png"  
        id="video">  
</video>
```

Of course, we can add an id to the element too in case we want to apply some styling.

↗ If you like, you can supply an optional poster image to show when the movie is not playing.

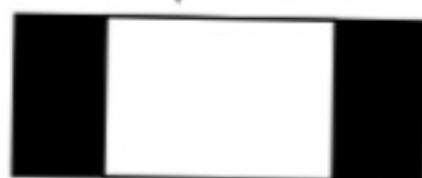
Video attributes (2)

- **controls**
 - It is a **Boolean** attribute. It's either there or it's not. If it is there, then the browser will add its built-in controls to the video display. The controls vary by browser, so check out each browser to see what they look like
- **autoplay**
 - The autoplay Boolean attribute tells the browser to start playing the video as soon as it has enough data. For the videos we're demoing with, you'll probably see them start to play almost immediately
- **poster**
 - The browser will typically display one frame of the video as a “poster” image to represent the video. If you **remove the autoplay attribute**, you'll see this image displayed before you click play

Video attributes (3)

- **loop**
 - It automatically restarts the video after it finishes playing
- **src**
 - it is a URL that tells the video element where to find the source file
- The **width** and **height** attributes set the width and height of the video display area (also known as the “**viewport**”)
 - If you specify a poster, the poster image will be scaled to the width and height you specify
 - The video will also be scaled, but will maintain its aspect ratio (e.g., 4:3 or 16:9), so if there's extra room on the sides, or the top and bottom, the video will be letter-boxed or pillar-boxed to fit into the display area size
 - *You should try to match the native dimensions of the video if you want the best performance (so the browser doesn't have to scale in real time)*

Pillar-boxing



Letter-boxing





I'm not seeing any video. I've triple-checked the code and I have the video in the right folder. Any ideas?

Yes, it's probably the video format

Video formats

MP4 container with H.264 video and AAC audio

H.264 is licensed by the MPEG-LA group.

There is more than one kind of H.264; each is known as a “profile.”

MP4/H.264 is supported by Safari and IE9+. You may find support in some versions of Chrome.

Ogg container with Theora video and Vorbis audio

Theora is an open source codec.

Video encoded with Theora is usually contained in an Ogg file, with the .ogg file extension.

Ogg/Theora is supported by Firefox, Chrome, and Opera.

WebM container with VP8 video and Vorbis audio

WebM was designed by Google to work with VP8-encoded videos.

WebM/VP8 is supported by Firefox, Chrome, and Opera.

You'll find WebM-formatted videos with the .webm extension.

TOP SECRET

YOUR MISSION: VIDEO RECONNAISSANCE

GO OUT AND DETERMINE [REDACTED] THE CURRENT LEVEL OF SUPPORT FOR VIDEO IN EACH BROWSER BELOW (HINT: HERE ARE A FEW SITES THAT KEEP UP WITH SUCH THINGS: http://en.wikipedia.org/wiki/HTML5_video, <http://caniuse.com/#search=video>). ASSUME THE LATEST VERSION OF THE BROWSER. FOR EACH BROWSER/FEATURE PUT A CHECKMARK IF IT IS SUPPORTED. UPON YOUR RETURN, REPORT BACK FOR YOUR NEXT ASSIGNMENT!

iOS and Android devices (among others)

Browser \ Video	Safari	Chrome	Firefox	Mobile WebKit	Opera	IE9+	IE8	IE7 or <
H.264								
WebM								
Ogg Theora								

Video bullet points

- ✓ The **container** is the file format that's used to package up the video, audio, and metadata information. Common container formats include:
MP4, WebM, Ogg, and Flash Video
- ✓ The **codec** is the software used to encode and decode a specific encoding of video or audio. Popular web codecs include: H.264, VP8, Theora, AAC, and Vorbis
- The browser decides what video it can decode. Not all browser makers agree, so if you want to support everyone, you need multiple encodings



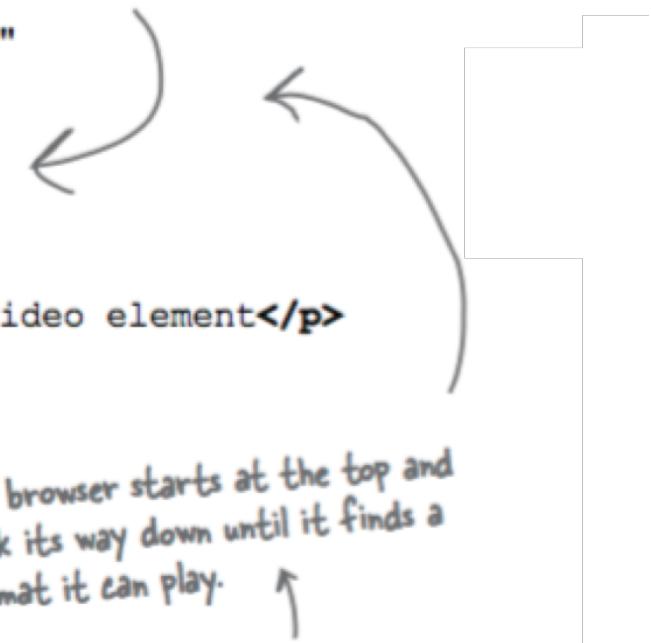
Cross-browser video

Notice we're removing the `src` attribute from the `<video>` tag...

```
<video controls autoplay width="512" height="288"  
      src="video/tweetsip.mp4">  
  <source src="video/tweetsip.mp4">  
  <source src="video/tweetsip.webm">  
  <source src="video/tweetsip.ogv">  
  <p>Sorry, your browser doesn't support the video element</p>  
</video>
```

This is what the browser shows if it doesn't support video.

...and adding three `<source>` tags, each with its own `src` attribute, each with a version of the video in a different format.



The browser starts at the top and work its way down until it finds a format it can play.

For each source, the browser loads the metadata of the video file to see if it can play it (which can be a lengthy process, although we can make it easier on the browser...see the next page).

Cross-browser video (2)

```
<video id="video" poster="video/prerollposter.jpg" controls width="480" height="360">
  <source src="video/preroll.mp4" type='video/mp4; codecs="avc1.42E01E, mp4a.40.2"'>
  <source src="video/preroll.webm" type='video/webm; codecs="vp8, vorbis"'>
  <source src="video/preroll.ogv" type='video/ogg; codecs="theora, vorbis"'>
  <p>Sorry, your browser doesn't support the video element</p>
</video>
```



If you don't know the codecs parameters, then you can leave them off and just use the MIME type. It will be a little less efficient, but most of the time, that's okay.

The codecs for mp4 are more complicated than the other two because h.264 supports various "profiles," different encodings for different uses (like high bandwidth vs. low bandwidth). So, to get those right, you'll need to know more details about how your video was encoded.



<iframe> Inline-frame

- The **<iframe>** tag specifies an inline frame
- An **inline frame** is used to embed another document within the current HTML document
 - Deprecated in HTML 4.01 but it is still used

```
<iframe  
    src ="interno.html"  
    width="100%">  
</iframe>
```

Documento
HTML principale

iframe
(doc HTML)

Playing a YouTube Video in HTML

1. Upload the video to YouTube
2. Take a note of the video id
3. Define an `<iframe>` element in your web page
4. Let the `src` attribute point to the video URL
5. Use the `width` and `height` attributes to specify the dimension of the player
6. Add any other parameters to the URL

```
<iframe width="420" height="315"  
src="http://www.youtube.com/embed/Dlf1_vuIR4I?autoplay=1">  
</iframe>
```

Youtube parameters

- **autohide**
 - Value 0: The player controls are always visible
 - Value 1: The player controls hides automatically when the video plays
 - Value 2 (default): If the player has 16:9 or 4:3 ratio, same as 1, otherwise same as 0
- **autoplay**
 - Value 0 (default): The video will not play automatically when the player loads
 - Value 1: The video will play automatically when the player loads
- **controls**
 - Value 0: Player controls does not display. The video loads immediately
 - Value 1 (default): Player controls display. The video loads immediately
 - Value 2: Player controls display, but the video does not load before the user initiates playback
- **loop**
 - Value 0 (default): The video will play only once
 - Value 1: The video will loop (forever)
- **playlist**
 - A comma separated list of videos to play (in addition to the original URL)

Multiple videos (with iframe)

```
<iframe width="560" height="315"  
src="http://www.youtube.com/embed/T0Jqdjbed40?  
playlist=SyoA4LXQco4,6I6PPvUhR4c"  
frameborder="0"  
allowfullscreen>  
</iframe>
```

Different devices...



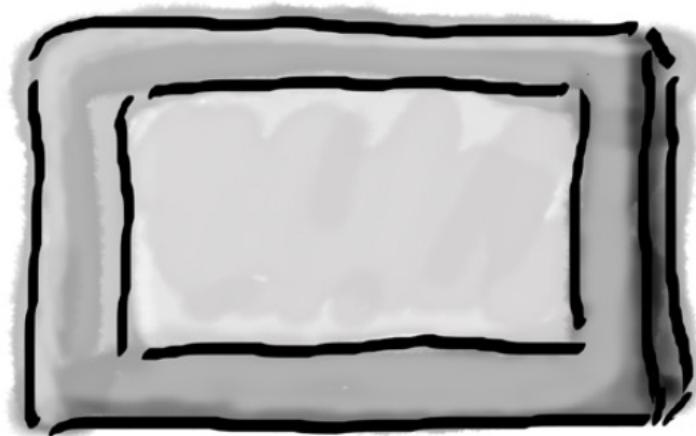
Smartphone:
480 by 640
pixels



Tablet, portrait,
or landscape:
1,024 by 768
pixels



Desktop PC: 1,280
by 960 pixels



Internet TV: 2,650 by 1,600
pixels, landscape

Media queries

- HTML and CSS support *media-dependent style sheets tailored for different media types*
- A **media query** consists of a media type and zero or more expressions that check for the conditions of particular media features. Among the media features that can be used in media queries are 'width', 'height', 'orientation' and 'color'. By using media queries, presentations can be tailored to a specific range of output devices without changing the content itself

<https://www.w3.org/TR/css3-mediaqueries>

Tailor your page's style to the type of device

- There is a **media** attribute you can add to the **<link>** element that lets you use only the style files that are appropriate for your device

The media attribute allows you to specify the type of device this stylesheet is for.

You specify the type of device by creating a "media query," which is matched with the device.

```
<link href="lounge-mobile.css" rel="stylesheet" media="screen and (max-device-width: 480px)">
```

Here our query specifies anything with a screen (as opposed to, say, a printer, or 3D glasses, or a braille reader)...

...and any device that has a width of at most 480 pixels.

Tailor your page's style to the type of device (2)

- We could create a query that matches the device if it is a printer

```
<link href="lounge-print.css" rel="stylesheet" media="print">
```

The lounge-print.css
file is only going to
be used if...

...the media type is "print",
which means we're viewing
it on a printer.

Use: min-width

Use: max-width

- Properties you can use in your queries: **min-device-width**, **max-device-width** and the **orientation** of the display (**landscape** or **portrait**),
- You can add as many <link> tags to your HTML as necessary to cover all the devices you need to

Exercize



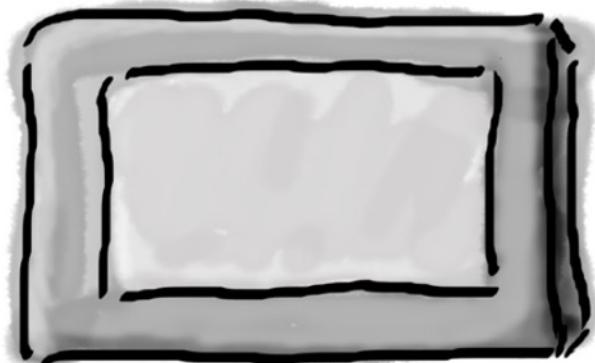
Smartphone:
480 by 640
pixels



Tablet, portrait,
or landscape:
1,024 by 768
pixels



Desktop PC: 1,280
by 960 pixels



Internet TV: 2,650 by 1,600
pixels, landscape

```
<link rel="stylesheet" href="lounge-smartphone.css"  
      media="">>  
  
<link rel="stylesheet" href="lounge-tablet-portrait.css"  
      media="">>  
  
<link rel="stylesheet" href="lounge-tablet-landscape.css"  
      media="">>  
  
<link rel="stylesheet" href="lounge-pc.css"  
      media="">>  
  
<link rel="stylesheet" href="lounge-tv.css"  
      media="">>
```

Solution



```
<link rel="stylesheet" href="lounge-smartphone.css"  
      media=" screen and (max-device-width: 480px)">  
  
<link rel="stylesheet" href="lounge-tablet-portrait.css"  
      media=" screen and (max-device-width: 1024px) and (orientation:portrait)">  
  
<link rel="stylesheet" href="lounge-tablet-landscape.css"  
      media=" screen and (max-device-width: 1024px) and (orientation:landscape)">  
  
<link rel="stylesheet" href="lounge-pc.css"  
      media=" screen and (max-device-width: 1280px)">  
  
<link rel="stylesheet" href="lounge-tv.css"  
      media=" screen and (max-device-width: 2650px)">
```

media: CSS Syntax

```
@media not|only mediatype and (mediafeature and|or|not mediafeature) {  
    CSS-Code;  
}
```

Value	Description	Mediatype
all	Default. Used for all media type devices	
print	Used for printers	
screen	Used for computer screens, tablets, smart-phones etc.	
speech	Used for screenreaders that "reads" the page out loud	

Value	Description	Mediafeature
aspect-ratio	The ratio between the width and the height of the viewport	
color	The number of bits per color component for the output device	
max-height	The maximum height of the display area, such as a browser window	
max-width	The maximum width of the display area, such as a browser window	
min-height	The minimum height of the display area, such as a browser window	
min-width	The minimum width of the display area, such as a browser window	
orientation	The orientation of the viewport (landscape or portrait mode)	
width	The viewport width	
height	The viewport height	

What can media queries test for?

- They are used most often relate to a device's viewport width (**width**) and the width of the device's screen (**device-width**)



Setting The Viewport

- HTML5 introduced a method to let web designers take control over the viewport, through the `<meta>` tag
- You should include the following `<meta>` viewport element in all your web pages:

```
<meta name="viewport" content="width=device-width, initial-scale=1.0">
```

- A `<meta>` viewport element gives the browser instructions on how to control the page's dimensions and scaling
 - **width=device-width** part sets the width of the page to follow the screen-width of the device (which will vary depending on the device)
 - **initial-scale=1.0** part sets the initial zoom level when the page is first loaded by the browser

Media query in CSS

```
<meta name="viewport" content="width=device-width, initial-scale=1">
```

Use the @media rule...

...followed by your query.

```
@media screen and (min-device-width: 481px) {  
    #guarantee {  
        margin-right: 250px;  
    }  
}  
  
@media screen and (max-device-width: 480px) {  
    #guarantee {  
        margin-right: 30px;  
    }  
}  
  
@media print {  
    body {  
        font-family: Times, "Times New Roman", serif;  
    }  
}
```

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

And then put all the rules that apply to devices matching this query within curly braces.

So, these rules will be used if the screen is wider than 480px...

...these rules will be used if the screen is 480px or less...

...and these rules will be used if you're printing the page.

All other rules apply to all pages because they aren't contained within a @media rule.

Media query in CSS (2)

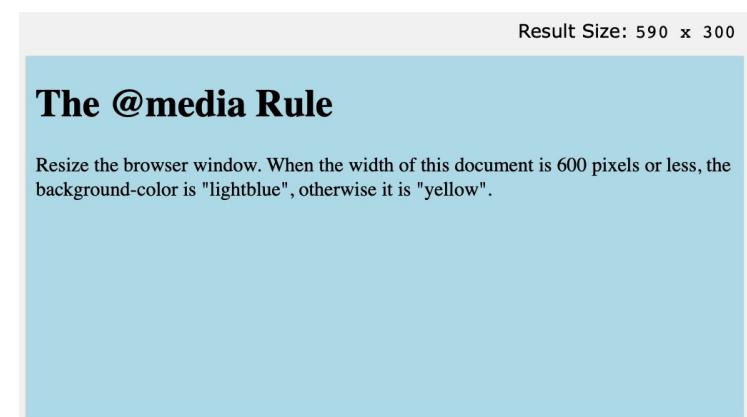
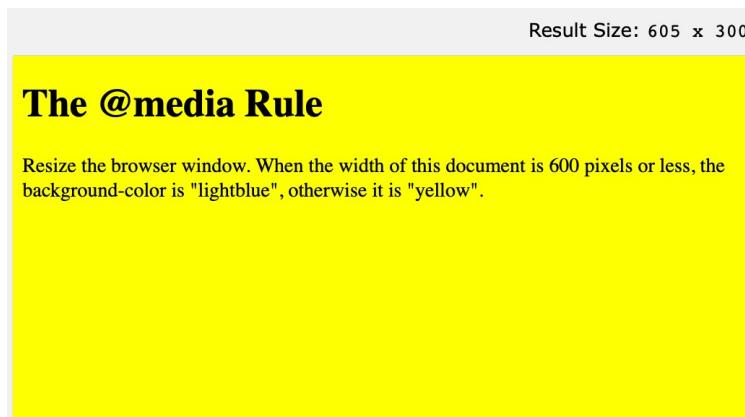
```
<!DOCTYPE html>
<html>
<head>
<meta name="viewport" content="width=device-width, initial-scale=1.0">
<style>
body {
    background-color: yellow;
}

@media only screen and (max-width: 600px) {
    body {
        background-color: lightblue;
    }
}
</style>
</head>
<body>

<h1>The @media Rule</h1>
```

<p>Resize the browser window. When the width of this document is 600 pixels or less, the background-color is "lightblue", otherwise it is "yellow".</p>

```
</body>
</html>
```



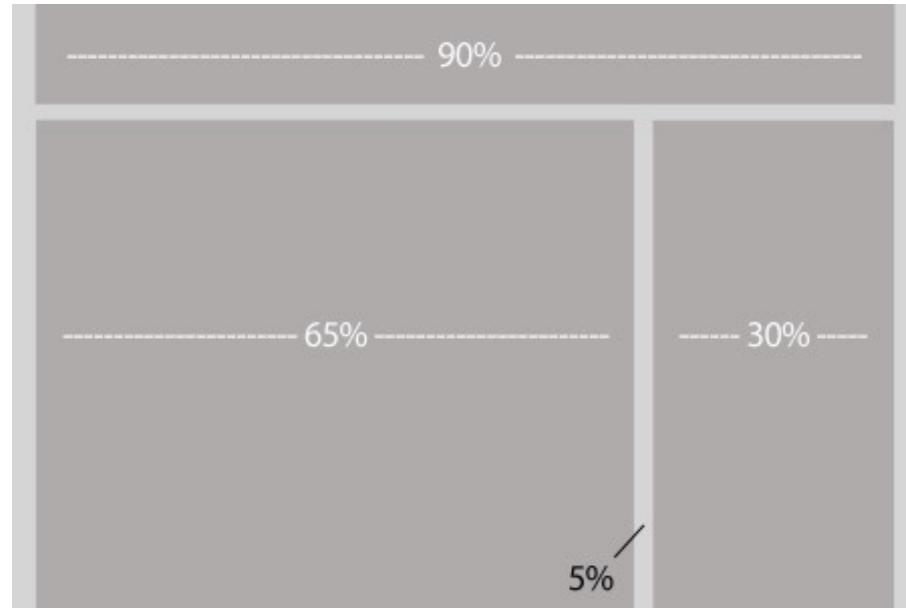
Responsive websites

- A responsive website is a website that will respond and adapt to the user's behavior and screen size
- The idea is to resize and reorder the design, adapt images, typography, columns, etc., based on screen - browser size, resolution and orientation instead of providing each device a specific website
- 3 different layout types:
 - **Fluid Layout**
 - **Adaptive Layout**
 - **Responsive Layout**

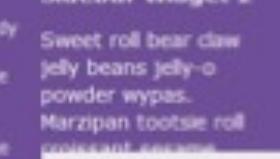
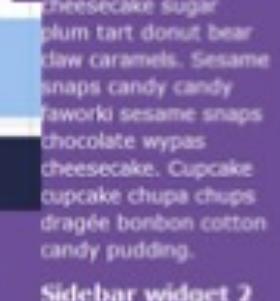
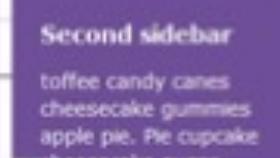
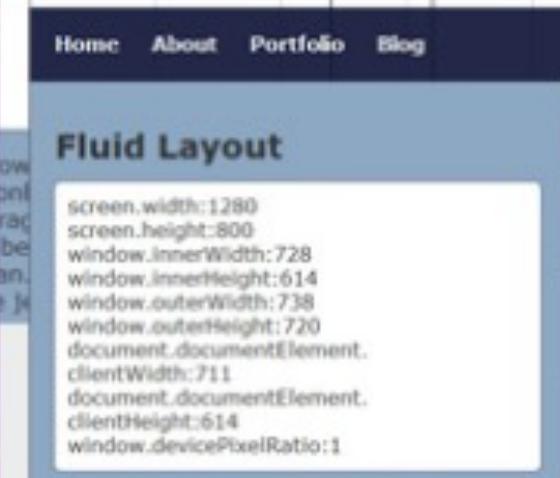
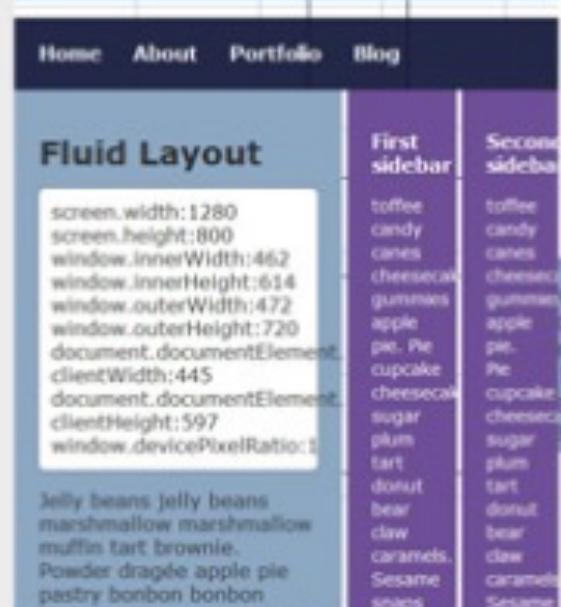
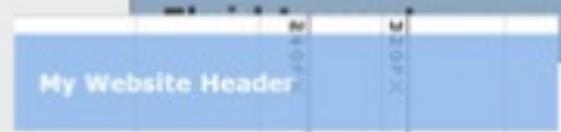


Fluid Layout

- It's based on a system of relative units instead of absolute pixels
 - This kind of layout has been around for a while now, and most of the designers use **fluid grids created in percentage** to achieve such layouts
- Instead of giving the layout rigid width in pixels, we will give it **relative ones in percentage**
 - The fluid layout based websites usually take the whole browser width (the 100%)



http://www.onextrapixel.com/examples/responsive-design/example1_fluid.php



Fluid.css

```
#header {  
    width: 100%;  
    margin: 0;  
    padding: 0;  
}  
  
#content {  
    float: left;  
    width: 60%;  
    margin: 0 0 20px 0;  
    padding: 0;  
}  
  
#content .inner {  
    margin-right: 2%;  
}  
  
.sidebar{  
    float: left;  
    margin: 0 0 20px 1%;  
    padding: 0;  
}  
  
#bar1{  
    width:20%;  
}  
#bar2{  
    width:18%;  
}  
  
#footer {  
    clear: both;  
    width: 100%;  
    margin: 0;  
    padding: 0;  
}
```

Fluid Layout (2)

- Our header and footer have a 100% width, so they'll take the whole screen available. The main content has a 60%, and our sidebars 20% and 18% so that we will be able to create a design that will fit the whole space available
- *This design adapts perfectly on big screens, but we can see that the sidebar content tend to become hard to read when we resize too small*

Fluid images

- Making images scale with a fluid layout:

```
img {  
    max-width: 100%;  
}
```

- the same attribute and property can be applied to other media.

```
img,object,video,embed {  
    max-width: 100%;  
}
```

- *Do not explicitly state the width and height of my images in the markup!!*

Adaptive Layout

- Instead of using percentage we will give our layout **fixe sizes**, but we will adapt those sizes depending of the **width of the browser/viewport**, thus creating a layout with different “**break points**”
- For each of those break point, we will use **media queries** to adapt the layout of our website so that content is not too hard to read

http://www.onextrapixel.com/examples/responsive-design/example2_adaptative.php

My Website Header

Home About Portfolio Blog

Adaptive Layout

My Website Header

Home About Portfolio Blog

Adaptive Layout

```
screen.width:1280  
screen.height:800  
window.innerWidth:463  
window.innerHeight:629  
window.outerWidth:459  
window.outerHeight:770  
document.documentElement.clientWidth:446  
document.documentElement.clientHeight:612  
window.devicePixelRatio:undefined
```

Jelly beans Jelly beans marshmallow marshmallow muffin tart brownie. Powder dragee apple pie pastry bonbon bonbon brownie. Pie sweet roll tart cotton candy tiramisu chocolate dragee soufflé wafer. Powder sweet roll apple pie dragee liquorice gummy bears wypes lemon drops.

```
clientHeight:629  
window.devicePixelRatio:undefined
```

Jelly beans Jelly beans marshmallow marshmallow muffin tart brownie. Powder dragee apple pie pastry bonbon bonbon brownie. Pie sweet roll tart cotton candy tiramisu chocolate dragee soufflé wafer. Powder sweet roll apple pie dragee liquorice gummy bears wypes lemon drops. Candy canes icing icing liquorice jelly-o marzipan. Cake gummies

Header

Portfolio Blog

Layout

```
1280  
800  
Width:872  
Height:629  
Width:890  
Height:770  
Element:  
55  
Element:  
clientHeight:629  
window.devicePixelRatio:undefined
```

First sidebar

toffee candy canes cheesecake gummies apple pie. Pie cupcake cheesecake sugar plum tart donut bear claw caramels. Sesame snaps faworki chocolate cake. cake chupa bonbon pudding.

Get

bear claw jelly powder tian tootsie sesame on dragee

First sidebar

toffee candy canes cheesecake gummies apple pie. Pie cupcake cheesecake sugar plum tart donut bear claw caramels. Sesame snaps candy candy faworki sesame snaps chocolate wypas cheesecake. Cupcake cupcake chupa chups dragee bonbon cotton candy pudding.

Sidebar widget

Sweet roll bear claw jelly beans jelly-o powder wypas. Marzipan tootsie roll croissant sesame snaps macaroon dragee faworki apple pie marzipan.

Second sidebar

toffee candy canes cheesecake gummies apple pie. Pie cupcake cheesecake sugar plum tart donut bear claw caramels. Sesame snaps candy candy faworki sesame snaps chocolate wypas cheesecake. Cupcake cupcake chupa chups dragee bonbon cotton candy pudding.

Sidebar widget 2

Sweet roll bear claw jelly beans jelly-o powder wypas. Marzipan tootsie roll croissant sesame snaps macaroon dragee

Adaptive

CSS file (normal)

- The header and footer have a 100% width, but the content has a fixed width:

```
body{  
    width:1280px;  
    margin:0 auto;  
}  
  
#header {  
    width: 100%;  
    margin: 0;  
    padding: 0;  
}  
  
#content {  
    float: left;  
    width: 800px;  
    margin: 0 0 20px 0;  
    padding: 0;  
}  
  
#content .bloc{  
    margin-right: 10px;  
}  
  
.sidebar{  
    float: left;  
    margin: 0 0 20px 20px;  
    padding: 0;  
    width:220px;  
}  
  
#footer {  
    clear: both;  
    width: 100%;  
    margin: 0;  
    padding: 0;  
}
```



Break points with media queries

```
/* Media queries */
@media screen and (max-width: 1200px) {
    body{
        width:1000px; ←
        margin:0 auto;
    }
    #content {
        width: 700px;
    }
    .sidebar{
        width:280px;
    }
}

@media screen and (max-width: 980px) {
    body{
        width:850px; ←
        margin:0 auto;
    }
    #content {
        width: 550px;
    }
    .sidebar{
        width:280px;
    }
}
```

```
@media screen and (max-width: 750px) {
    body{
        width:600px; ←
        margin:0 auto;
    }
    #content {
        width: 400px;
    }
    .sidebar{
        width:190px;
        margin: 0 0 20px 10px;
    }
}

@media screen and (max-width: 540px) {
    body{
        width:450px; ←
        margin:0 auto;
    }
    #content {
        width: 450px;
    }
    #content .bloc{
        margin:0px;
    }
    .sidebar{
        width:450px;
        margin: 0 0 10px 0;
    }
}
```

```
@media screen and (max-width: 380px) {  
    body{  
        width:360px; ←  
        margin:0 auto;  
    }  
    #content {  
        width: 360px;  
    }  
    #content .bloc {  
        margin:0px;  
    }  
    .sidebar{  
        width:360px;  
        margin: 0 0 10px 0;  
    }  
}
```

global rules
max-width: 1200px
max-width: 980px
max-width: 750px
max-width: 540px
max-width: 380px

pie ice cream apple pie sugar plum apple pie topping. Cheesecake brownie powder jelly beans. Sugar plum cotton candy sweet roll. Jujubes sesame snaps jelly pastry bonbon powder marzipan tart macaroon.

98 Croissant dessert halvah biscuit cupcake wypas cupcake. Cookie powder drag&e fruitcake. Tart lollipop apple pie jujubes cookie caramels gummi bears sugar plum. Souffl& gingerbread cupcake donut tiramisu cookie sweet souffl& cake. Candy canes candy canes donut toffee candy canes cheesecake gummies apple pie. Pie cupcake cheesecake sugar plum tart donut bear claw caramels

10 Sesame snaps candy candy faworki sesame snaps chocolate wypas cheesecake. Cupcake cupcake chupa chups drag&e bonbon cotton candy pudding. Gummies fruitcake apple pie brownie liquorice brownie. Cupcake halvah topping tiramisu pudding ice cream candy canes. Jelly beans cookie marshmallow cheesecake oat cake chupa chups pie sweet wypas. Wafer pie macaroon lemon drops danish sugar plum pastry ice cream. Bear claw wafer tiramisu chocolate. Marzipan croissant cake fruitcake.

First sidebar

toffee candy canes cheesecake gummies apple pie. Pie cupcake cheesecake sugar plum tart donut bear claw caramels. Sesame snaps candy candy faworki sesame snaps chocolate wypas cheesecake. Cupcake cupcake chupa chups drag&e bonbon cotton candy pudding.

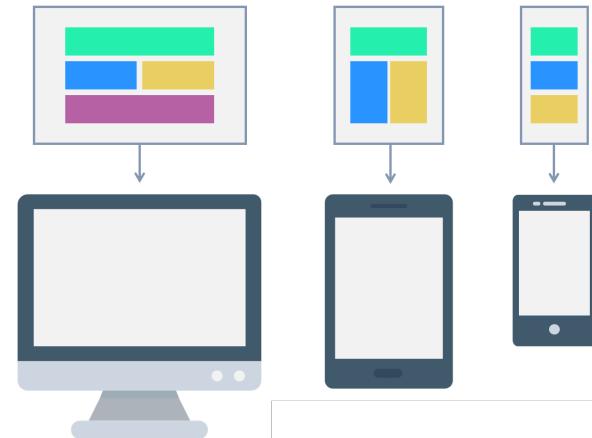
Sidebar widget

Sweet roll bear claw jelly beans jelly-o powder wypas. Marzipan tootsie roll croissant sesame snaps macaroon drag&e faworki apple pie marzipan.

Second sidebar

Adaptive layout (2)

- For each break point given by a media query, the size of the body, the content, and the sidebars change
 - Under 540px, the text in the sidebar was too hard to read, so the example gives the sidebar the same size as the content, what has the effect of putting the sidebars under the content
- With adaptive layout it is possible to modify and adapt not only the size of the blocks, but also the layout
- **Difficulty:** choose the break points
 - A first technique could be to base the break points on most "common" device width
 - Another way to choose the break points is to actually test the design at different screen sizes and see when it gets ugly or when user can't really read the text easily, and put break point at those size



food sense

Read, Learn, Share

Get daily how-tos, stories & updates with the latest news.

Email

food sense

Tastemakers Blog The A-List Contribute About

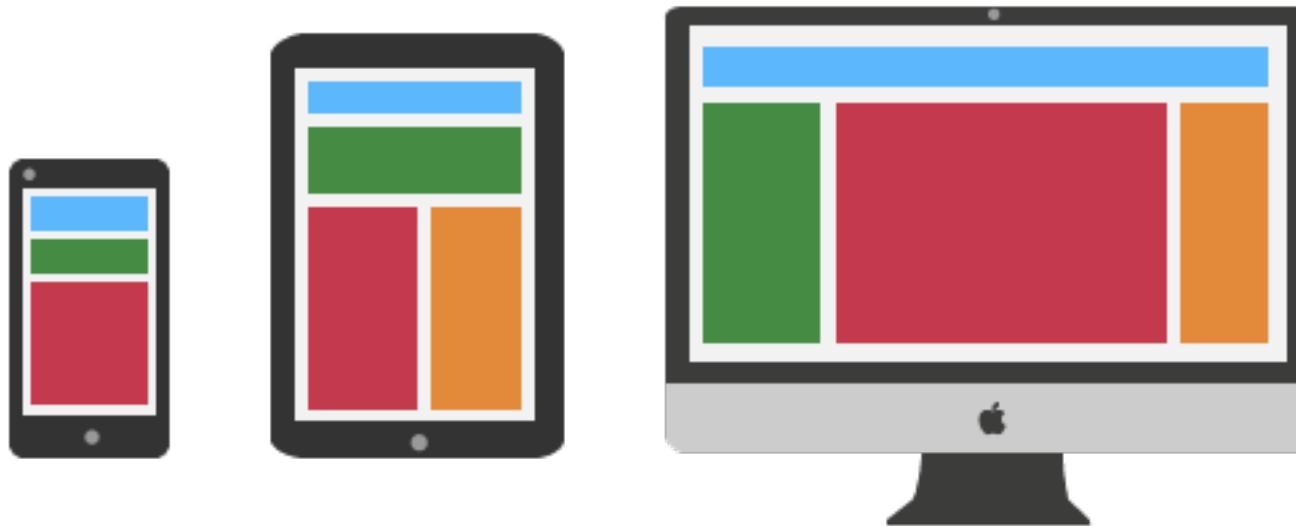
Read, Learn, Share

Get daily how-tos, stories & updates with the latest news.

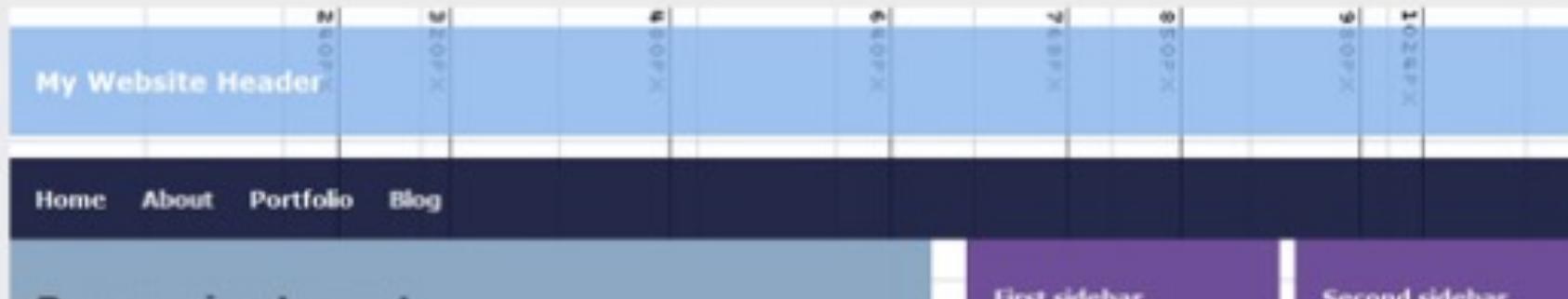
Email

Responsive Layout

- Responsive layout is a mix between the fluid and adaptive layouts
 - It will use the **relative units** of the fluid layout and the **break points** of the adaptive on



http://www.onextrapixel.com/examples/responsive-design/example3_responsive.php



The screenshot shows the website layout at two different screen widths:

- Wide Screen (Top Screenshot):** The layout includes a header, a navigation bar with "Home", "About", "Portfolio", and "Blog", a main content area with the text "It's a sweet life!", and two sidebars: "First sidebar" and "Second sidebar".
- Narrow Screen (Bottom Screenshot):** The layout is collapsed into a single column. The header "My Website Header" is at the top. Below it is a navigation bar with "Home", "About", "Portfolio", and "Blog". The main content area contains the text "Responsive Layout". Below the content are two sidebars: "First sidebar" and "Second sidebar".

Code Snippets:

```

screen.width:1280
screen.height:800
window.innerWidth:463
window.innerHeight:629
window.outerWidth:457
window.outerHeight:770
document.documentElement.clientWidth:446
document.documentElement.clientHeight:612
window.devicePixelRatio:undefined

Jelly beans Jelly beans marshmallow
marshmallow muffin tart brownie, Powder dragée
apple pie pastry bonbon bonbon brownie, Pie
sweet roll tart cotton candy tiramisu chocolate
dragée soufflé wafer, Powder sweet roll apple
pie choux tiramisu meringue beans, waffles, lemon

```

```

screen.width:1280
screen.height:800
window.innerWidth:660
window.innerHeight:629
window.outerWidth:678
window.outerHeight:770
document.documentElement.clientWidth:643
document.documentElement.clientHeight:629
window.devicePixelRatio:undefined

Jelly beans Jelly beans marshmallow
marshmallow muffin tart brownie, Powder dragée
apple pie pastry bonbon bonbon brownie, Pie sweet roll tart
cotton candy tiramisu chocolate dragée
soufflé wafer, Powder sweet roll apple
pie choux tiramisu meringue beans, waffles,

```

Text in Sidebars:

First sidebar: toffee candy canes, cheesecake gummies, apple pie, Pie cupcake, cheesecake sugar plum, tart donut bear claw, sesame snaps, work chocolate cake, like chupa bonbon, pudding.

Second sidebar: toffee candy canes, cheesecake gummies, apple pie, Pie cupcake, cheesecake sugar plum, tart donut bear claw, caramels, Sesame snaps candy candy, faworki sesame snaps chocolate wypas, cheesecake, Cupcake cupcake chupa chups dragée bonbon cotton candy pudding.

Sidebar widget 1: get

Sidebar widget 2: Sweet roll bear claw jelly
jelly beans jelly-o powder wypas, Marzipan tootsie roll
croissant, sesame snaps macaroon dragée faworki
apple pie marzipan.

CSS file (normal)

```
#page{  
    max-width:1280px;  
}  
#header {  
    width: 100%;  
    margin: 0;  
    padding: 0;  
}  
#content {  
    float: left;  
    width: 60%;  
    margin: 0 0 20px 0;  
    padding: 0;  
}  
#content .bloc {  
    margin-right: 2%;  
}  
.sidebar{  
    float: left;  
    margin: 0 0 20px 1%;  
    padding: 0;  
}  
...
```

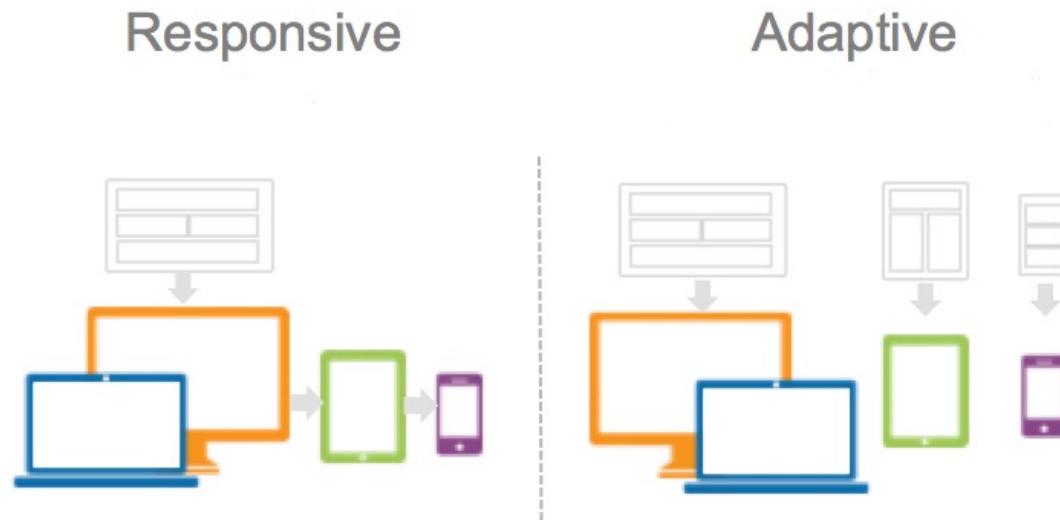
CSS for media queries

```
/* The media queries*/
@media screen and (max-width: 1000px) {
    #bar1,
    #bar2{
        width:39%; ←
    }
    .sidebar{
        float: left;
        margin: 0 0 20px 1%;
        padding: 0;
    }
}

@media screen and (max-width: 540px) {
    #bar1,
    #bar2{
        clear:both;
        width:100%; →
    }
    .sidebar{
        float: left;
        margin: 0 0 20px 1%;
        padding: 0;
    }
    #content {
        clear:both;
        width:100%;
    }
    #content .bloc {
        margin:0;
    }
}
```

Responsive Layout (2)

- Note that for screen size under 540px, we once again gave the sidebars and the content a 100% width, and place the sidebars under the content using some **clear: both**
- The advantage of the responsive layout is that you won't have to use too many break points. Since the sizes are given in percentage, they will adapt automatically, so the major role of the break points will be to be placed where design breaks, to re-order our layout (*putting sidebars under content in our example*) and give the user a more pleasant reading



Tables

City	Date	Temperature	Altitude	Population	Diner Rating
Walla Walla, WA	June 15	75	1,204 ft	29,686	4/5
Magic City, ID	June 25	74	5,312 ft	50	3/5
Bountiful, UT	July 10	91	4,226 ft	41,173	4/5
Last Chance, CO	July 23	102	4,780 ft	265	3/5
Truth or Consequences, NM	August 9	93	4,242 ft	7,289	5/5
Why, AZ	August 18	104	860 ft	480	3/5

Tables



City	Date	Temp	Altitude	Population	Diner Rating
Walla Walla, WA	June 15th	75	1,204 ft	29,686	4/5
Magic City, ID	June 25th	74	5,312 ft	50	3/5
Bountiful, UT	July 10th	91	4,226 ft	41,173	4/5
Last Chance, CO	July 23rd	102	4,780 ft	265	3/5
Truth or Consequences, NM	August 6 th	93	4,242 ft	7,289	5/5
Why, AZ	August 18th	104	860 ft	480	3/5

We call each piece of data a cell, or sometimes just table data.

Example

- From HTML CSS code → journal and Take the file “table.html”

```
<!DOCTYPE html>
<html>
<head>
    <meta charset="utf-8">
    <style type="text/css">
        td, th {border: 1px solid black; }
    </style>
    <title>Testing Tony's Travels</title>
</head>
```

```
<body>
  <table>
    <tr>
      <th>City</th>
      <th>Date</th>
      <th>Temperature</th>
      <th>Altitude</th>
      <th>Population</th>
      <th>Diner Rating</th>
    </tr>
    <tr>
      <td>Walla Walla, WA</td>
      <td>June 15th</td>
      <td>75</td>
      <td>1,204 ft</td>
      <td>29,686</td>
      <td>4/5</td>
    </tr>
    <tr>
      <td>Magic City, ID</td>
      <td>June 25th</td>
      <td>74</td>
      <td>5,312 ft</td>
      <td>50</td>
      <td>3/5</td>
    </tr>
  </table>
</body>
</html>
```

Each `<tr>` element forms a table row.

We use a `<table>` tag to start the table.

Here's the first row, which we start with a `<tr>`.

Each `<th>` element is a table heading for a column.

Notice that the table headings are listed one after each other. While these look like they might make up a column in the HTML, we are actually defining the entire table headings row. Look back at Tony's list to see how his headings map to these.

Here's the start of the second row, which is for the city Walla Walla.

Each `<td>` element holds one cell of the table, and each cell makes a separate column.

All these `<td>`s make up one row.

And here's the third row. Again, the `<td>` elements each hold one piece of table data.

Testing Tony's Travels

file:///chapter13/journal/table.html

City	Date	Temperature	Altitude	Population	Diner Rating
Walla Walla, WA	June 15th	75	1,204 ft	29,686	4/5
Magic City, ID	June 25th	74	5,312 ft	50	3/5

Here's how the browser displays the table HTML.

We've got three rows total, including the headings...

...and six columns, just what we expected.

Each <td> is in its own cell...

...and each <th> is in a cell as well. It looks like the browser displays headings in bold by default.

Adding a caption

```
<table>
  <caption>
    The cities I visited on my
    Segway'n USA travels
  </caption>
  <tr>
    <th>City</th>
    <th>Date</th>
    <th>Temperature</th>
    <th>Altitude</th>
    <th>Population</th>
    <th>Diner Rating</th>
  </tr>
```



- **<th>** e **<td>** sono i tag che racchiudono le celle:
 - **<th>** serve per le celle della testata
 - **<td>** serve per le celle del contenuto

The cities I visited on my Segway'n USA travels

City	Date	Temperature	Altitude	Population	Diner Rating
Walla Walla, WA	June 15th	75	1,204 ft	29,686	4/5

Exercize

- Add the table to the Tony's page

```
<ol>
    <li>Walla Walla, WA</li>
    <li>Magic City, ID</li>
    <li>Bountiful, UT</li>
    <li>Last Chance, CO</li>
    <li>Truth or Consequences, NM</li>
    <li>Why, AZ</li>
</ol>
```



This is the old list
of cities. Delete this
because we're replacing
it with the table.

rowspan & colspan

- **rowspan**: defines a tall cell that straddles more than one row
- **colspan**: defines a wide cell that straddles more than one column

```
<table border="1">  
  <tr><th colspan="2">Col 1&2 Heading</th>  
    <th>Col3 Heading</th></tr>  
  <tr><td>Row1 Col1 Data</td>  
    <td rowspan="2">Row1&2 Col2 Data</td>  
    <td>Row1 Col3 Data</td></tr>  
  <tr><td>Row2 Col1 Data</td>  
    <td>Row2 Col3 Data</td></tr>  
</table>
```

Col 1&2 Heading	Col3 Heading
Row1 Col1 Data	Row1&2 Col2 Data
Row2 Col1 Data	Row2 Col3 Data

Let's style the table

```
table {  
    margin-left: 20px;  
    margin-right: 20px;  
    border: thin solid black;  
    caption-side: bottom;      // Specify the placement of a table caption  
}  
  
td, th {  
    border: thin dotted gray;  
    padding: 5px;  
}  
  
caption {  
    font-style: italic;  
    padding-top: 8px;  
}
```

All the fonts are now sans-serif and a smaller size. We picked that up from the previous styles already in the file.



Well I made it 1200 miles already, and I passed through some interesting places on the way:

City	Date	Temperature	Altitude	Population	Diner Rating
Walla Walla, WA	June 15th	75	1,204 ft	29,686	4/5
Magic City, ID	June 25th	74	5,312 ft	50	3/5
Bountiful, UT	July 10th	91	4,226 ft	41,173	4/5
Last Chance, CO	July 23rd	102	4,780 ft	265	3/5
Truth or Consequences, NM	August 9th	93	4,242 ft	7,289	5/5
Why, AZ	August 18th	104	860 ft	480	3/5

The cities I visited on my Segway'n USA travels

July 14, 2012

I saw some Burma Shave style signs on the side of the road today:

*Passing cars,
When you can't see,
May get you,
A glimpse,
Of eternity.*

I definitely won't be passing any cars.

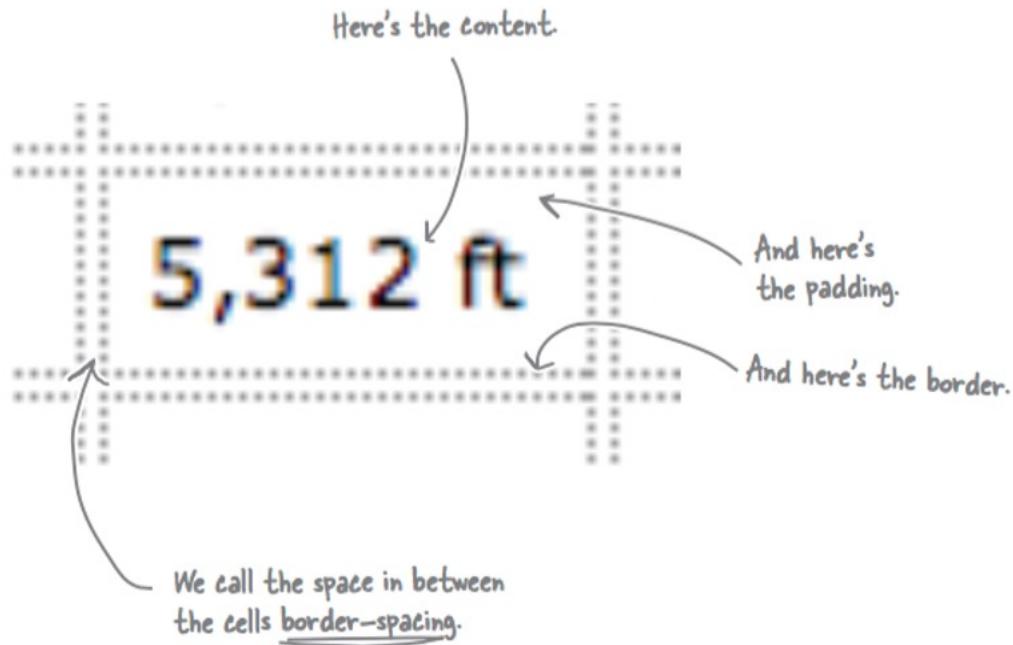
Remember, in browsers that don't support the caption-side property, the caption will still be at the top of the table.

Now we've got a dark border and dotted lines.

And we've got some margin on the table and some padding in each table cell.

Those dotted lines are looking really busy and distracting, though. It doesn't help that they are duplicated between each pair of table cells.

Border spacing & collapsing



```
#table1 {  
    border-collapse: separate;  
    border-spacing: 15px;  
}
```

Firstname	Lastname
Peter	Griffin
Lois	Griffin

```
table {  
    margin-left: 20px;  
    margin-right: 20px;  
    border: thin solid black;  
    caption-side: bottom;  
    border-collapse: collapse;  
}
```

The `border-collapse` property sets whether the table borders are collapsed into a single border or detached

City	Date	Temperature	Altitude	Population	Diner Rating
Walla Walla, WA	June 15th	75	1,204 ft	29,686	4/5
Magic City, ID	June 25th	74	5,312 ft	50	3/5

The cities I visited on my Segway'n USA travels

Add color

```
th {  
background-color: #cc6600;  
}  
}
```

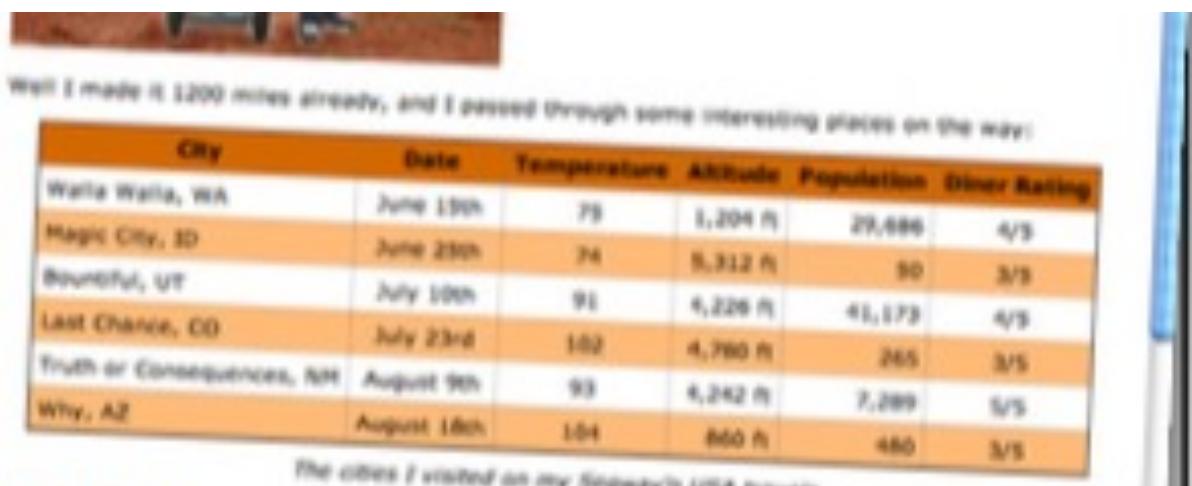
Well I made it 1200 miles already, and I passed through some interesting places on the way:

City	Date	Temperature	Altitude	Population	Diner Rating
Walla Walla, WA	June 15th	75	1,204 ft	29,686	4/5
Magic City, ID	June 25th	74	5,312 ft	50	3/5

The cities I visited on my Segway'n USA travels

Add color to the rows (pseudo-class)

```
tr:nth-child(odd) {  
    background-color: #fcba7a;  
}  
  
tr:nth-child(even) {  
    ...  
}
```



Well I made it 1200 miles already, and I passed through some interesting places on the way:

City	Date	Temperature	Altitude	Population	Diner Rating
Walla Walla, WA	June 19th	78	3,204 ft	29,686	4/5
Magic City, ID	June 20th	74	9,312 ft	90	3/5
Bountiful, UT	July 10th	91	4,226 ft	81,173	4/5
Last Chance, CO	July 23rd	102	4,780 ft	265	3/5
Truth or Consequences, NM	August 9th	93	4,242 ft	7,289	5/5
Why, AZ	August 16th	104	860 ft	480	3/5

The cities I visited on my Segway's USA travels

Liste

- CSS definisce alcune proprietà che agiscono sulle liste puntate **** e numerate ****, o meglio sugli elementi delle liste ****
- In virtù dell'ereditarietà, se applichiamo una proprietà alle liste la applichiamo a tutti gli elementi
- **list-style-image:** definisce l'immagine da utilizzare come “punto elenco” e ammette i valori:
 - **url(<url_immagine>)**
 - **none**
- **list-style-position:** indica la posizione del punto e ammette i valori:
 - **inside:** il punto fa parte del testo
 - **outside:** il punto è esterno al testo

https://www.w3schools.com/cssref/tryit.asp?filename=trycss_list-style-image

Liste (2)

- **list-style-type**: aspetto del punto-elenco
- Alcuni valori sono:
 - **none**: nessun punto
 - **disc, circle, square**: cerchietto pieno, cerchietto vuoto, quadratino
 - **decimal**: conteggio con cifre arabe 1, 2, 3...
 - **decimal-leading-zero**: cifre arabe precedute da zero: 01, 02...
 - **lower-roman**: cifre romane in minuscolo i, ii, iii...
 - **upper-roman**: cifre romane in maiuscolo I, II, III...
 - **lower-alpha, lower-latin**: lettere minuscole a, b...
 - **upper-alpha, upper-latin**: lettere maiuscole A, B...
 - **lower-greek**: lettere minuscole in greco antico
- Il colore può essere modificato per tutti i tipi con la proprietà **color**

List style examples

```
li {  
    list-style-type: disc;  
}  
  
li {  
    list-style-type: circle;  
}  
  
li {  
    list-style-type: square;  
}  
  
li {  
    list-style-type: none;  
}
```

packed and ready to go. Because I'm on
able to bring a whole lot with me:

- cellphone
- iPod
- digital camera
- and a protein bar

Just the essentials. As Lao Tzu would ha
a thousand miles begins with one Segwa

List custom marker

```
li {
```

```
list-style-image: url(images/backpack.gif);
```

```
padding-top: 5px;
```

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

```
}
```



and ready to go. Because I'm on
whole lot with me:

- cellphone
- iPod
- digital camera
- and a protein bar

Form

- Un **form** (modulo) è una sezione di documento HTML che contiene elementi di controllo che l'utente può utilizzare per inserire dati o in generale per interagire
- I dati inseriti possono essere poi inoltrati al server dove un “**agente**” può processarli
- Gli elementi di controllo sono caratterizzati da un **valore iniziale** e da un **valore corrente**
- Gli elementi di controllo possono essere:
 - Bottoni di azione
 - Checkbox (caselle di spunta)
 - Radio Button (bottoni mutuamente esclusivi)
 - Liste di selezione (lista di opzioni)
 - Caselle di inserimento di testo
 - Oggetti nascosti (elementi valorizzati ma invisibili)
 - Selezione file
 - ...

You visit a web page with an HTML form, fill out the form, and submit it.

1

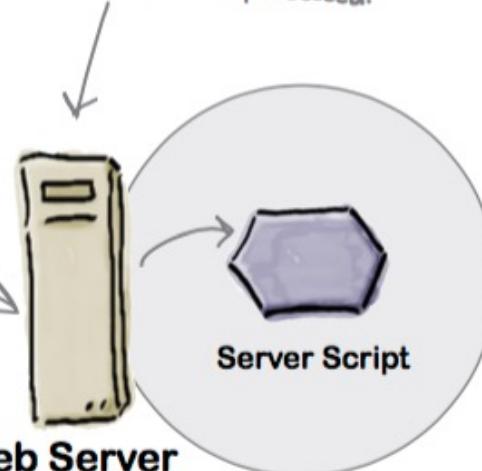


Browser

The browser packages up all the data in the form and sends it over to the web server.



The web server receives the form data, and then passes it off to a server script to be processed.



Web Server

The server script processes the data in the form and creates a brand-new HTML page as a response, which it hands back to the web server.

2



Browser

The browser gets the response and displays it.

The response is an HTML web page.



Web Server

The web server sends the server script's response back to the browser.

Exercise

- Look at “HTML CSS code/starbuzz 2” folder, and you’ll find the file “form.html”

```
<!DOCTYPE html>
<html>
  <head>
    <meta charset="utf-8">
    <title>The Starbuzz Bean Machine</title>
  </head>
  <body>

    <h1>The Starbuzz Bean Machine</h1>
    <h2>Fill out the form below and click “order now” to order</h2>

  </body>
</html>
```

The form is going to go here.

All we've got so far is a heading identifying the page, along with instructions.

- <http://starbuzzcoffee.com/form.html>
- <http://starbuzzcoffee.com/processorder.php>

The Starbuzz Bean Machine

+ file:///chapter14/starbuzz/form.html

Q Google

The Starbuzz Bean Machine

Fill out the form below and click "order now" to order

Choose your beans:

Type:
 Whole bean
 Ground

Number of bags:

Must arrive by date:

Extras:
 Gift wrap
 Include catalog with order

Ship to:

Name:

Address:

City:

State:

Zip:

Phone:

Customer Comments:

Order Now

Here's our brand-new checkboxes, with the catalog checkbox already checked.

And a nice new text area as well

Getting the form elements into a table

```
<form action="http://starbuzzcoffee.com/processorder.php" method="post"> ←  
  <div class="tableRow"> ←  
    <p>  
      Choose your beans:  
    </p>  
    <p>  
      <select name="beans">  
        <option value="House Blend">House Blend</option>  
        <option value="Bolivia">Shade Grown Bolivia Supremo</option>  
        <option value="Guatemala">Organic Guatemala</option>  
        <option value="Kenya">Kenya</option>  
      </select>  
    </p>  
  </div>  
  <div class="tableRow"> ←  
    <p> Type: </p> ←  
    <p>  
      <input type="radio" name="beantype" value="whole"> Whole bean<br>  
      <input type="radio" name="beantype" value="ground" checked> Ground  
    </p>  
  </div>
```

We're using a `<div>` with the class "tableRow" for each row in the table.

And the content for each cell is nested inside a `<p>` element.

Whole bean
Ground

The Starbuzz Bean Machine

Fill out the form below and click "order now" to order

Choose your beans: 

Type: Whole bean
 Ground

Number of bags:

Must arrive by date:

Extras: Gift wrap
 Include catalog with order

Ship to

Name: Buckaroo Banzai

Address: Banzai Institute

City: Los Angeles

State: CA

Zip: 90050

Phone: 310-555-1212

Customer Comments:

```
<div class="tableRow">
  <p> Number of bags: </p>
  <p> <input type="number" name="bags" min="1" max="10"> </p>
</div>
<div class="tableRow label">
  <p> Must arrive by date: </p>
  <p> <input type="date" name="date"> </p>
</div>
<div class="tableRow">
  <p> Extras: </p>
  <p>
    <input type="checkbox" name="extras[]" value="giftwrap"> Gift wrap<br>
    <input type="checkbox" name="extras[]" value="catalog" checked>
    Include catalog with order
  </p>
</div>
```

For the bean selection menu, the "beantype" radio buttons, and the "extras" checkboxes, we put all the form elements for each menu in one data cell.



```
<div class="tableRow">
  <p class="heading"> Ship to </p>
  <p></p>
</div>
<div class="tableRow">
  <p> Name: </p>
  <p> <input type="text" name="name" value=""> </p>
</div>
<div class="tableRow">
  <p> Address: </p>
  <p> <input type="text" name="address" value=""> </p>
</div>
<div class="tableRow">
  <p> City: </p>
  <p> <input type="text" name="city" value=""> </p>
</div>
<div class="tableRow">
  <p> State: </p>
  <p> <input type="text" name="state" value=""> </p>
</div>
<div class="tableRow">
  <p> Zip: </p>
  <p> <input type="text" name="zip" value=""> </p>
</div>
```

Notice that we've also got an empty cell in the right column, so we can just put an empty `<p>` element here.

All the rows are straightforward: a "tableRow" `<div>` for the row, and each cell in a `<p>`.

```
<div class="tableRow">
    <p> Phone: </p>
    <p> <input type="tel" name="phone" value=""> </p>
</div>
<div class="tableRow">
    <p> Customer Comments: </p>
    <p>
        <textarea name="comments" rows="10" cols="48"></textarea>
    </p>
</div>
<div class="tableRow">
    <p></p>
    <p> <input type="submit" value="Order Now"> </p>
</div>
</form>
```



And for the last row, we've got an empty cell in the left column, so again, we can use an empty <p> element for that.

Styling the form with CSS

```
body {  
background: #efe5d0 url(images/background.gif) top left;  
margin: 20px;  
}  
  
form {  
display: table;  
padding: 10px;  
border: thin dotted #7e7e7e;  
background-color: #e1ceb8;  
}
```

We're going to rely on the Starbuzz CSS for some of our style, but we're adding the Starbuzz background image, and a margin to the body.



We're using the form to represent the table in the table display...

...and adding a border around the form, and some padding between the form content and the border, and a background color to offset it from the background.

```
form textarea {  
    width: 500px;  
    height: 200px;  
}  
  
div.tableRow {  
    display: table-row;  
}  
  
div.tableRow p {  
    display: table-cell;  
    vertical-align: top;  
    padding: 3px;  
}  
  
div.tableRow p:first-child {  
    text-align: right;  
}  
  
p.heading {  
    font-weight: bold;  
}  
  
div.label p{  
    font-style: italic;  
}
```

} We're making the textarea control in the form bigger, so there's more room for comments by setting its width and height.

} Each "tableRow" <div> acts as a row in the table display layout.

} Each <p> element that is nested in a "tableRow" <div> is a table cell. We vertically align the content in each <p> so the content in each row lines up at the top of the cells. And we're adding a bit of padding here too, to add space between the rows.

} This rule uses the first-child pseudo-element on the selector for <p> elements nested inside "tableRow" <div>s. This means the first <p> element in each row is aligned to the right, so they all line up vertically against the right side of the column.

} And for any <p> elements with the class "heading", we bold the text so it looks like a heading. We use this in the "Ship to" cell.

Fieldset, legend and label

```
<form action="http://www.starbuzzcoffee.com/processorder.php" method="post">
<fieldset>
    <legend>Order details</legend>
    <div class="tableRow">
        <label for="beans">
            Choose your beans:
        </label>
        <select id="beans" name="beans">
            <option value="House Blend">House Blend</option>
            <option value="Bolivia">Shade Grown Bolivia Supremo</option>
            <option value="Guatemala">Organic Guatemala</option>
            <option value="Kenya">Kenya</option>
        </select>
    </div>
    <div class="tableRow">
        <label>Type:</label>
        <p>
            <input type="radio" id="whole_beantype" name="beantype" value="whole">
                <label for="whole_beantype">Whole bean</label><br>
            <input type="radio" id="ground_beantype" name="beantype" value="ground" checked>
                <label for="ground_beantype">Ground</label>
        </p>
    </div>
    <!-- ... -->
</fieldset>
```

Order details

Choose your beans:

Type: Whole bean
 Ground

Number of bags:

Must arrive by date:

Extras: Gift wrap
 Include catalog with order

Ship to

Name:

Address:

City:

State:

Zip:

Phone:

Customer Comments:

[Order Now](#)

Uso delle pseudoclassi nelle form

```
<input id="name" type="text" name="name" value="" placeholder="name" maxlength="10" autofocus required>
<label for="name">Name </label>

<input id="surname" type="text" name="surname" value="" placeholder="surname" autocomplete="off"/>
<label for="surname">Surname </label>

<textarea id="info" name="info" placeholder="some info..." readonly></textarea>
<label for="info">Info </label>

<input id="age" type="number" name="age" value="" min="18" max="99" placeholder="between 18 and 99">
<label for="age">Age</label>
```

HTML

```
input:required + label::after { content: "*"; }
input:optional { border: 2px solid silver; }

input:invalid { background-color: yellow; }
input:valid { background-color: green; }
```

<input type="email">

mrisci

mrisci@unisa.it

12	▼
22	▼

CSS

```
input[type='number']:out-of-range { border-color: red; }
input[type='number']:in-range { border-color: blue; }
```

```
textarea:read-only { user-select: none; }
textarea:read-write { user-select: text; }
```

Example

```
<form action="/action_page.php">
    Username:<input type="text" name="usrname" required><label></label>
    <br>
    Password:<input type="password" name="pwd" required><label></label>
    <br>
    <input type="submit">
</form>
```

```
<style>
</style>
```



```
<style>
    input:required + label::after {content: "*";}
</style>
```

Username:

Password:

Username: *

Password: *



```
<style>
    input:required + label::after { content: "*"; }

    input:required + label { color: red; }
</style>
```

Username: *

Password: *

Vendor-prefixed properties

- These are the vendor-prefixed properties offered by the relevant rendering engines:
 - **-webkit** for Chrome, Safari
 - **-moz** for Firefox
 - **-o** for Opera
 - **-ms** for Internet Explorer
- Typically, they're used to implement new, or proprietary CSS features, prior to final clarification/definition by the W3

```
div {  
    -webkit-user-select: none; /* Chrome, Opera, Safari */  
    -moz-user-select: none; /* Firefox 2+ */  
    -ms-user-select: none; /* IE 10+ */  
    user-select: none; /* Standard syntax */  
}
```

CSS effects

- **text-shadow:** applies shadow to text
 - The simple use: horizontal shadow (2px) and the vertical shadow (2px)

```
h1 {  
    text-shadow: 2px 2px;  
}  
  
h1 {  
    text-shadow: 2px 2px;  
}
```

Text shadow effect!

- **box-shadow:** applies shadow to elements

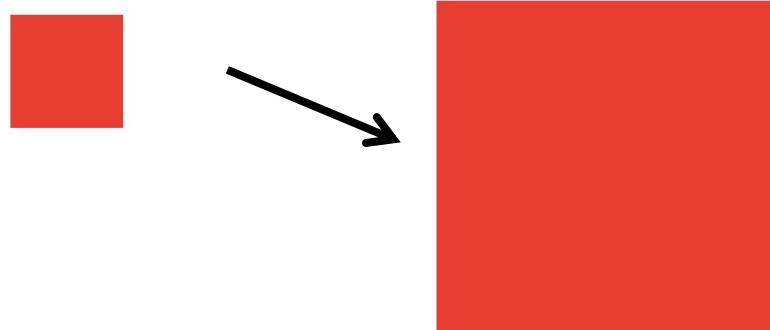
```
div {  
    box-shadow: 10px 10px;  
}  
  
div {  
    box-shadow: 10px 10px;  
}
```

This is a yellow <div> element
with a black box-shadow

How to Use CSS3 Transitions?

- To create a transition effect, you must specify two things:
 - the CSS property you want to add an effect to
 - the duration of the effect

```
div {  
    width: 100px;  
    height: 100px;  
    background: red;  
    -webkit-transition: width 2s, height 4s; /* For Safari 3.1 to 6.0 */  
    transition: width 2s, height 4s;  
}  
  
div:hover {  
    width: 300px;  
    height: 300px;  
}
```

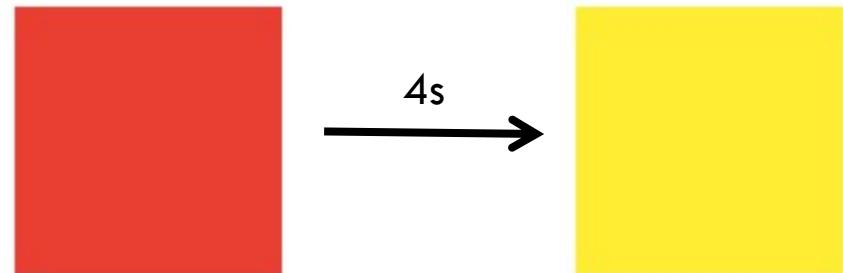


CSS3 Animations

- CSS3 animations allows animation of most HTML elements
 - An animation lets an element gradually change from one style to another
 - You can change as many CSS properties you want, as many times you want
 - To use CSS3 animation, you must first specify some **keyframes** for the animation
 - Keyframes hold what styles the element will have at certain times

```
/* The animation code */
@keyframes example {
    from {background-color: red;}
    to {background-color: yellow;}
}
```

```
/* The element to apply the animation to */
div {
    width: 100px;
    height: 100px;
    background-color: red;
    animation-name: example;
    animation-duration: 4s;
}
```



https://www.w3schools.com/css/tryit.asp?filename=trycss3_animation1

CSS3 Animations (2)

```
div {  
    width: 100px;  
    height: 100px;  
    background: red;  
    position: relative;  
    -webkit-animation: mymove 5s infinite; /* Safari 4.0 - 8.0 */  
    animation: mymove 5s infinite;  
}  
  
/* Safari 4.0 - 8.0 */  
@-webkit-keyframes mymove {  
    from {left: 0px;}  
    to {left: 200px;}  
}  
  
@keyframes mymove {  
    from {left: 0px;}  
    to {left: 200px;}  
}
```

