

Not this kind of boxing.



Intro To Boxing

Not this kind either.



Intro To Boxing

This kind of boxing!

The image shows a hand-drawn box diagram on a piece of paper. The diagram is divided into four main sections, each containing a box with numbered elements. To the right of the diagram is a list of elements corresponding to the boxes. The elements are: header, header-logo, header-button, hero, hero-logo, hero-title, L1, L2, L3, L4, L5, L6, L7, L8, L9, L10, L11, L12, L13, L14, L15, L16, L17, L18, L19, L20, L21, L22, L23, L24, L25, L26, L27, L28, L29, L30, L31, L32, L33, L34, L35, L36, L37, L38, L39, L40, L41, L42, L43, L44, L45, L46, L47, L48, L49, L50, L51, L52, L53, L54, L55, L56, L57, L58, L59, L60, L61, L62, L63, L64, L65, L66, L67, L68, L69, L70, L71, L72, L73, L74, L75, L76, L77, L78, L79, L80, L81, L82, L83, L84, L85, L86, L87, L88, L89, L90, L91, L92, L93, L94, L95, L96, L97, L98, L99, L100. The diagram is a visual representation of a page layout, showing the relative positions and sizes of various elements. The list of elements is a textual representation of the same layout, providing a clear and concise way to describe the page structure.

box diagramming technique

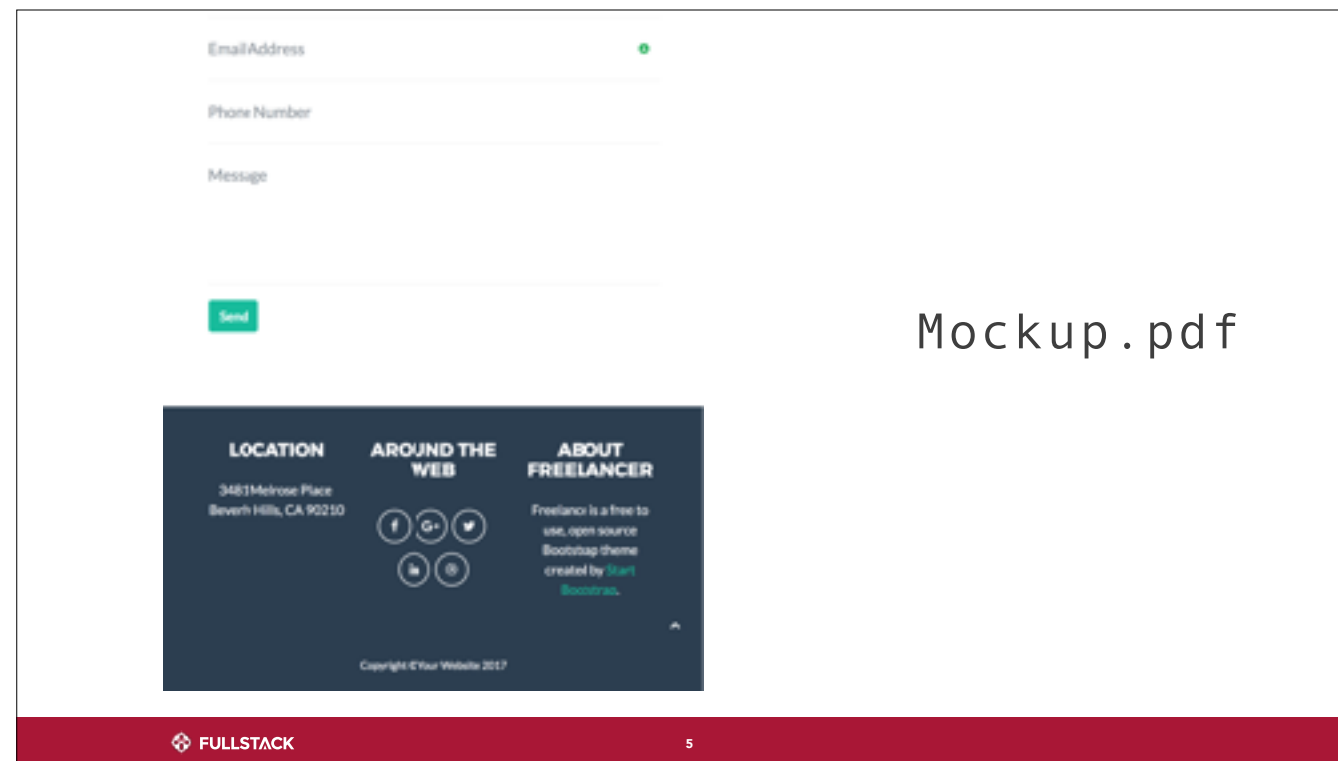
for css layout

FULLSTACK

4

INTRO TO BOXING

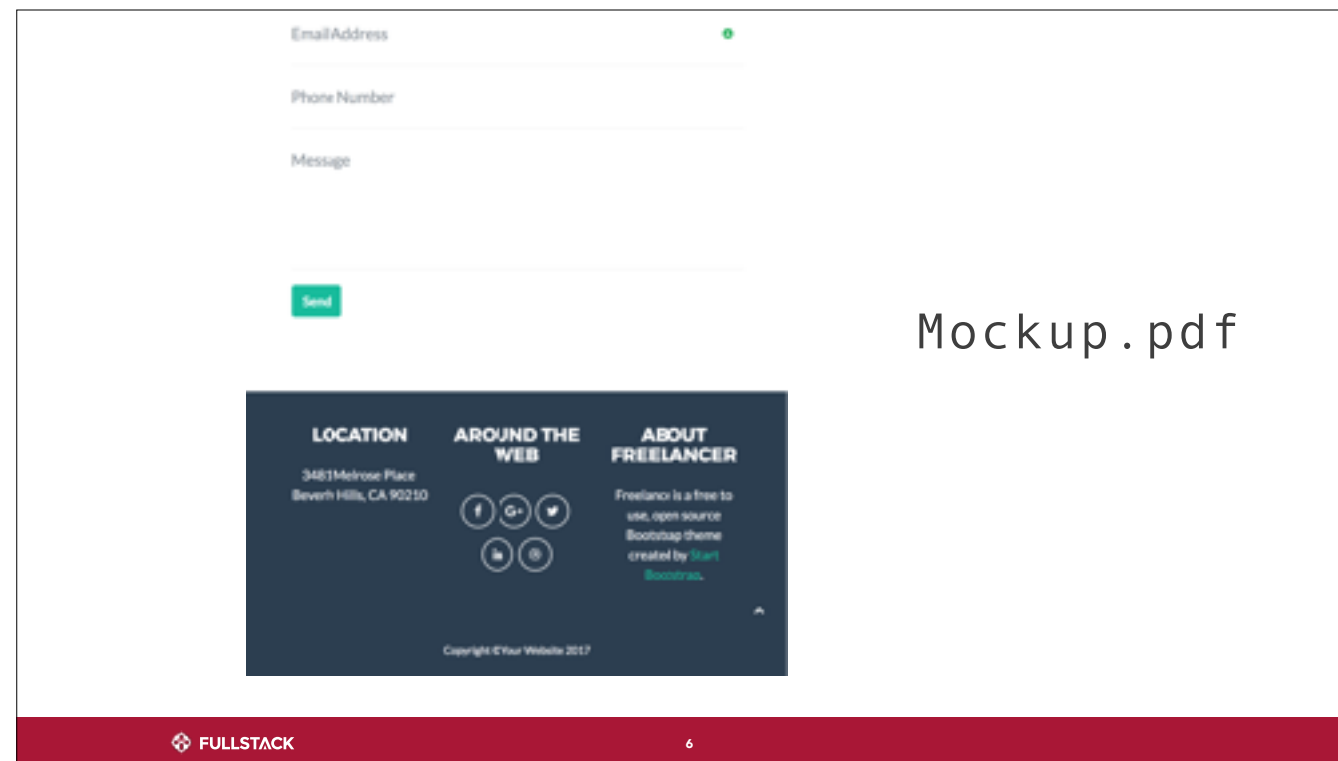
Today we're going to learn the fundamentals of how the browser layout engine positions elements on a page.



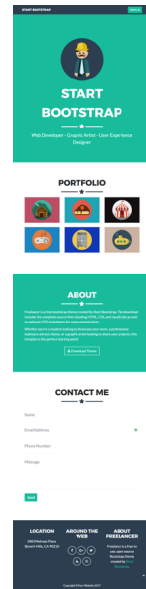
Common scenario: one or more designers deliver a mockup or composition.

This is visual specification of the application.

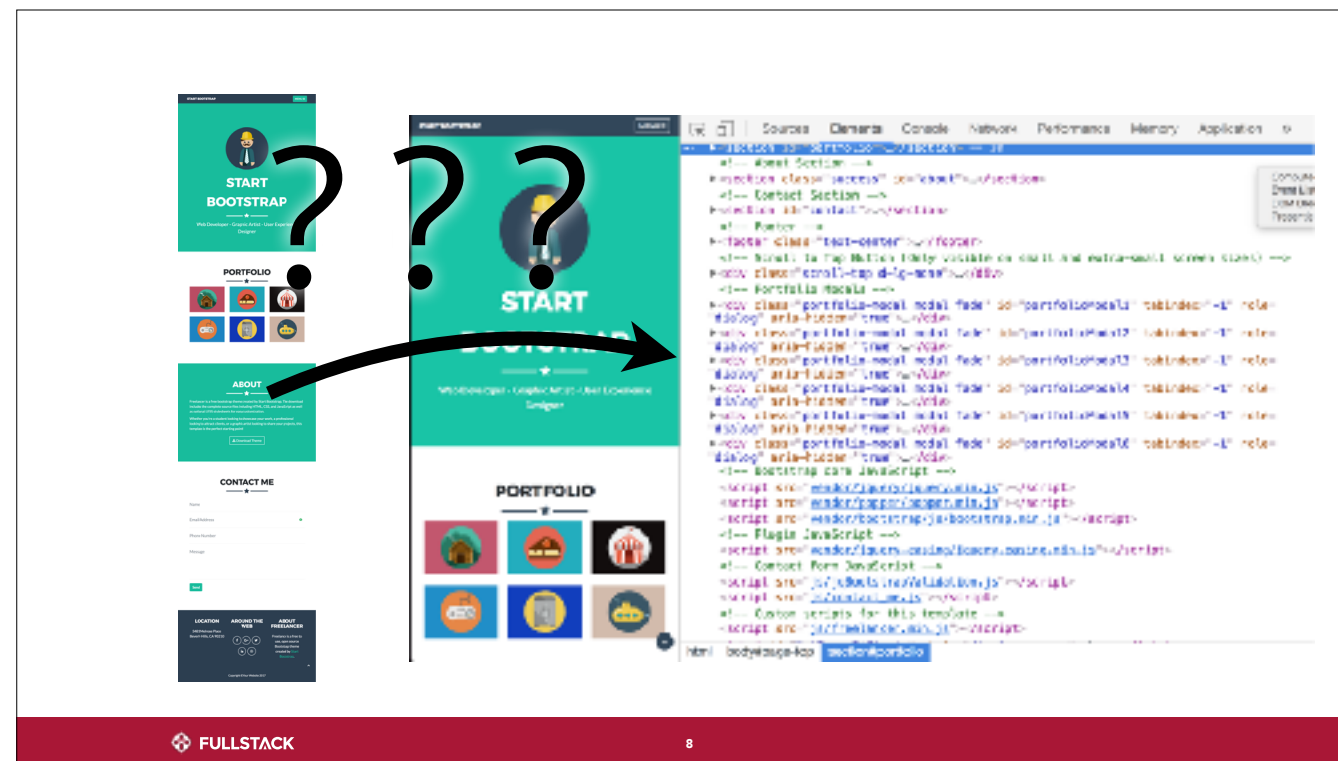
How opaque/detailed this specification is a spectrum from a flat image to a series of well structured user stories.
It might be high or low fidelity (wireframes vs full comp)



Mockup.pdf







Mockup.pdf




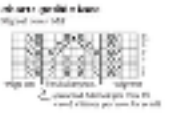

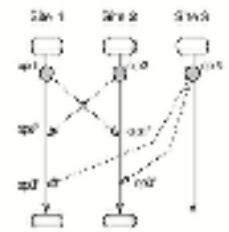
The problem to solve is:


How do we transform this visual specification into something a web browser understands?

How do we map something visual into something structural/hierarchical?

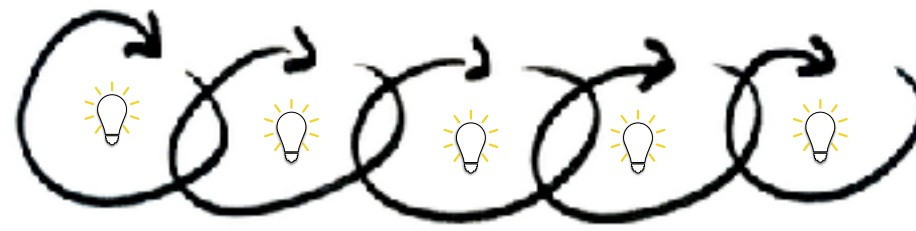
DIAGRAMS


9
INTRO TO BOXING

Diagrams! (Clockwise from top left: boxing diagram, baseball score, dancing, real-time text editing, a digram describing diagrams, and a knitting diagram.)

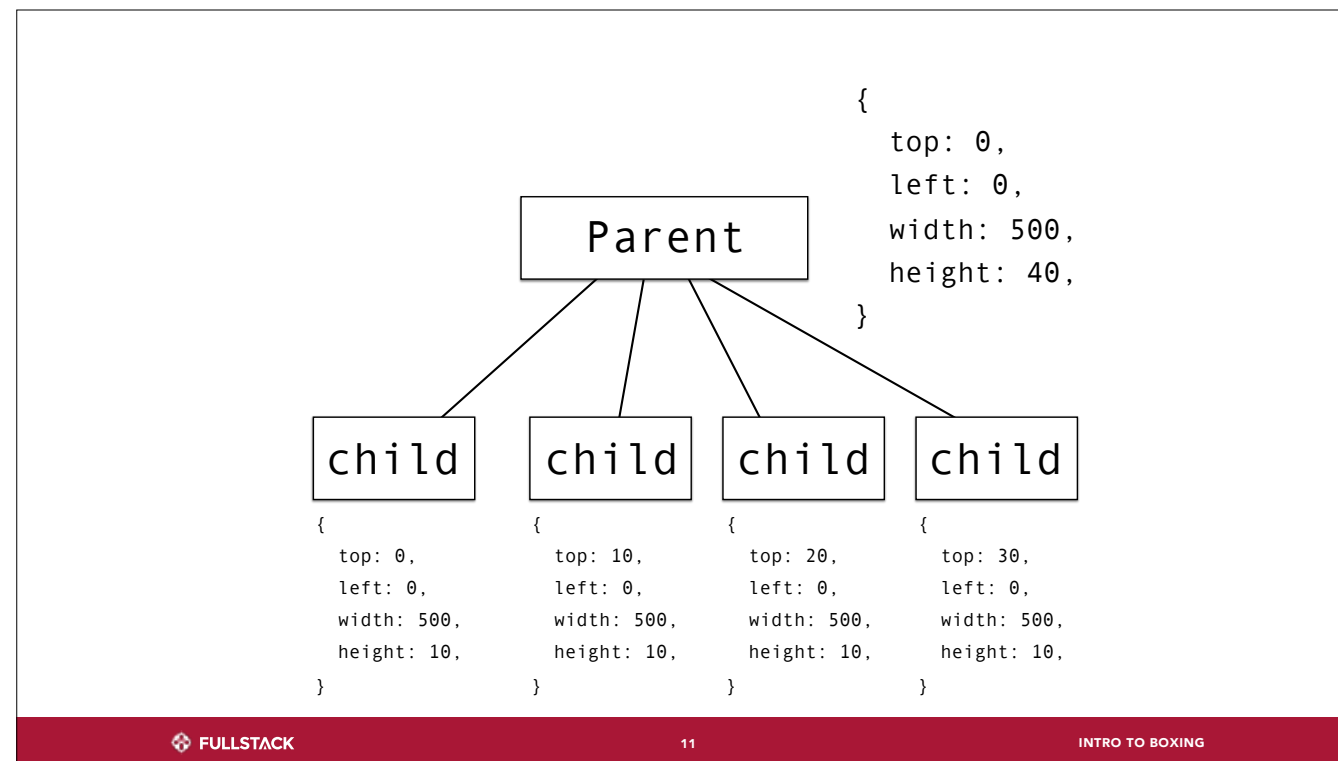
Diagrams fix cognition into an externalized form.



Diagramming Facilitates Iterative Cognition

Diagrams are not just for reading. The process of drawing diagrams aids cognition. A series a small thoughts/insights add up to the full idea.

“Bring brick, not a cathedral.”

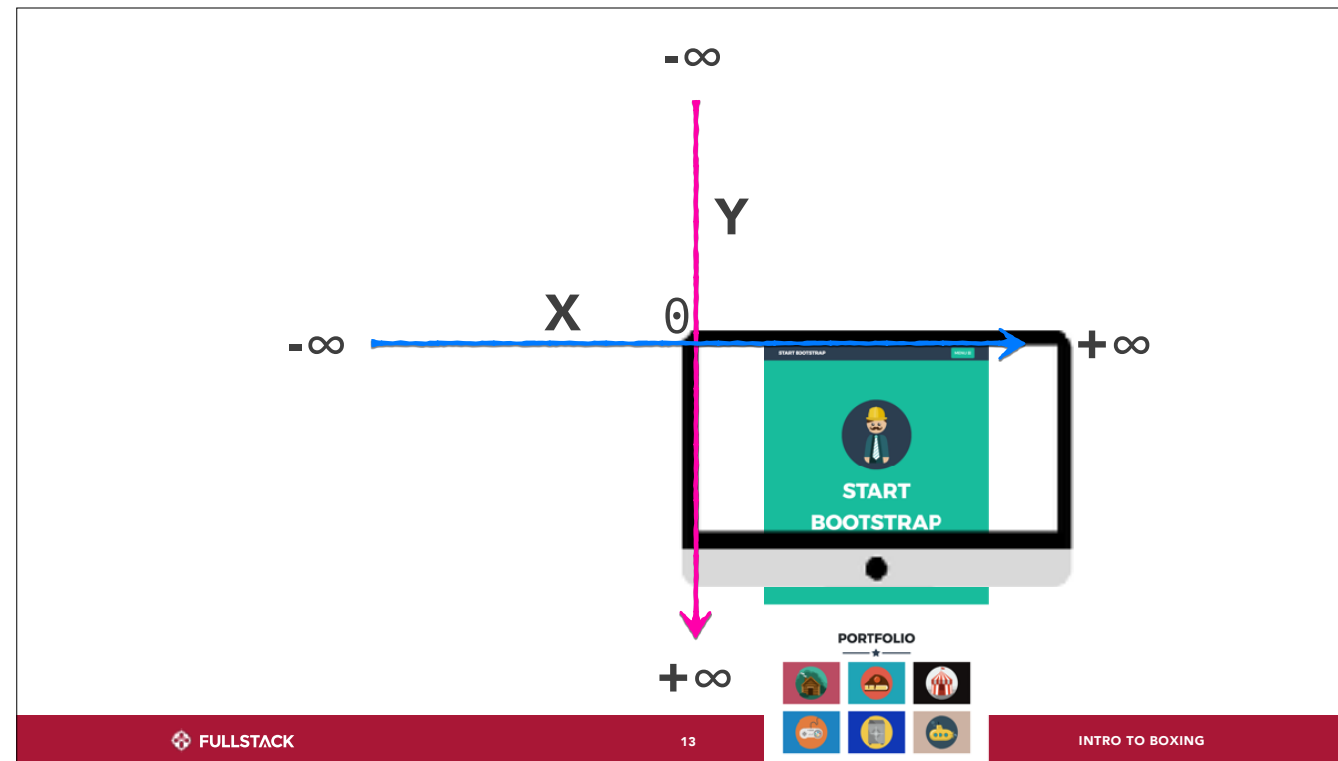


So with these nodes, we might get geometries that look like this.

There are rules that determine how to calculate those geometries.

Before we can calculate geometry, we must know the shape of the universe.

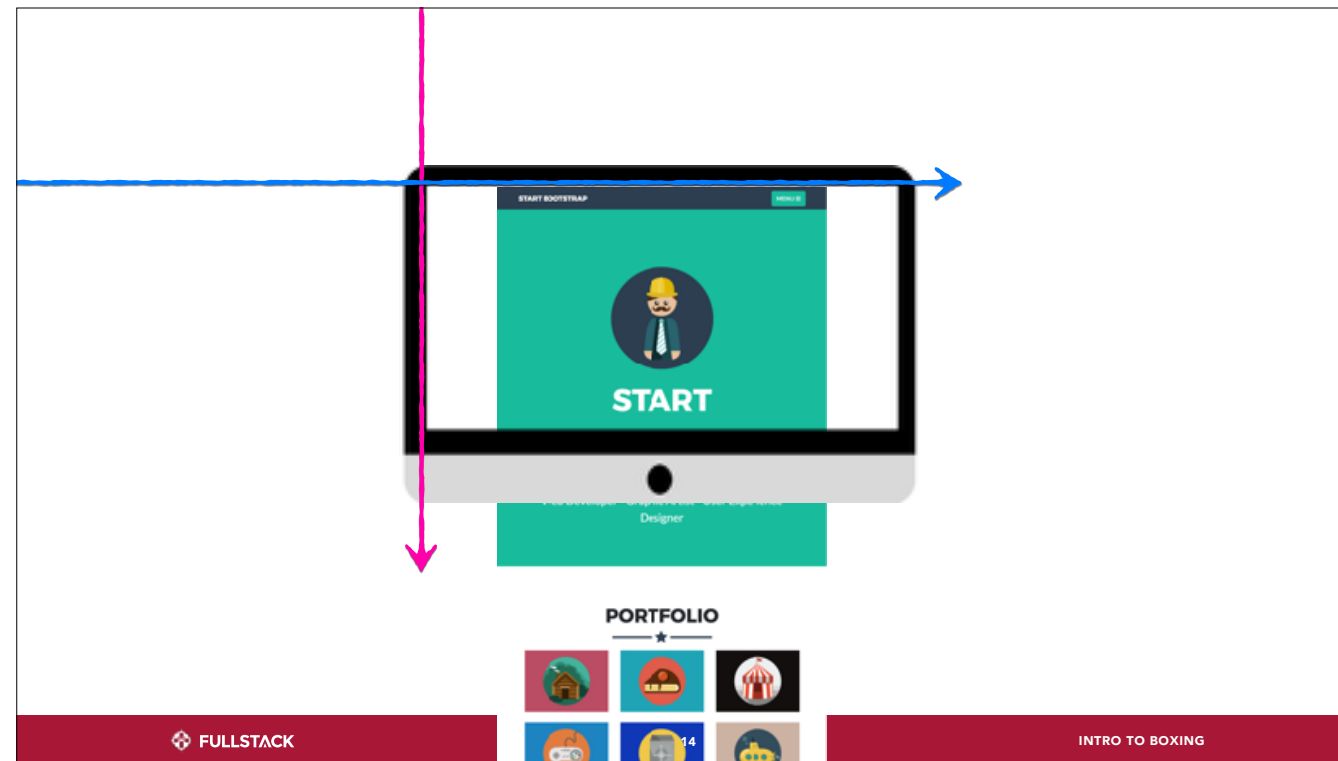
What is the Shape of the ^{CSS} Universe?



The css universe is centered at X0, Y0

The Y axis is upside-down compared to a regular geometric plane.

(There is a Z axis... but we won't consider that for now.)



This plane is not the same plane as the screen. As you scroll, the 0 point moves outside the viewport.

How Tall/Wide Is This Box?

width (x)



height (y)

As a default, the answer depends on the content we place inside the box.

Measuring Things

Relative

50%

10rem

3em

Absolute

50px

10pt

3in

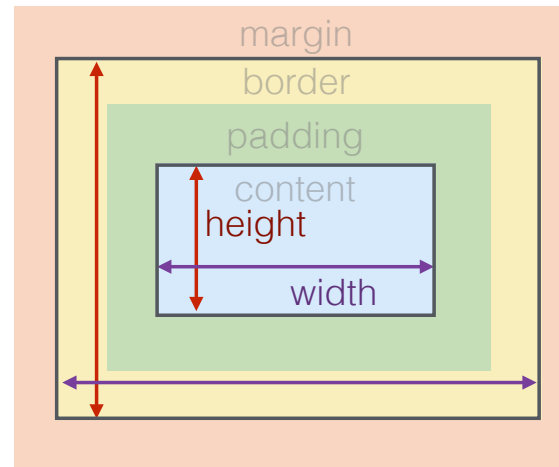
There are a lot of ways to measure in the CSS universe.

Some measurements are absolute, and others are different depending on where in the document tree the measurement is applied.

Relative measures are useful for sizing an element relative to it's parent (%), or relative to the font-size of a document (rem).

BOX MODEL

```
border box  
{  
  box-sizing: border-box;  
}  
content box  
{  
  box-sizing: content-box;  
}
```

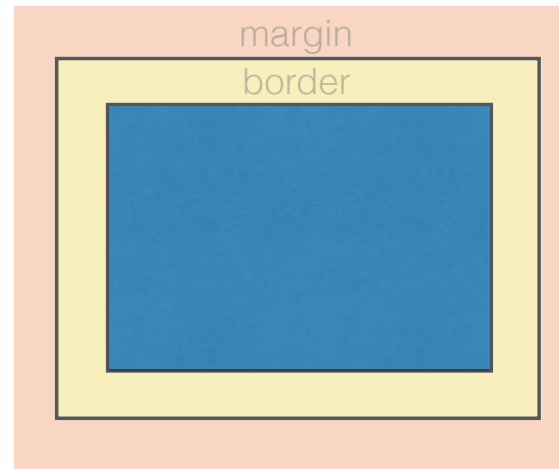


The box has a few 'layers', margin/border/padding/content

There are a few ways to calculate the size of a box.

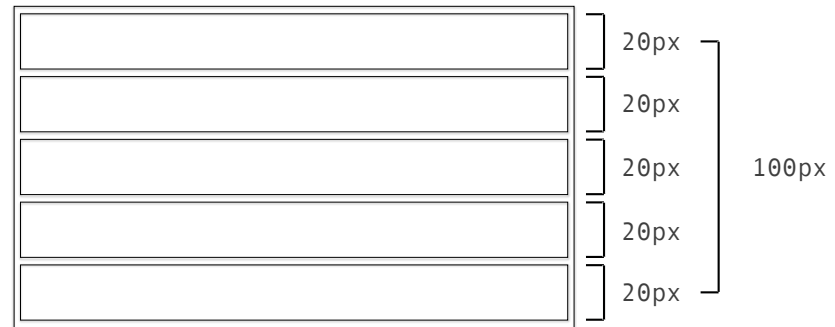
BOX MODEL

background

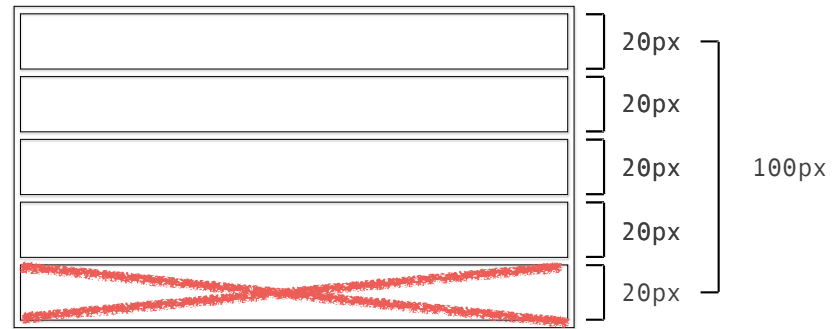


parent.height = \sum children.height

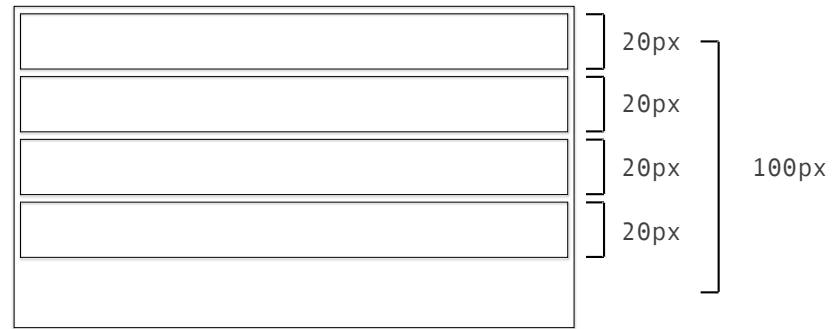
Boxes shrink and grow to fit their contents.



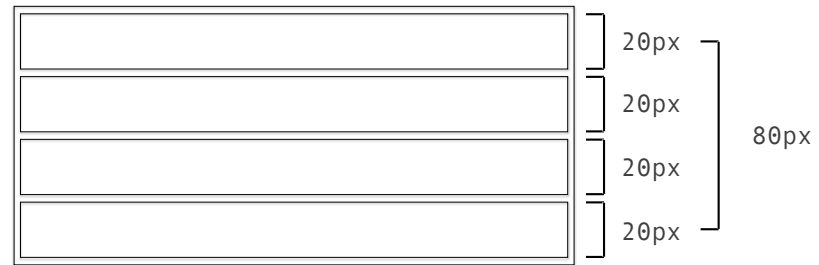
`parent.height = Σ children.height`



`parent.height = Σ children.height`

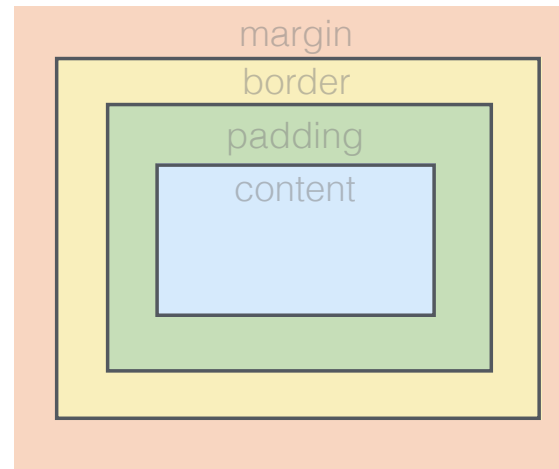


`parent.height = Σ children.height`



`parent.height = Σ children.height`

BOX MODEL



And remember, every box follows the box model.

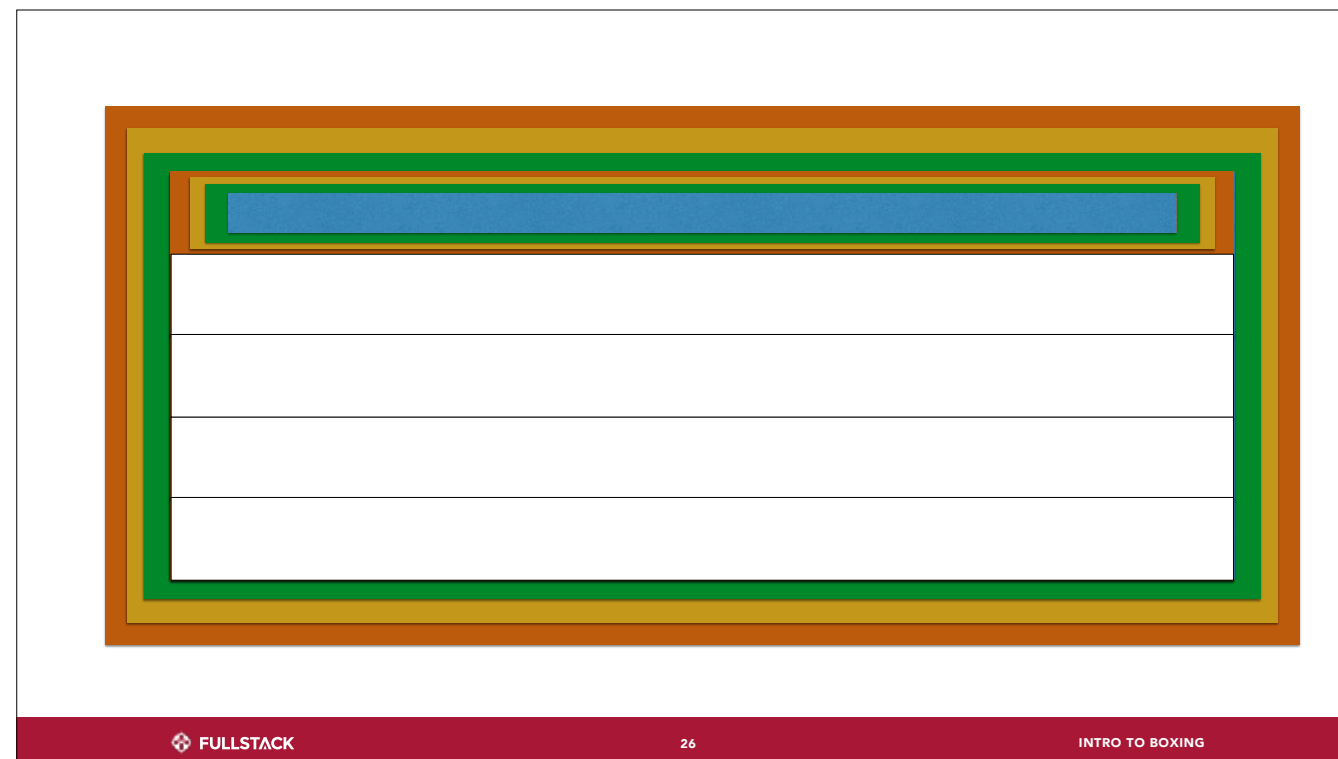
A diagram illustrating a nested rectangular structure, likely representing a container or a box. The structure consists of three concentric rectangles:

- An outermost orange border.
- A middle yellow border.
- An innermost green border.

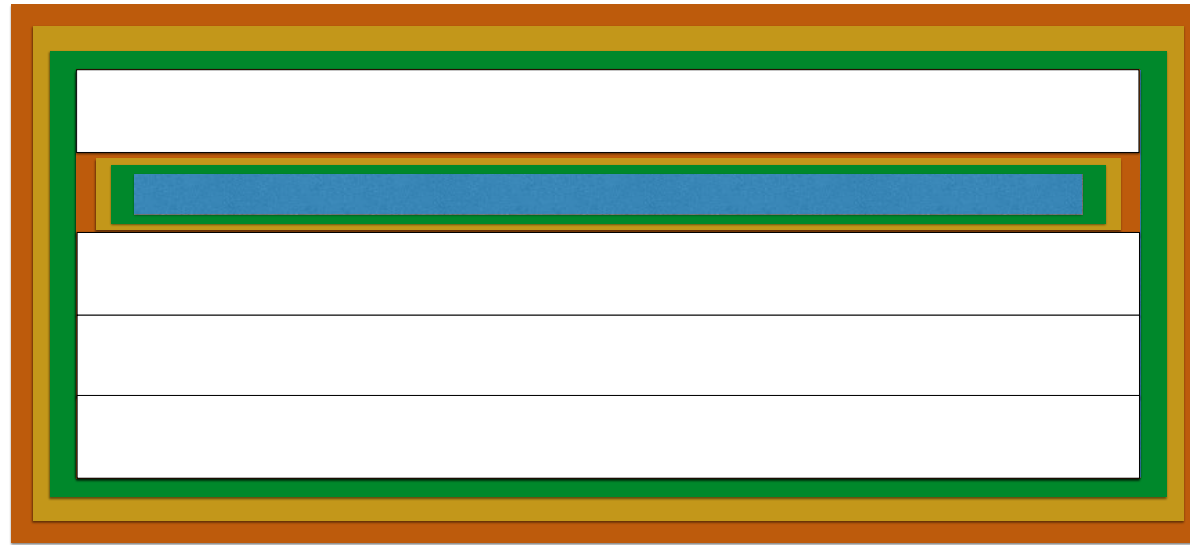
The innermost green-bordered rectangle is divided into five horizontal white sections by thin black lines, suggesting a list or a series of items.

FULLSTACK 25 INTRO TO BOXING

The parent



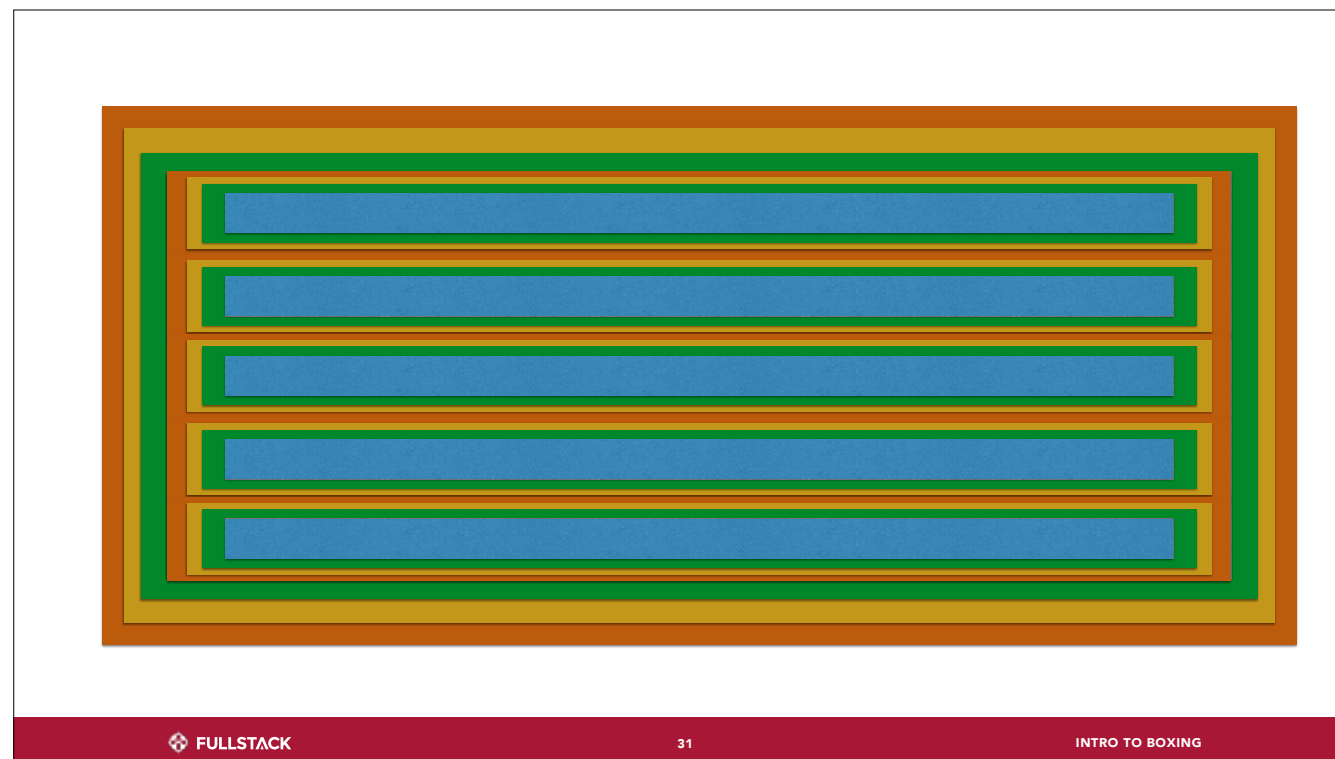
And every child.











Boxes all the way down.