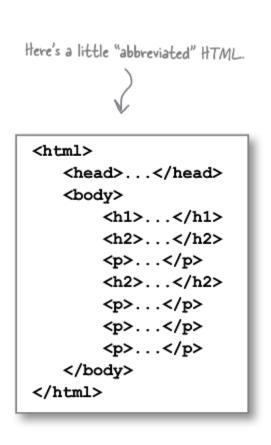
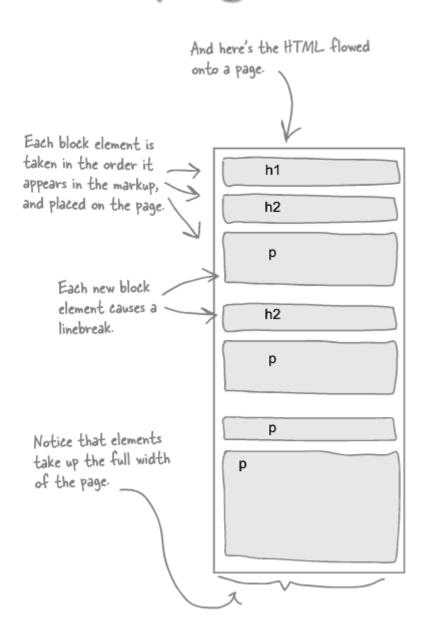
Lecture 11: Flow, Arranging elements, Common Layout Techniques

Dr. Jason Schanker

Flow: How the browser lays out pages

The browser uses flow to lay out the pages. Unless told otherwise, the browser puts the block elements listed at the top of the HTML on the top of the page and proceeds downward through the HTML, displaying each block element below the previous one, separating the blocks with linebreaks.





Block flow exercise

Look at the lounge.html file and draw the flow of the block elements (h1, h2, multiple p and div elements, u1).

Flow (cont.): Laying out inline elements

➤ Within the block elements, inline elements are laid out in the order they appear in the HTML from left to right; top to bottom. When laying out the inline elements, the browser tries to fill the available horizontal space within the block in which they are nested, introducing line breaks only when necessary (or when explicitly instructed with e.g., $\langle br \rangle$).

The text is treated as inline elements (with their own boxes).

If we take the inline content of this element and flow it onto the page, we start at the top left.

The inline elements are laid next to one another horizontally, as long as there is room on the right to place them.

Join us any evening for
these and all our other wonderful
<a href="beverages/elixir."
html" title="Head First Lounge
Elixirs">elixirs.



Here, there's room to fit all the inline elements horizontally. Notice that text is a special case of an inline element. The browser breaks it into inline elements that are the right size to fit the space.

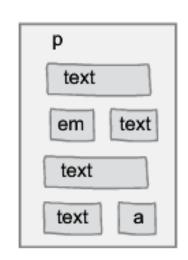
Flow (cont.): Laying out inline elements

> If the width of the available space of the block element decreases (e.g., because of browser resizing), the inline elements are adjusted accordingly, possibly breaking up text inline elements and making the block taller, as necessary to fit the content.

So what if we make the browser window a little thinner, or we reduce the size of the content area with the width property? Then there's less room to place the inline elements in. Let's see how this works.

Now the content has been flowed left to right until there's no more room, and then the content is placed on the next line. Notice the browser had to break the text up a little differently to make it fit nicely.

And if we make the content area even thinner, look what happens. The browser uses as many lines as necessary to flow the content into the space.



em

a

text

text

text

Padding/Margins of Inline Elements

When the browser lays out inline elements, it fully accounts for left and right margins for content appearing on the same line. However, top and bottom margins of nonreplaced inline elements* (ones in which dimensions are not determined externally, e.g., images) do **not** affect the flow of the other elements around them nor do the left and right margins affect the flow on lines below.

* http://stackoverflow.com/questions/10324527/margin-top-in-inline-element http://reference.sitepoint.com/css/replacedelements

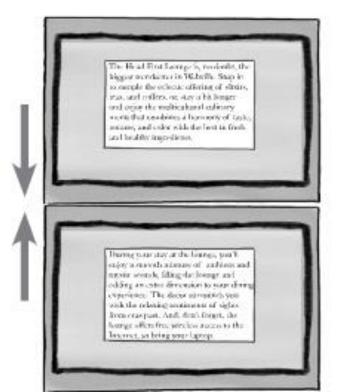
Consider this: 50 pixel greenish blue inset border and 20 pixel margins. See how the paragraph text is affected? The left and right margins are present on the line with the text, but not the one below.

And if I start a new paragraph, I may still go over the border.

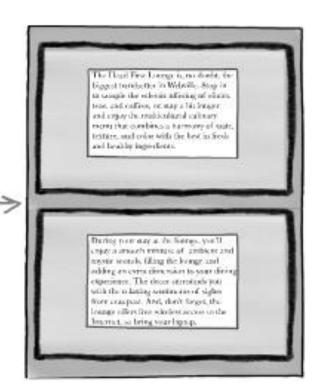
Padding/Margins of Block Elements

When the vertical margins of two block elements touch, only the larger of the two is used; their margins are **not** summed. This can even happen with a block element which is nested inside another.

When the browser places two block elements on top of each other, it collapses their margins.



Their shared margin is
the size of the larger
of the two margins.
Say the top element's
bottom margin is 10
pixels, and the bottom
element's top margin
is 20 pixels. Then the
collapsed margin will be
20 pixels.



Wrapping around block elements

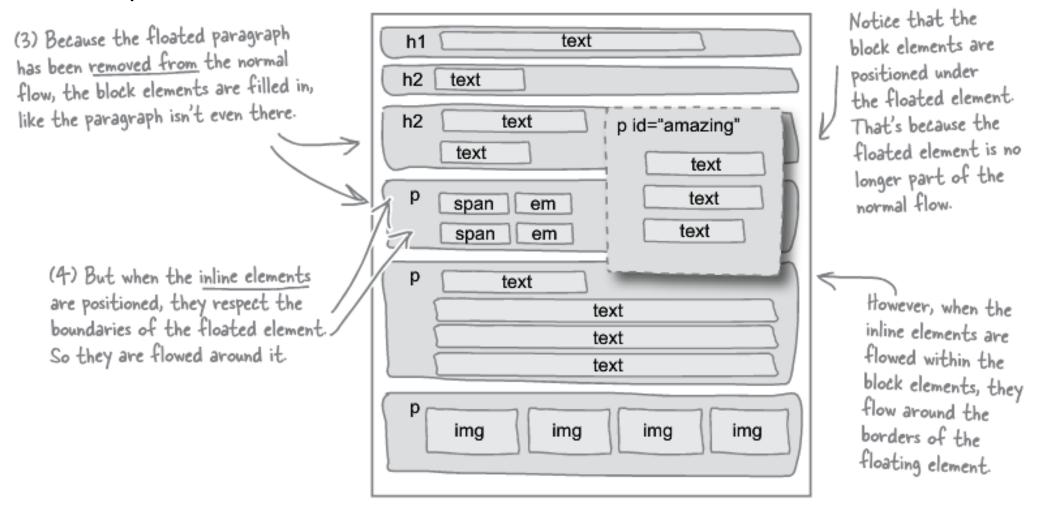
➤ Unless specified otherwise, blocks appear one on top of the other. Do you think adjusting the width of a block element will make block elements below it move up? Why or why not? Try adding the following width property to the lounge.css file to see what happens. (200 pixels is 1 quarter of the typical 800 pixels desktop browser window.)

➤ What rule might you consider adding to make the Elixirs section appear on the right with the blocks below it wrapping around it? →



Floating Elements

➤ When you float a block element to the right, it aligns the element on the right in the vertical position it would be normally placed. But it removes the element from the block flow so that other block elements below it move up as if it's not present. However, inline elements within these blocks are placed as to not encroach upon the boundaries of the floated element.



Floating Exercise I

Reposition the elixirs section in the HTML and add

width: 200px; and float: right; to the elixirs rule in the CSS so that the top of the Elixirs section floats on the right with its top lining up with the "Welcome to the Head First Lounge" text.



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 lactor.



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 theck in an email over an elixir, or are here for an out-ofthe-ordinary discour, you'll find our knowledgeable service staff pay attention to every detail. If you're not fully satisfied, have a Blueberry Blue Elixir on us.



Starbuzz Exercise: Sidebar I

>Save the source of:

http://wickedlysmart.com/hfhtmlcss/chapter11/starbuzz/index.html

and get the CSS file from:

<u>http://wickedlysmart.com/hfhtmlcss/chapter11/starbuzz/starbuzz.css</u> and get save the following in an images directory placed in the same folders as the files:

http://wickedlysmart.com/hfhtmlcss/chapter11/starbuzz/images/bag.gif http://wickedlysmart.com/hfhtmlcss/chapter11/starbuzz/images/header.gif

- ➤ Modify the HTML and CSS as necessary to make the sidebar have a width of 280 pixels and float on the right below the header div. Once you get this to work, adjust the browser window so that the text wraps underneath the right sidebar. To get the two-column look back, adjust the main section's padding (i.e., Add padding on the right to account for the TOTAL width of the sidebar, padding and margin included.) The space between the columns is called the gutter.
- ▶ Problem with this technique: Main content comes after the sidebar content in the HTML. Structurally, this does not seem correct because the main content is more important. More importantly, we often only want one column on mobile devices with small screens. In this one-column scenario, the sidebar will appear before the main content.



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

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 guestiantly and helping to meet the caffeine needs of a whole new set of customers.

ARBUZZ COFFEE BEVERAGES

We arriety of caffeinated beverages to choose from at Starbuzz, including our house. A 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 (ath you. Order your coffee today using one)

ORDER ONLINE with the BEAN MACHINE

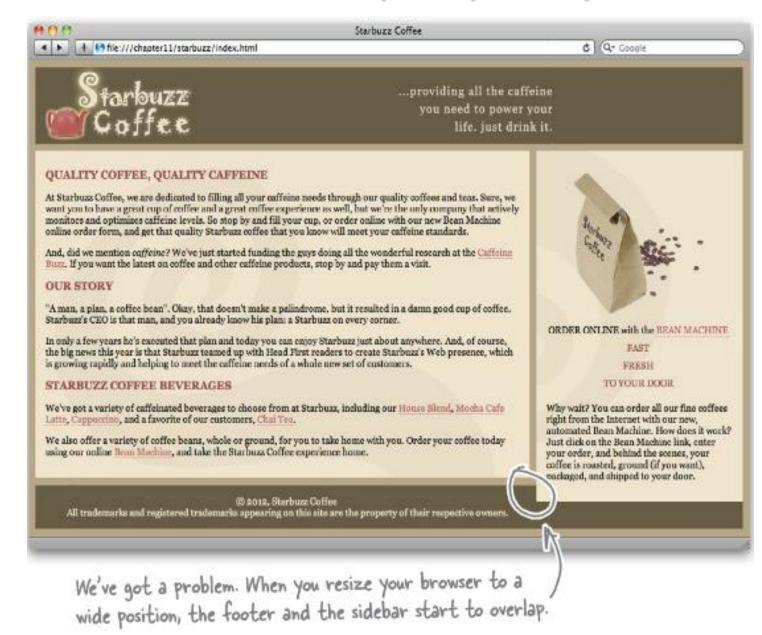
FAST

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 seenes, your coffee is roasted, ground (if you want), packaged, and shipped to your door.

Fixing one last problem: Clear property

- ➤ Since we didn't adjust the margin of the footer, the sidebar floats over it. To fix this (and allow the footer to extend underneath), we can add clear: right; to the footer rule, which requests that no content on the right float over it other possible values are left and both). To do this, the footer will move down as necessary.
- Problem: If main column is shorter than sidebar, there will be a gap between the main column and the footer.



Exercise: Sidebar Exercise Attempt II

- Instead of making the sidebar float to the right, have the main content float to the left. Make the width of the main content bar 420 pixels. Do this by mirroring the previous approach. The sidebar plays the role of the main content (e.g., no width) while the main content takes the role of the sidebar.
- Tradeoff: Often, no approach will be perfect. Whereas this approach is better in terms of information (main content will be above sidebar content in the HTML), it is worse for design (sidebar expands in width as we make the browser window wider). Ideas for other approaches?

Liquid and Frozen Designs

- Designs so far expanded to fill the browser width, making them <u>liquid</u> <u>layouts</u>.
 - ☐ Potential Problem: Liquid layouts are more difficult to design well and manage.
- ➤Ones in which the layout remains unchanged with browser resizing is a **frozen layout**.
 - ■Potential Problem: Frozen layouts don't make the best use of the available space, leaving behind empty areas devoid of content. If made too wide, they can cause the undesirable horizontal scrolling requirement. Also, frozen layouts are generally poor choices for mobile.
- Designs in between liquid and frozen layouts are <u>jello layouts</u>. This takes a frozen layout but makes it look better by centering it, distributing the empty space.
 - □ Potential Problem: Jello layouts still suffer from the problems of frozen layouts.

Exercise: The two-column Frozen Layout

➤ Enclose all of the body element content in between <div id="allcontent"> and </div>. Then include the following CSS:

Now we're going to use this <div> to constrain the size of all the elements and content in the "allcontent" <div> to a fixed width of 800 pixels. Here's the CSS rule to do that:

```
#allcontent {
    width: 800px;
    padding-top: 5px;
    padding-bottom: 5px;
    background-color: #675c47;
}
```

We're going to set the width of "allcontent" to 800 pixels. This will have the effect of constraining everything in it to fit within 800 pixels.

While we're at it, since this is the first time we're styling this <div>, let's add some padding and give it its own background color. You'll see this helps to tie the whole page together.

The outer "allcontent" <div> is always 800 pixels, even when the browser is resized, so we've effectively frozen the <div> to the page, along with everything inside it.

Exercise: Morphing to the Jello Layout

```
#allcontent {
    width: 800px;
    padding-top: 5px;
    padding-bottom: 5px;
    background-color: #675c47;
    margin-left: auto;
    margin-right: auto;
Rather than having fixed left and right
    margins on the "allcontent" <div>, we're
    setting the margins to "auto".

**Bound of the margins on the "allcontent" <div>, we're
    setting the margins to "auto".

**Bound of the margins on the "allcontent" <div>, we're
    setting the margins to "auto".

**Bound of the margins on the "allcontent" <div>, we're
    setting the margins to "auto".

**Bound of the margins on the "allcontent" <div>, we're
    setting the margins to "auto".

**Bound of the margins on the "allcontent" <div>, we're
    setting the margins to "auto".

**Bound of the margins on the "allcontent" <div>, we're
    setting the margins to "auto".

**Bound of the margins on the "allcontent" <div>, we're
    setting the margins to "auto".

**Bound of the margins on the "allcontent" <div>, we're
    setting the margins to "auto".

**Bound of the margins on the "allcontent" <div>, we're
    setting the margins to "auto".

**Bound of the margins on the "allcontent" <div>, we're
    setting the margins to "auto".

**Bound of the margins on the "allcontent" <div>, we're
    setting the margins to "auto".

**Bound of the margins on the "allcontent" <div>, we're
    setting the margins on the "allcontent" <div
, we're
    setting the margins on the "allcontent" <div
, we're
    setting the margins on the "allcontent" <div
, we're
    setting the margins on the "allcontent" <div
, we're
    setting the margins on the "allcontent" <div
, we're
    setting the margins on the "allcontent" <div
, we're
    setting the margins on the "allcontent" <div
, we're
    setting the margins on the "allcontent" <div
, we're
    setting the margins on the "allcontent" <div
, we're
    s
```

If you remember, when we talked about giving a content area a width of "auto", the browser expanded the content area as much as it needed to. With "auto" margins, the browser figures out what the correct margins are, but also makes sure the left and right margins are the same, so that the content is centered.

Absolute positioning

- We can also specify an absolute position for an element in which case it is completely removed from the flow and all other elements, block or inline, are positioned without regard for this element. In this case, text and other content may be hidden underneath absolutely positioned elements.
- Absolute positioning can be specified by setting the position property to absolute and then specifying a values for top, bottom, left, or right in either pixels or percentages to specify the distance away from the top, bottom, left, and right borders of the browser window WHEN it's showing the top left corner of the web page, respectively.
 - □ Example: top: 20px; right: 10%; positions the element 20 pixels down from the top and 10% of the width of the browser window to the left of the right border so e.g., 80 pixels if the browser is 800 pixels wide.
- The default value for the position property is static, which gives the browser the decision of exact element placement.
- Absolutely positioned elements can be layered via the z-index attribute; if one element has a higher z-index value than another one, then it is placed on top.

Absolute Positioning Example

The sidebar is positioned 200

pixels from the right of the page. Because sidebar is now absolutely positioned, it div id="header" is removed from the flow and positioned according div id="sidebar" to any top, left, right, div id="main" or bottom properties that are specified. div id="footer" Because the sidebar is out of the flow, the other elements don't even know it is there. and they ignore it totally. Elements that are in the flow don't even wrap their inline content around an absolutely positioned element. They are totally oblivious to it being on the page.

And the sidebar is positioned 100 pixels from the top of the page.

#sidebar {

position: absolute;
top: 100px; Next we set top and right: 200px; width: 280px;

And we also give the <div> a width.

Absolute Positioning Sidebar Exercise

- Set the sidebar to have an absolute position of 128 pixels below the top of the page and 0 pixels from its right (touching the right side). Then specify its width to be 280 pixels. For the main content area, set the right margin to be 330 pixels to account for (width + 2×cellpadding + 2×margin = 280 + 2×15 + 2×10 = 280+30+20)
 - □Q: Where did the 128 pixels come from? See if you can visualize how the page will be laid out before viewing.
- ➤ **Problem**: When the width is small, the right column may go on top of the footer and the clear property won't fix this since the absolutely positioned sidebar is completely ignored by other elements being placed on the page.

CSS Table Display Layout

Tables are like spreadsheets, a grid of his is the first column cells which may or may not be surrounded by borders.

In the mid 1990's, multicolumn layouts is the or layouts in which you wanted items first row to line up were done with borderless HTML tables. However, now with CSS tables available for layout, HTML tables should only be used when structurally the content is tabular in nature, not for e.g., a 2-column layout which is about presentation.

A CSS table is created by nesting block elements; one has a table display which contains blocks with row displays which contain blocks with cell displays.

In this table, we have three rows and two columns, for a total <div> > of six cells. <div> > The table will automatically <div> > expand to accommodate the cell widths and heights. lace a block element. Notice that, because we only put block elements into a table, we've wrapped the images in a <div>.

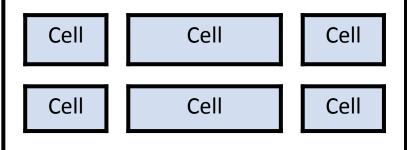
CSS Table Two-column Exercise Part I

- Sketch the setup for creating a table for the two-column layout.
- II. Introduce new div elements as necessary to represent the table and single row with appropriately named id attributes into HTML code. Do this so that the elements are nested properly for implementing the table setup.
- III. Add new CSS rules for the elements representing the table and the table row in which you specify display:table; and display:table-row;, respectively. Add display:table-cell; to the rules selecting the div elements serving as the cells.

Border-spacing and Vertical-align Properties

>Border-spacing: In a block element styled with a table display, the value of this attribute specifies the spacing between borders (effectively acting as a combined margin attribute for the table's block elements representing cells: margin space surrounds borders in box

model).



Border spacing in white

▶Vertical-align: In a block element styled with a table-cell display, you can specify top, middle or bottom so that the contents of the cell are aligned at its top, middle, or bottom, respectively.

CSS Table Two-column Exercise Part II

- IV. Specify 10 pixels of border spacing in the table rule and remove the margins in the block elements working as table cells. Then set the vertically align the content of the table cells at their tops.
- V. Does the spacing look a little off to you at the top and bottom? Try correcting it.
 <u>Hint</u>: Border spacing and margins do <u>NOT</u> merge when they touch.

Exercise: 3-column CSS

```
<div id="drinks">
 <h1>BEVERAGES</h1>
  House Blend, $1.49
   Mocha Cafe Latte, $2.35
   Cappuccino, $1.89
  Chai Tea, $1.85
 <h1>ELIXIRS</h1>
    We proudly serve elixirs brewed by our friends
    at the Head First Lounge.
   Green Tea Cooler, $2.99
   Raspberry Ice Concentration, $2.99
   Blueberry Bliss Elixir, $2.99
   Cranberry Antioxidant Blast, $2.99
  Chai Chiller, $2.99
   Black Brain Brew, $2.99
```

>Add a column to appear on the left that fills 20% of the width of the browser window. Include 15 pixels of cellpadding, make its content be aligned at the top of the cell, and use a background color with hex code #efe5d0. The HTML for this column is on the left.

Summary of Layout Tools

➤ Floating Layout:

- □Common Use: Placing an image or a column of content and having the other content wrap around it.
- □ Possible Disadvantage: Content may need to be rearranged in the HTML in an order that is not reflective of the relative importance (e.g., sidebar's content placed before main content)

➤ Frozen/Jello Layout

- □ Common Use: For pages primarily viewed on more or less fixed-sized screens (e.g., only desktop computers); Good for having exact control over your layout.
- Possible Disadvantage: Does not make good use of available space; bad when your users view the page on many different screen sizes (e.g., mobile and desktop)

➤ Absolute Positioning Layout:

- ☐ Common Use: When you want to have exact control over one or more parts of your page, but want a liquid layout to make good use of the available space.
- ☐ Possible Disadvantage: To design the page to your liking, you may need to accept an undesirable overlapping of content for certain browser window sizes.

➤ Table Display Layout:

- □Common Use: For grid-like layouts; to tile your page layout into columns (and/or) rows.
- □Possible Disadvantage: More complex HTML (adding divs) that's harder to modify in response to needed changes in layout, particularly when you want to have a cell in a column that's wider than ones above/below.

When the browser window is more than BOOpx wide, you get all this extra space over here to the right.

Header Does Not Expand and Contract



The rest of the page resizes nicely as you widen and narrow the browser window.

And when the browser is narrower than 800px wide, the slogan part of the header image falls off the edge of the browser window!



← 日前 ○ Rie:///chapter11/sourbugg/index.html

C C Starburg Coffee

...providing all the caffel you need to power yo life, just drink

BEVERAGES

House Blend, \$1,45

Mode Calclain, 80.35

Cappacdno, 81.89

Chis Tea, \$1.54

ELIXIRS

We proudly serve elistics browed by confriends at the lived First Longs.

Gerca Tea Coolea, 80-99

Haspberry Ice Concentration, 80.00

Blackerry Blass Elber, Sa. 99

Cranberry Authoridant Hast, Str. 99

Chal Chiller, \$2.00

Black Brain Brow, 62, 99

QUALITY COFFEE, QUALITY CAFFEINE

At Starburz Coffee, we are declarated to filling all your caffeins needs through our quality offices and teat, Sare, we seemt you to have a great onpot office and a great coffee experience as well, but we're the only company that actively mentions and optimizes coffees brook. So that by and fill your cup, or order online with our new Boan Machine online order form, and gut that quality Starburz coffee that you know well most your coffees that you know well most your coffees that starburs.

And, did we mention applicac? We've just started funding the pays doing all the wonderful research at the Caffeline Buss. If you want the latest on coffee and other caffeans product, stop by and pay them a visit.

OUR STORY

"A man, o plan, a coffee bean". Okay, that doesn't make a palindroams, but it resulted in a durin good cup of orfice. Somburs's CEO is that man, and you already know his plan; a Starburs on every corner.

In only a few years he's excented that plan and today you can enjoy Starbaze just about anywhere. And, of ourse, the log cross that year is that Surbaze tenseed up with Head Pirst readers to create Starbaze's Web presence, which is grewing rapidly and helping to much the cuffeine needs of a whole new set of customers.

STARBUZZ COFFEE BEVERAGES

We've got a variety of caffeinated hoverages to choose from at Starbase, including our House Illend, Mocha Cafe Latte, Cappareine, and a favorite of our curtomers, Chaffee.

We also offer a variety of ooffee beans, whole or ground, for you to take home with you. Order your coffee today using our colline flow Markers, and take the Starbusz Coffee experience borne.



OUDER ONLINE with the REAL

MACHEN

PART

TO YOUR DOOR

Why would You can under all our flac coffices right from the Internet with our new, as tomated lies a Machine. How show it work? Just dick on the Bean Machine link, enter your order, and befored the second, your coffice is reasted, ground (if you want), puckaged, and shipped to your chair.

© note. Starbuse Collins

All trademarks and registered trademarks approxing on this sits on the property of their respective owners.

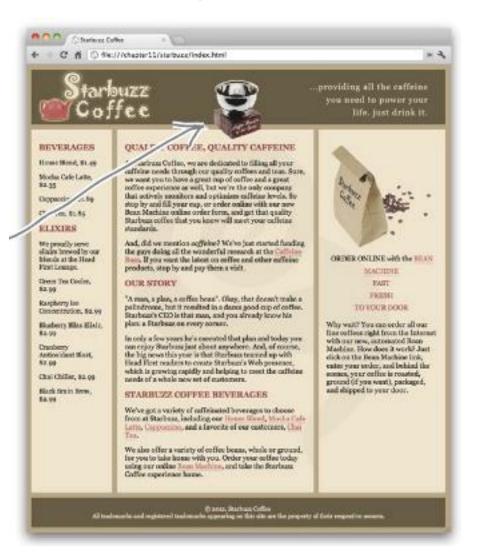
Exercise: Fixing the Header

➤ Use the following HTML for the header and modify the CSS (and header HTML if necessary) to have the slogan stay glued at a fixed width to the right of the browser window.

```
<div id="header">
  <!--<img src="images/header.gif" alt="Starbuzz Coffee header image">-->
  <img id="headerLogo"</pre>
        src="images/headerLogo.gif" alt="Starbuzz Coffee logo image">
  <img id="headerSlogan"</pre>
        src="images/headerSlogan.gif"
        alt="Providing all the caffeine you need to power your life.">
</div>
                                  We can easily split the header into two
                                  gif images (they both have a transparent
                                                                 ...providing all the saffsine
                                  background with a matte that works
                                                                     you need to power your
                                  perfectly with our coffee-colored
                                                                          life, just drink it.
                                  background color in the header).
```

Exercise: Positioning an Award Top and Center

Add the image at the URL http://www.starbuzzcoffee.com/images/awar d.gif to the page at the position shown to the right 30 pixels from the top and 365 pixels from the left, overlapping the header and main divs) by using HTML and CSS. Give the image an alt of, "Roaster of the Year award".



Fixed Positioning

- From can fix an element in a certain position in the user's viewport (part of the web page that's visible in the browser window) by setting the position property to fixed and top, left, right, or bottom accordingly in the rule selecting the element. You can even use negative values to make the element partially run off the page by that amount (e.g., right: −20px; pushes the element 20 pixels out of the right side of the viewport.)
- ➤ Unlike absolute positioning which positions the element with respect to the browser window's borders when displaying the top left corner of the web page, fixed positioning positions the element with respect to the borders of whichever part of the page is in the user's viewport at the moment, making it appear "stuck" or "fixed" at that position on the page.

Exercise: Fixing the ticket

- I. Add the image at the URL

 http://www.starbuzzcoffee.com/images/ticket.gif to

 the page and have it link to

 http://wickedlysmart.com/hfhtmlcss/chapter1

 1/starbuzz/freecoffee.html

 I diversity of the link a title of, "Click here to get your free coffee!"

 and the image an alt of "Starbuzz coupon ticket"

 Make this HTML the content of a diversity element with an id of coupon.
- II. Remove any potential borders with the following in the CSS:

```
#coupon a, img {
    border: none;
}
```

III. Give the image the fixed position shown to the right (350 pixels below the top and 90 pixels LEFT of the left of the browser window so it appears to be coming out of the left side of the browser window.



Extras: Relative positioning and NO CSS

- ➤ You can also set position to relative in which case the element remains in the flow of the page and is moved relative to where it would've be (e.g., left: 50px moves it 50 pixels left of where it would've been). In particular, it does <u>not</u> "fix" the element to the screen as you scroll. See: http://www.w3schools.com/cssref/pr class position.asp
- To see how much difference the CSS makes, comment out the place where you link in the sheet in the HTML by surrounding it with <!-- and --> (<!-- and --> enclose HTML comments, which are ignored when rendering the web page, but can be used as notes for yourself or other people looking/using your HTML).

Notes

☐ This is primarily a summary of Chapter 11 of Head First HTML and CSS, 2nd Edition by Elisabeth Robson and Eric Freeman, 2012. It contains images, exercises, and code from the book.