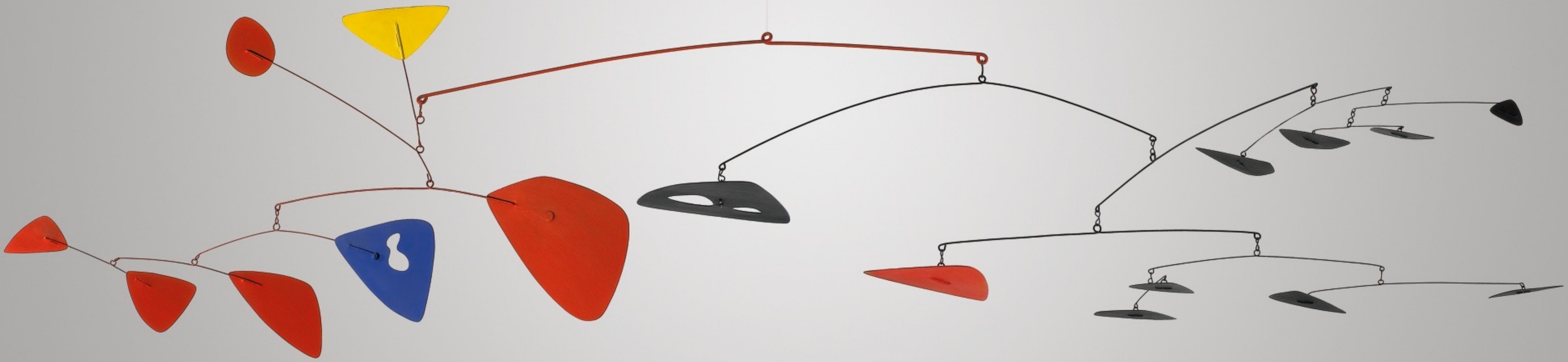


WELL-STRUCTURED HTML WITH CSS

WDI 26 WEEK01 DAY02 MODULE02

Objectives

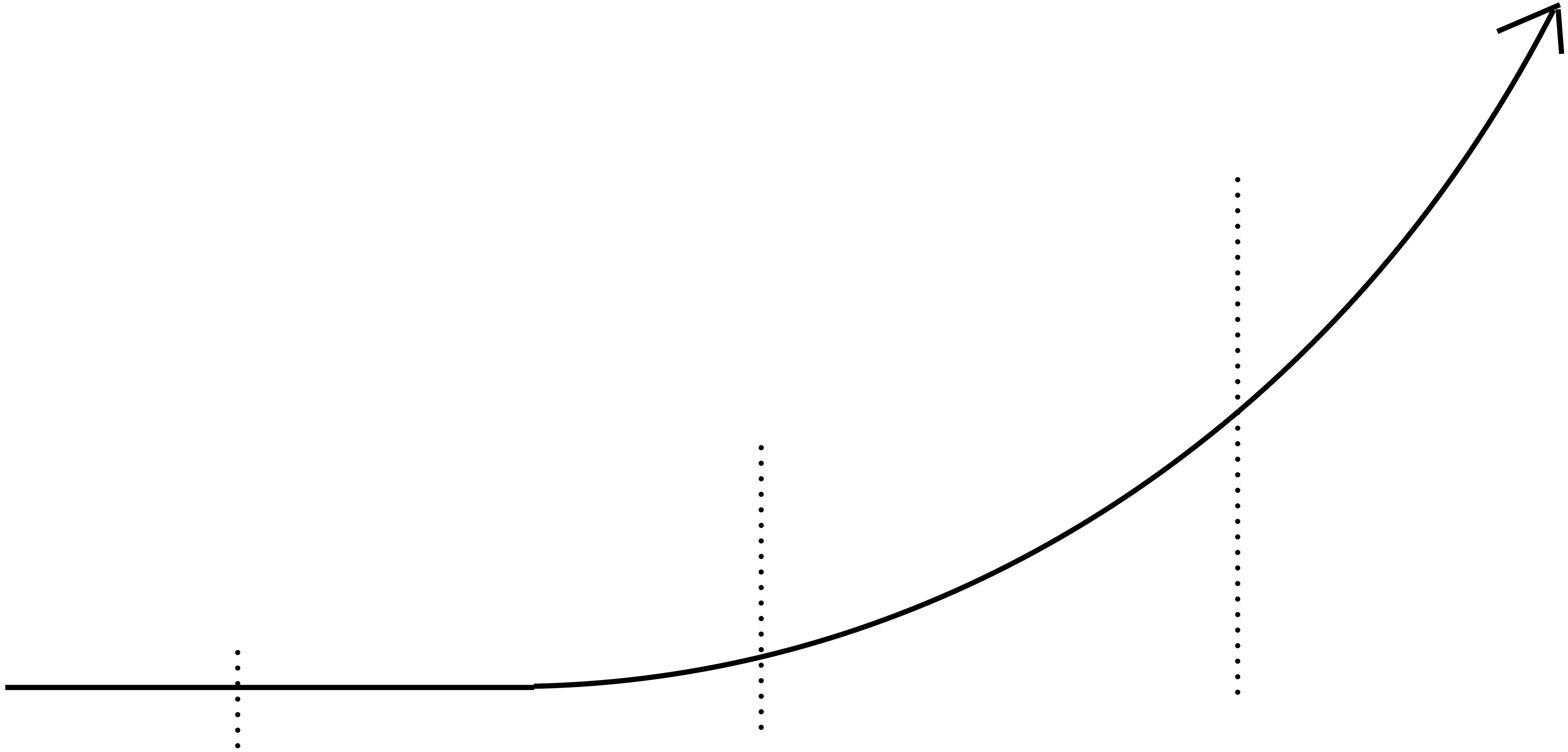
OBJECTIVES

- Hand-code well-structure HTML
- Write better CSS
- Debug CSS more quickly
- Understand HTML grid

WEEK01 DAY02 MODULE02 TIMING

5 min	**INTRO**
15 min	**CSS Review**
15 min	**ABSTRACT THINKING**
30 min	**CSS with EMMET**
10 min	**EXTRA PRACTICE (OPTION / Q&A)**

The WDI Learning Experience



THE BASICS

- **CSS** stands for Cascading Style Sheets

- An **element** is native HTML.
  A CSS **class** is used when you want to group
  some elements and apply the same styles to them.
  An **ID** is an unique identifier
  and can only be used ONCE per page or document.

SPECIFICITY

- **Specificity** allows the browser to determine which CSS rule applies to a given element on a page:

  Lower css rules / declarations overwrite higher ones.
  More ***specific*** selectors beat less specific ones
    - id > class > tag
      - no number of tags can beat a class
      - no number of classes can beat an id
    - Inline styles > Internal styles > External styles
    - **!important** trumps all of the above.

THE BOX MODEL and CSS BEST PRACTICES

- **The Box Model** A: HTML elements can have margin, border and padding properties. Margin is the space outside of the element's content. The border is the outer edge of the element. The padding is the space between the content and its border.

- **Inline, internal and external style sheets.**
  Inline is worst
  Internal uses the HTML <style></style> tag
  External offers separation of concerns

VALUES OF CSS DISPLAY ATTRIBUTE

- An inline element has no line break before or after it. This makes the element sit on the same line as another element, but without formatting it like a block. It only takes up as much width as it needs
- (not the whole line). Inline places all your elements on a single line. The bad news is that it doesn't maintain their "box"ness
- A block element has some whitespace above and below it and does not tolerate any HTML elements next to it. This makes the element a block box. It won't let anything sit next to it on the page and takes up the full width.
- An inline-block element is placed as an inline element (on the same line as adjacent content), but it behaves as a block element. This makes the element a block box but will allow other elements to sit next to it on the same line.
- If you assign none as the value of the display, this will make the element and its content disappear from the page entirely!

VALUES OF CSS POSITION ATTRIBUTE

- Specifying `position:absolute` _removes the element from the document_ and places it exactly where you tell it to be.

- An element with fixed position is positioned relative to the browser window. It will not move even if the window is scrolled, so a fixed positioned element will stay right where it is creating an effect a bit like the old school "frames" days.

- HTML elements are positioned static by default. A `position:static` element is always positioned according to the normal flow of the page and are not affected by the top, bottom, left, and right properties.

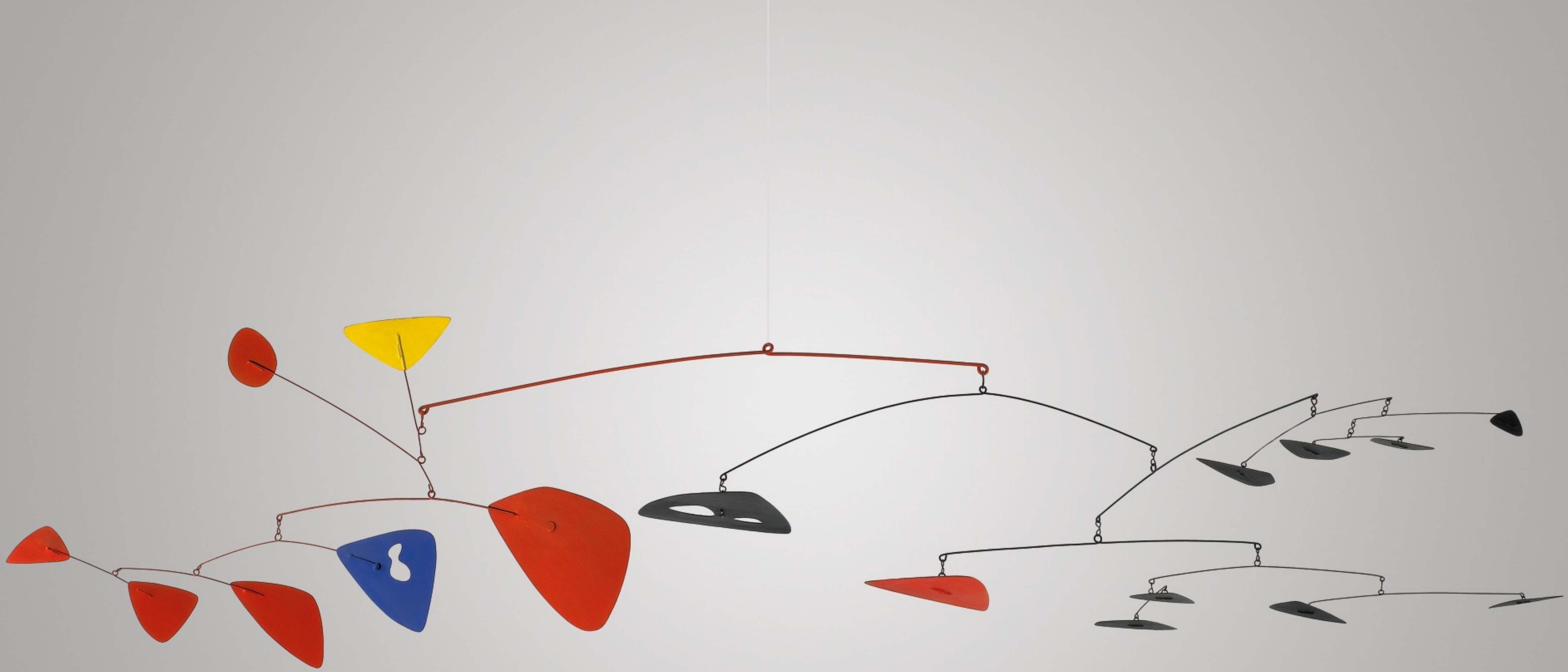
- Declaring `position:relative` allows you to position the element top, bottom, left, or right relative to where it would normally occur, relative to its parent div or container.

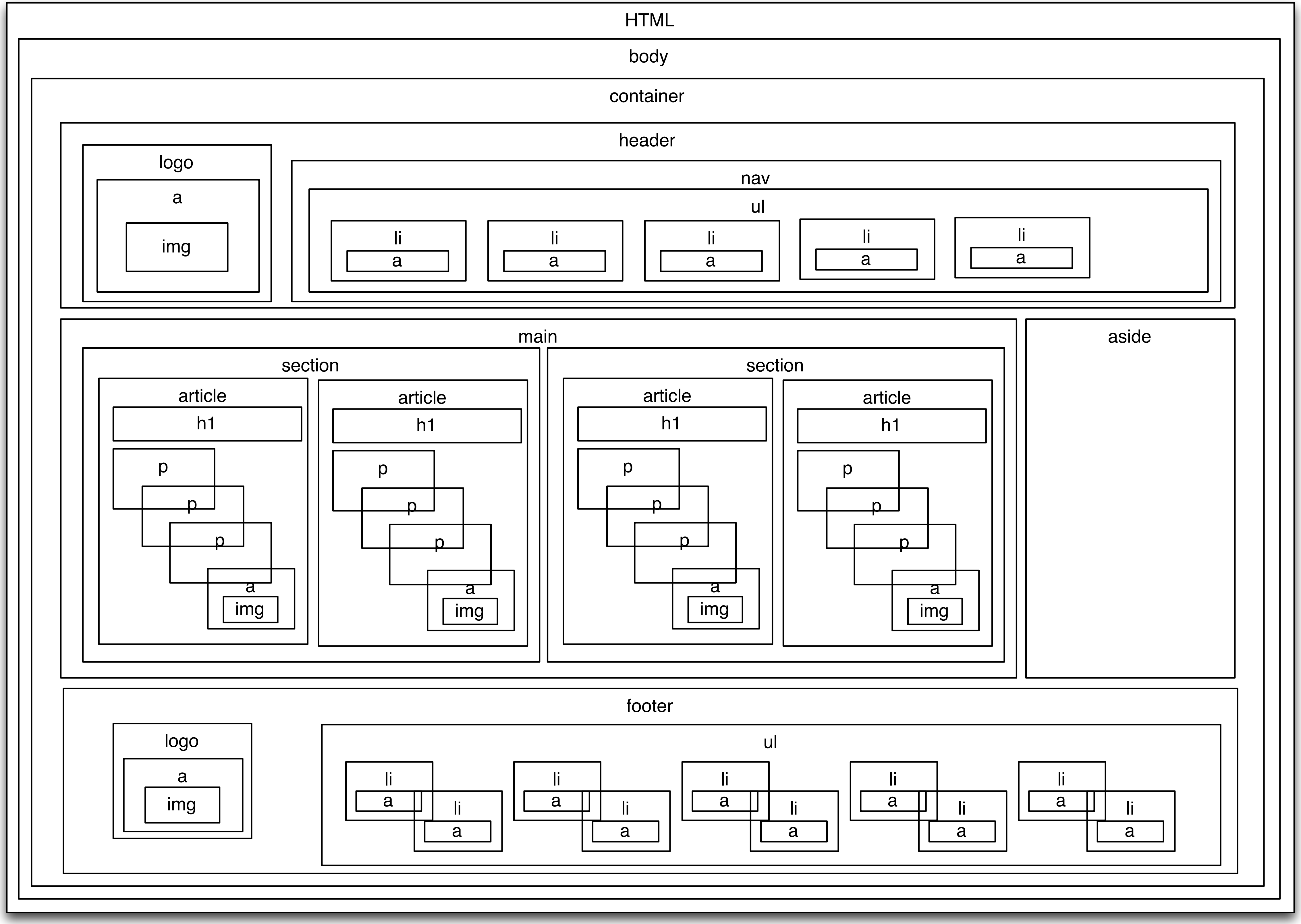
FLOATS vs. CLEARS

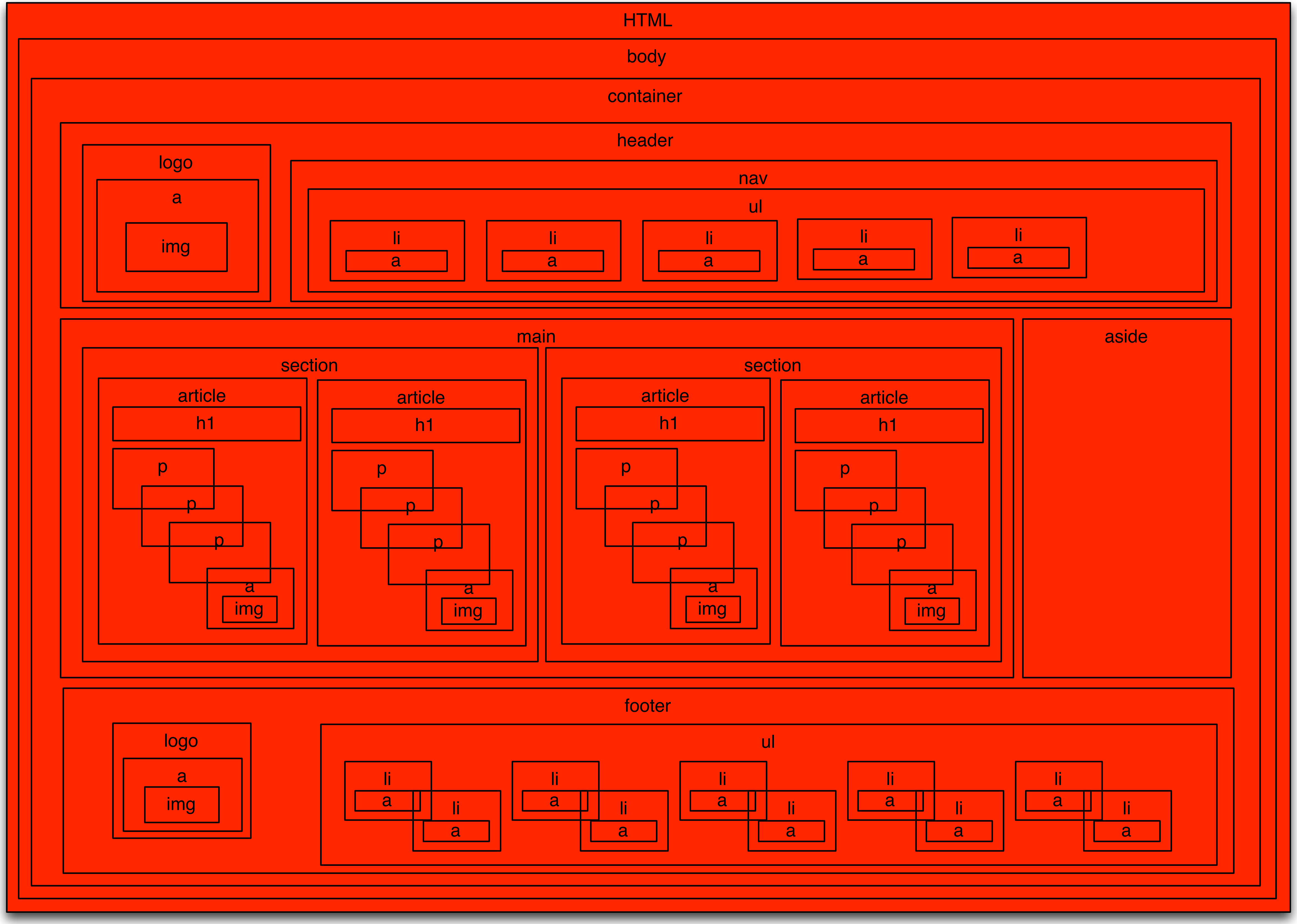
- While **floats** make other elements aware of their location and get text hugs, **clears** make other elements aware and are told not to touch.

REVIEW

- What does **CSS** stand for?
- Explain the difference between an **element**, a **class** and an **ID**
- Describe the concept of **specificity** and how it is used
- Explain the CSS **box model**
- What values can we use with the **'display'** property and what do they do?
- What are the three ways to **implement** CSS styling and which is best practice?
- What are the differences between **absolute**, **fixed**, **relative** and **static** positioning?
- Explain **float** and **clear**




HTML
body
container
header
logo
a
img
nav
ul
li
a
li
a
li
a
li
a
li
a
main
section
article
h1
p
p
p
a
img
article
h1
p
p
p
a
img
section
article
h1
p
p
p
a
img
article
h1
p
p
p
a
img
aside
footer
logo
a
img
ul
li
a
li
a
li
a
li
a
li
a
li
a
li
a
li
a
li
a
li
a



HTML

body

container

header

logo

a

img

nav

ul

li
a

li
a

li
a

li
a

li
a

main

section

article
h1
p
p
p
a
img

article
h1
p
p
p
a
img

section

article
h1
p
p
p
a
img

article
h1
p
p
p
a
img

aside

footer

logo

a

img

ul

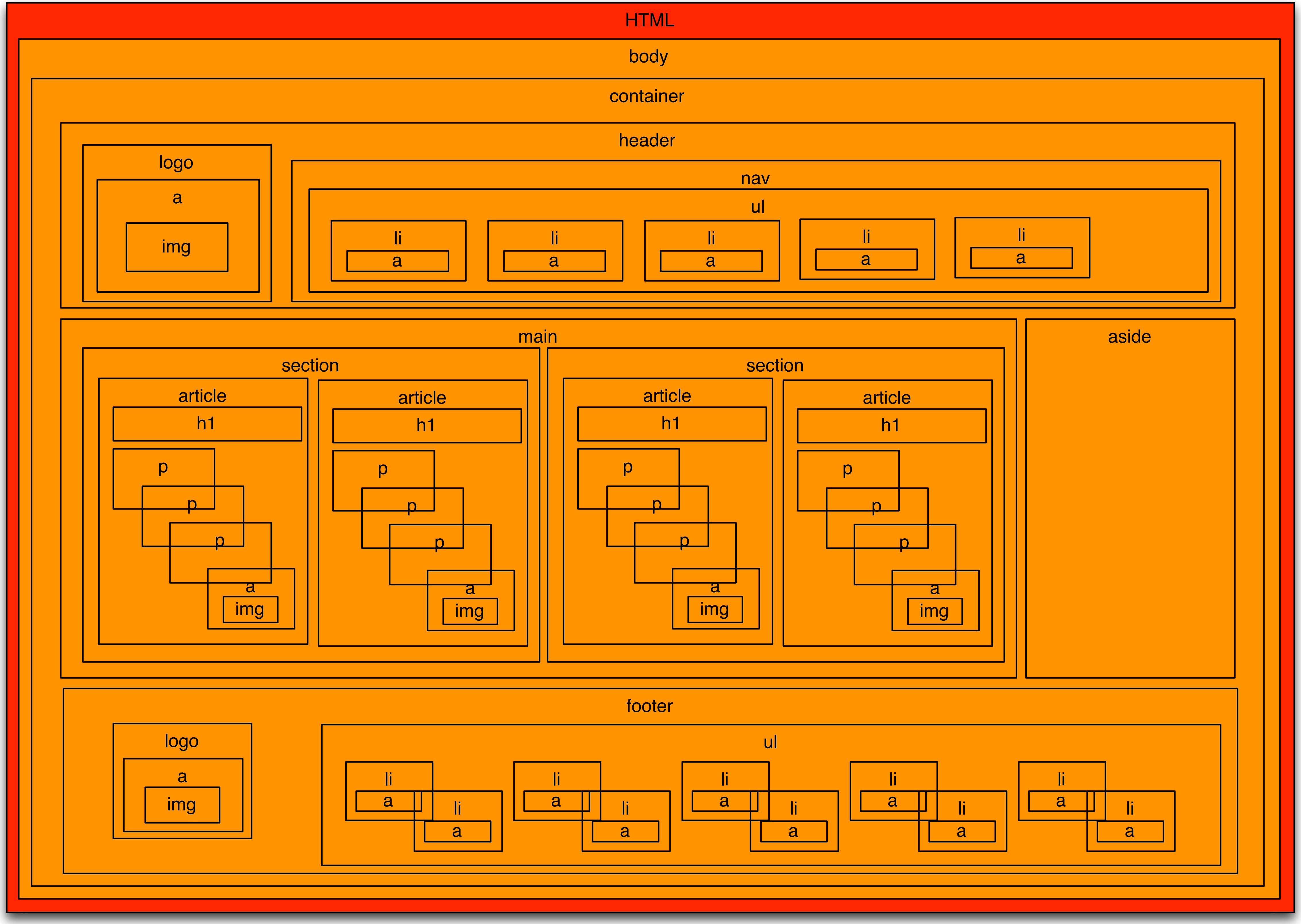
li
a
li
a

li
a
li
a

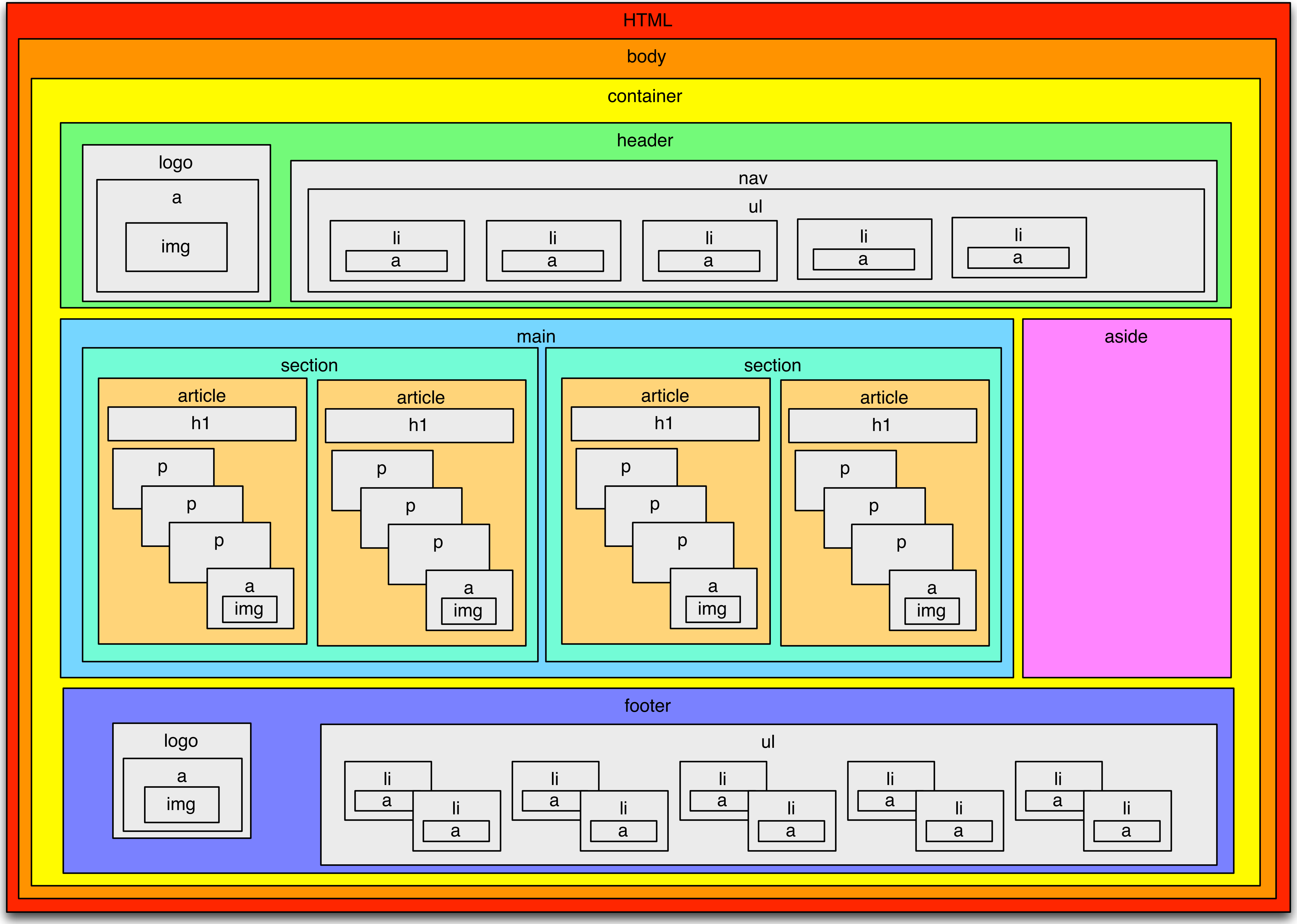
li
a
li
a

li
a
li
a

li
a
li
a



HTML
body
container
header
logo
a
img
nav
ul
li
a
li
a
li
a
li
a
li
a
main
section
article
h1
p
p
p
a
img
article
h1
p
p
p
a
img
section
article
h1
p
p
p
a
img
article
h1
p
p
p
a
img
aside
footer
logo
a
img
ul
li
a
li
a
li
a
li
a
li
a
li
a
li
a
li
a
li
a
li
a



HTML
body
container
header
logo
a
img
nav
ul
li
a
li
a
li
a
li
a
li
a
main
section
article
h1
p
p
p
a
img
article
h1
p
p
p
a
img
section
article
h1
p
p
p
a
img
article
h1
p
p
p
a
img
aside
footer
logo
a
img
ul
li
a
li
a
li
a
li
a
li
a
li
a
li
a
li
a
li
a
li
a

EMMET - A Cool (Optional) Tool

- Emmet Demo
- Emmet Cheatsheet
- Using what we've done in class, open the provided and see how far you can get through these exercises in 10 min!

  starter-code/emmet-basics.html
  starter-code/css/emmet-theme.css

  Note: You will have to link the CSS file first!

Challenges

- Make an unordered HTML list of the following animals:
- mouse
- canary
- penguin
- salmon
- cat
- goldfish
- dog
- sheep
- parakeet
- tuna

Challenges

- Apply the following colors to the list using IDs:
- mouse "color: gray"
- canary "color: orangered"
- penguin "color: black"
- salmon "color: salmon"
- cat "color: sienna"
- goldfish "color: gold"
- dog "color: tan"
- sheep "color: steelblue"
- parakeet "color: lime"
- tuna "color: purple"

Challenges

- Add the following background colors to your existing classes:
- mammal - lavenderblush
- bird - lightgray
- fish - light yellow

- Make the mammals bold
- Make the birds italic
- Make the fish underlined

More CSS Fun

- Create a new unordered ordered list add a list item for each the following plants:

- Dogwood Tree
- Oak Tree
- Saguaro
- Kelp
- Venus Fly Trap
- Ent

More CSS Fun (Optional)

- Give all ul's a border with a width of 3 pixels, a color of plum, and a style of dotted. Also, give them a border radius of 5px.

- Give all li's a top border of 3 pixels, a color of seagreen, and a style of solid.

Questions?

Questions

Thank you!