WEEK 08
3/16/15

Sizing and positioning with CSS

CORE LAB INTERACTION
MONDAY 7:00—9:40

This week

- Housekeeping
- Reading discussion
- Design discussion
- Week 7 recap and questions
- DIVs, classes, and IDs
- Break
- Work session

Last week's reading

- Ways of Seeing Instagram
- Designing a Brand Simple Enough for Non-Designers to Use
- History of the Selfie

<DIV>, IDS, AND CLASSES:

Recap and questions

Sizing and Positioning.

We’re going combine the structural elements ( ) and hooks (classes) we learned about last week with sizing and position.

With all of these skills, we’ll be able to create almost any layout we design(!)

Sizing

Along with positioning, we'll be using sizing to define the layout of our page.

As with a lot of the principles we've learned, less is more when it comes to sizing.

Pixels

Pros:
- Precise adjustments

Cons:
- Not flexible

Percentages

Pros:
- Very flexible
- Low maintenance

Cons:
- Not precise
- Don’t work in all scenarios
- Requires the parent container to have a clearly defined size

Demo!

Positioning

In CSS, we have a few options for positioning things:

- Static
- Relative
- Absolute
- Fixed
- We can also ‘float’ elements

Positioning

By default, elements have position static.

An element with a fixed position is positioned relative to the browser window, and will not move even if the window is scrolled.

An element with a relative position is positioned relative to its normal position.

An element with absolute position is positioned relative to the first parent element that has a position other than static.

Demo!

Positioning strategies

Choose the type of positioning based on what you want the element to accomplish.

Use mostly static or relative positioning.

Examples

Layering

We can use the ‘z-index’ value to change the layering of elements.

Demo!

Floating

With CSS float, an element can be pushed to the left or right, allowing other elements to wrap around it.

- Elements are floated horizontally, this means that an element can only be floated left or right, not up or down.
- A floated element will move as far to the left or right as it can. Usually this means all the way to the left or right of the containing element.
- The elements after the floating element will flow around it.
- The elements before the floating element will not be affected.

Demo!

Other strategies

As you do this more, you'll learn some other tricks for positioning elements. Let's look at one super useful one for arranging lists.

Demo!

Some caveats

SOME CAVEATS

Fixed sizing

Avoid setting the size of too many elements. It present a lot of problems down the road:

- New content might not fit in the original space or might interfere with other page elements
- Makes mobile websites hard
- Makes future changes and maintenance hard

SOME CAVEATS

Centering

Centering has always been much harder in CSS than it should be. There are different strategies depending upon the scenario.

Horizontal centering is a bit easier.

Vertical centering is tough – avoid it when possible (for now).

Demo!

Break

ASSIGNMENT 1: **DUE MONDAY, MARCH 30**

Tying it all together

Creating an accurate version of an existing webpage

- Copy last week's code into a new project
- Use all the tools – HTML and CSS – to recreate a very close version of the Warby Parker product page
- Good projects will visually represent the original; excellent projects will also implement the best practices we've discussed in class

ASSIGNMENT 2: **DUE MONDAY, MARCH 30**

Codecademy

CSS Element positioning

- Do the final two modules in Codecademy: CSS Positioning and Build a Resume
- Send me screenshots of the completed lessons

NEXT WEEK

Spring break!

Remember to turn in the assignments and do the reading before we meet

AFTER THE BREAK

Creating and building original designs

Note: There’s a small chance I’ll be late to class on Monday. I’ll keep you updated if it looks like I’m going to be.