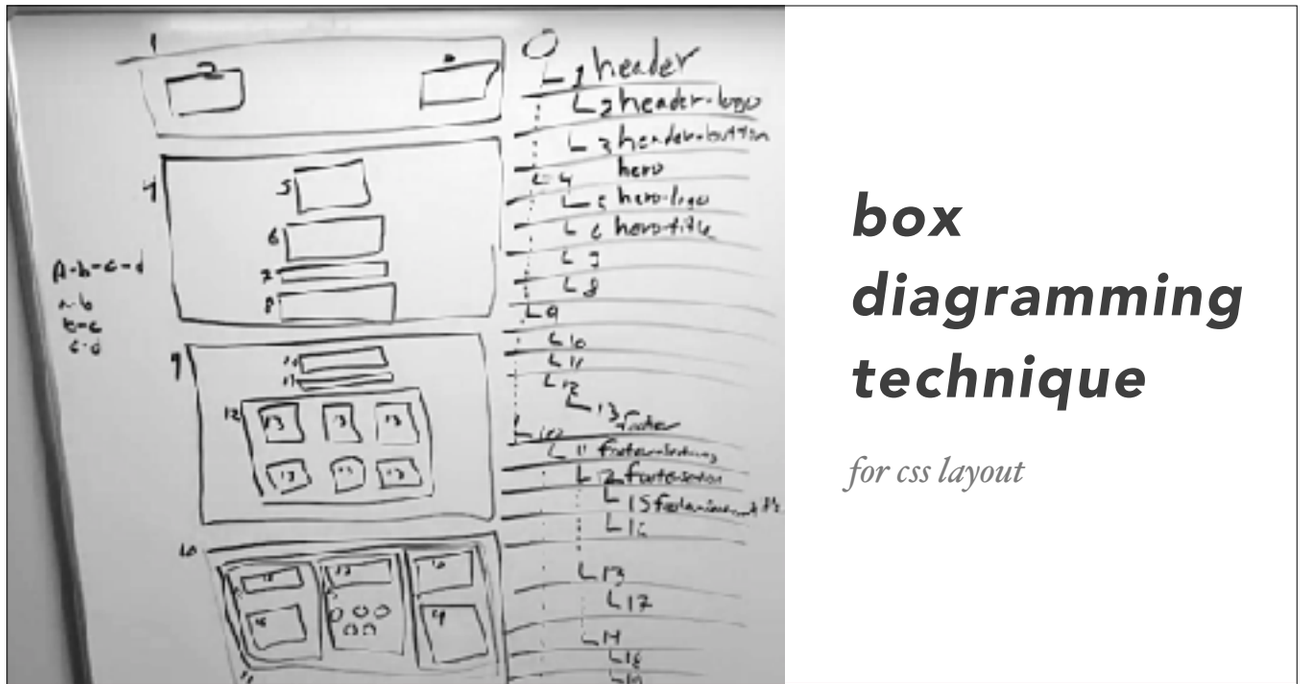
Not this kind of boxing.

Intro To Boxing

Not this kind either.

Intro To Boxing

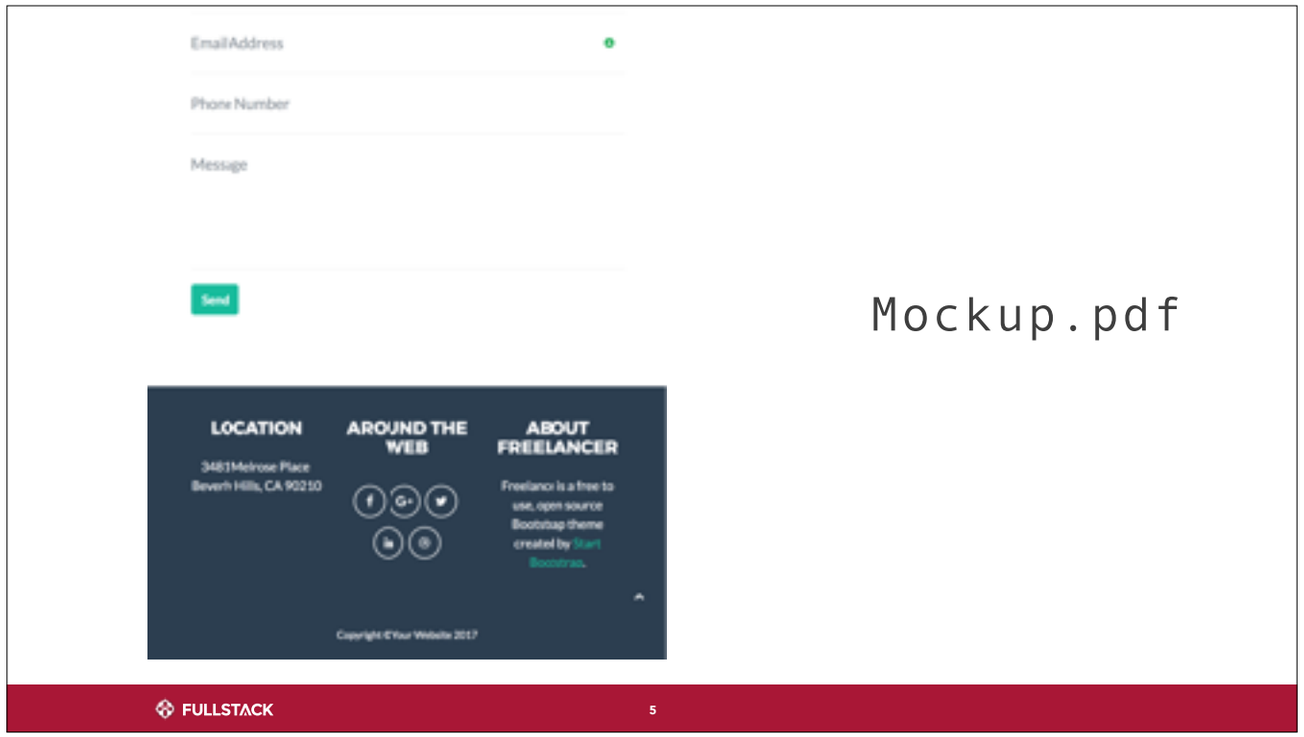
This kind of boxing!



box diagramming technique

for css layout

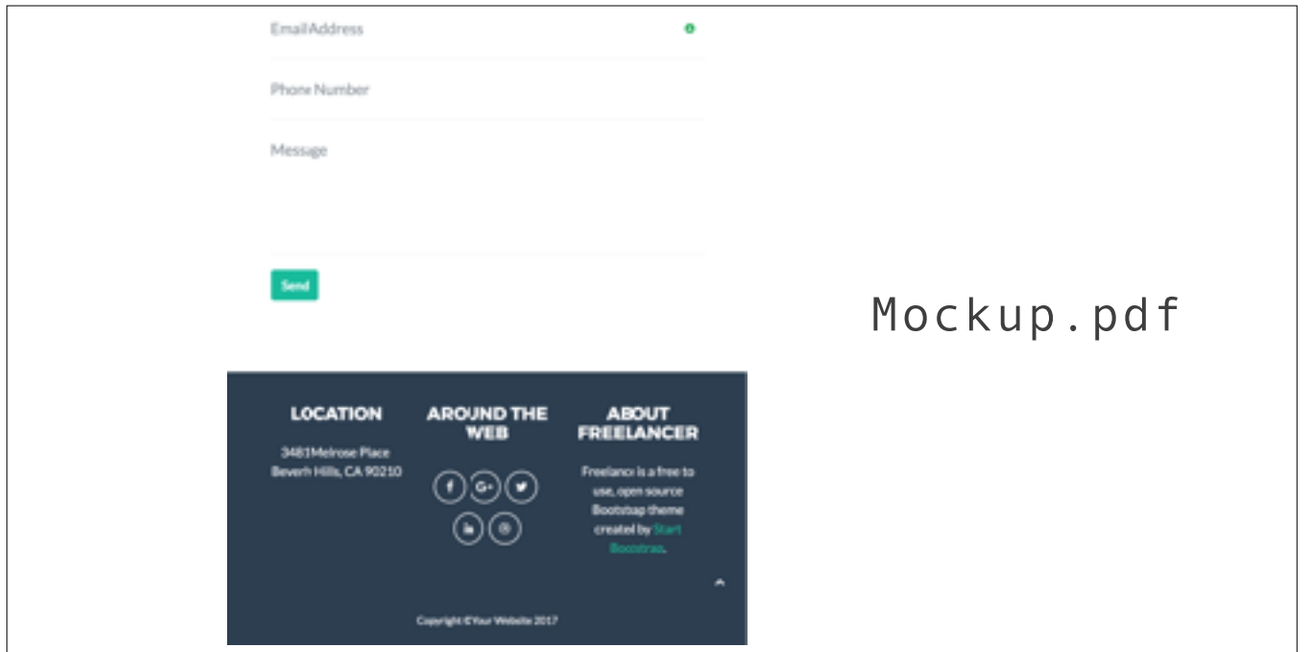
Today we're going to learn the fundamentals of how the browser layout engine positions elements on a page.



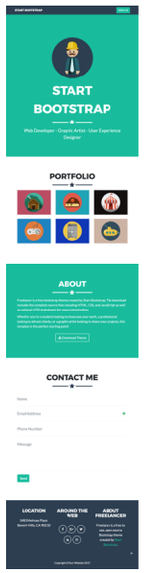
Common scenario: one or more designers deliver a mockup or composition.

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)


Email Address
Phone Number
Message
Send
LOCATION
3481 Melrose Place
Beverly Hills, CA 90210
AROUND THE WEB
ABOUT FREELANCER
Freelance is a free to use, open source Bootstrap theme created by Start Bootstrap.
Copyright © Your Website 2017

Mockup.pdf

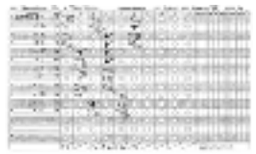

START
BOOTSTRAP
Web Developer - Graphic Artist - User Experience Designer
PORTFOLIO
ABOUT
CONTACT ME
LOCATION
AROUND THE WEB
ABOUT FREELANCER

Mockup.pdf

The problem to solve is:

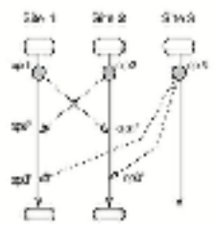
How do we transform this visual specification into something a web browser understands?

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

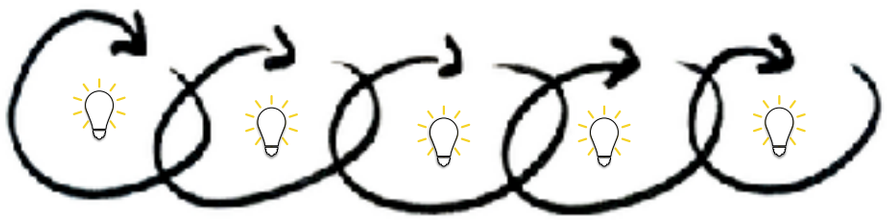

DIAGRAMS





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

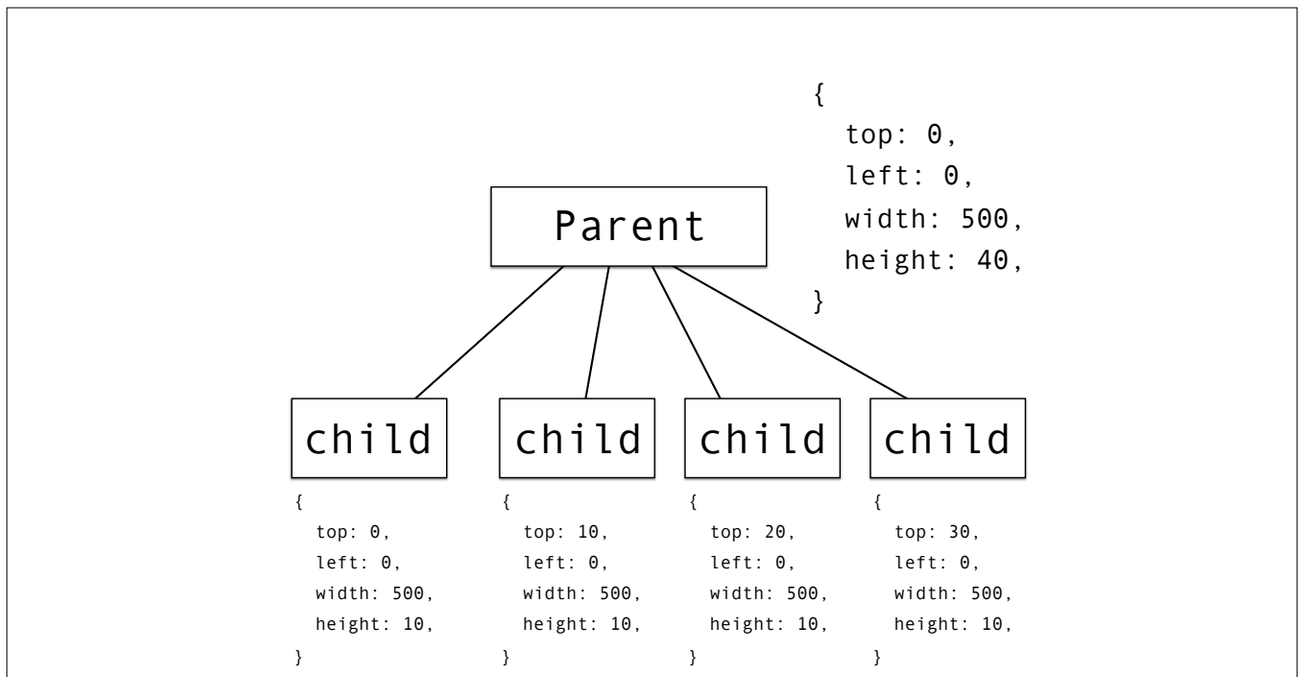
Diagrams fix cognition into an externalized form.



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.”

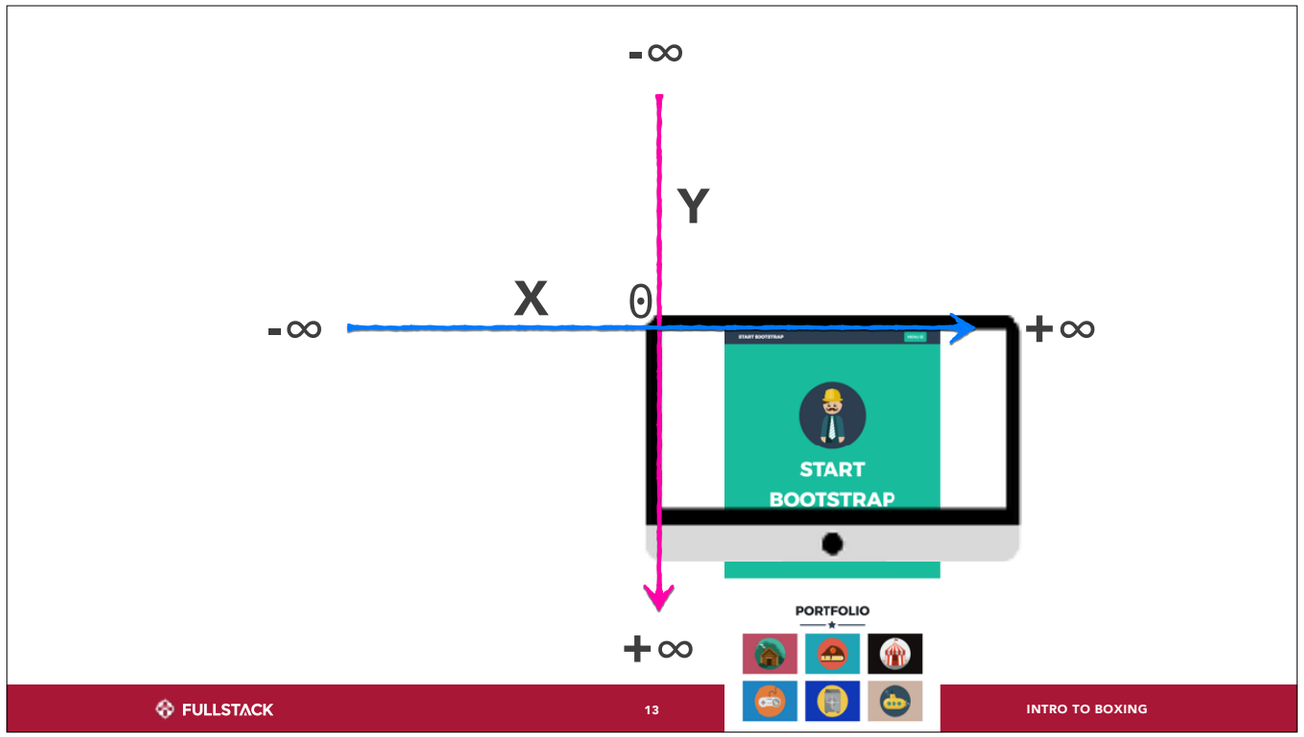


So with these nodes, we might get geometries that look like this.

There are rules that determine how to calculate those geometries.

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

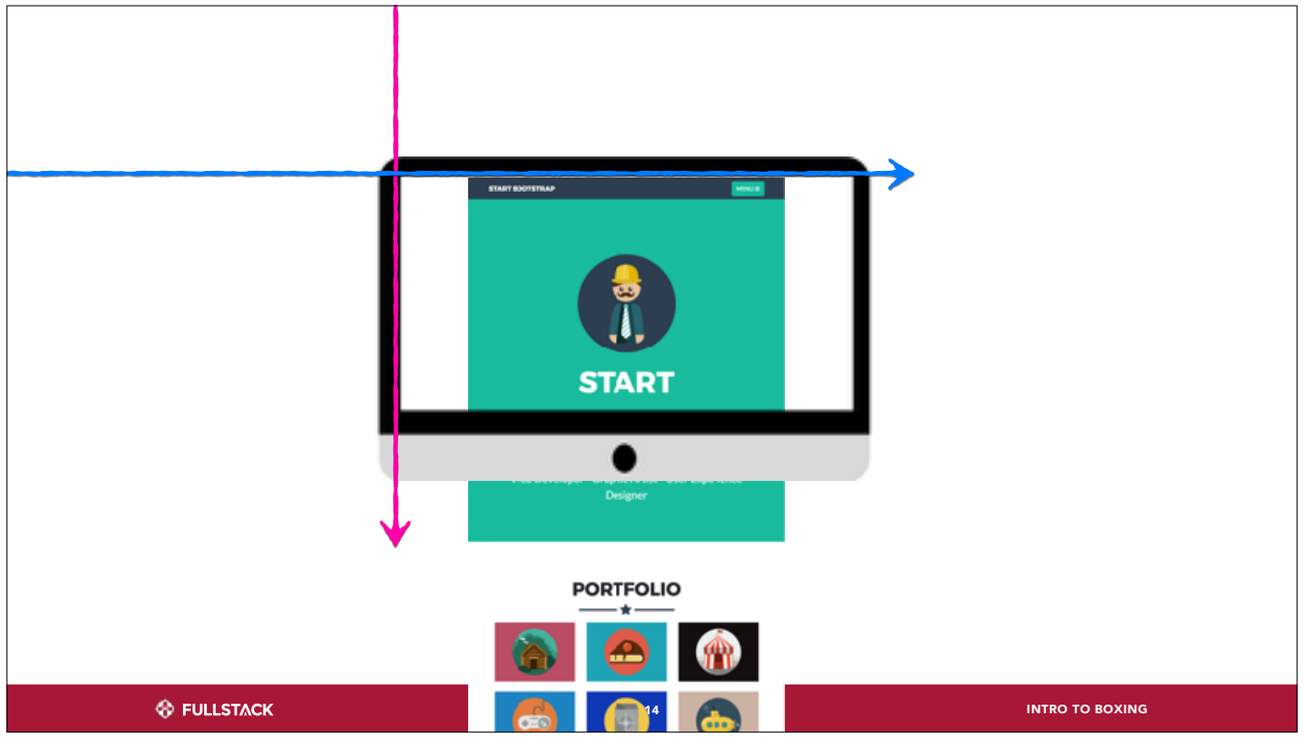
What is the Shape of the CSS Universe?



The css universe is centered at X0, Y0

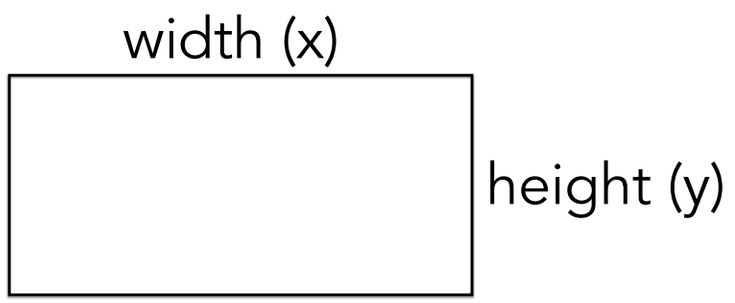
The Y axis is upside-down compared to a regular geometric plane.

(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?



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

Measuring Things

Relative	Absolute
`50%`	`50px`
`10rem`	`10pt`
`3em`	`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).

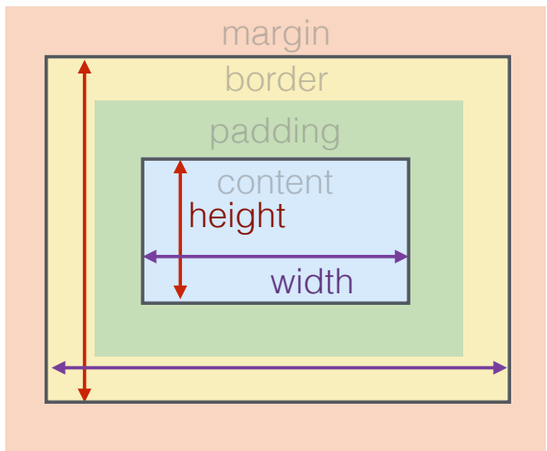
BOX MODEL

border box

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

content box

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

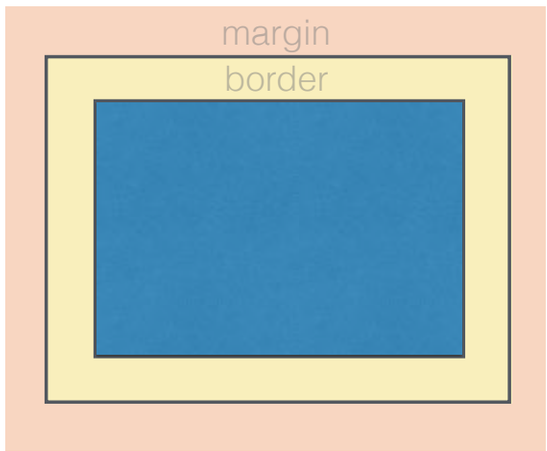


The box has a few ‘layers’, margin/border/padding/content

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

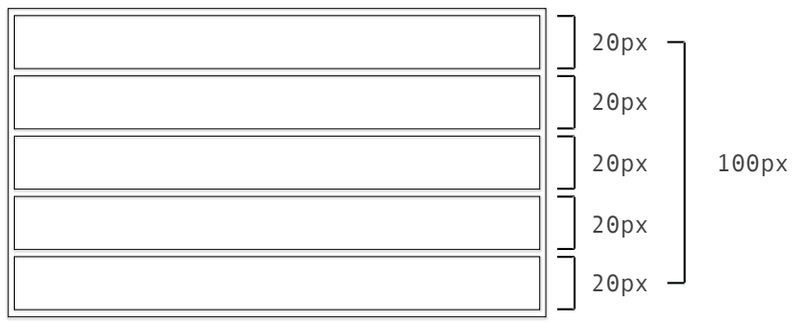
BOX MODEL

background

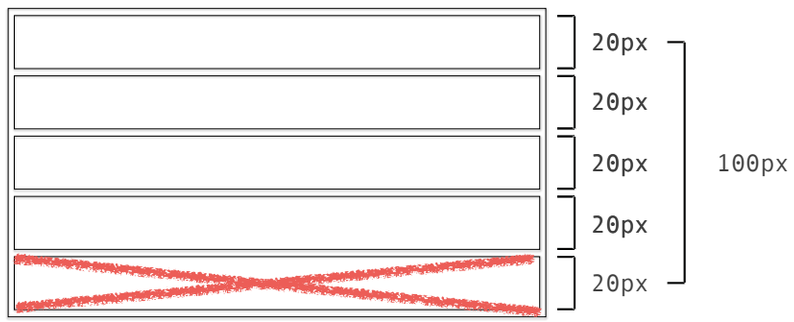


parent.height = Σ children.height

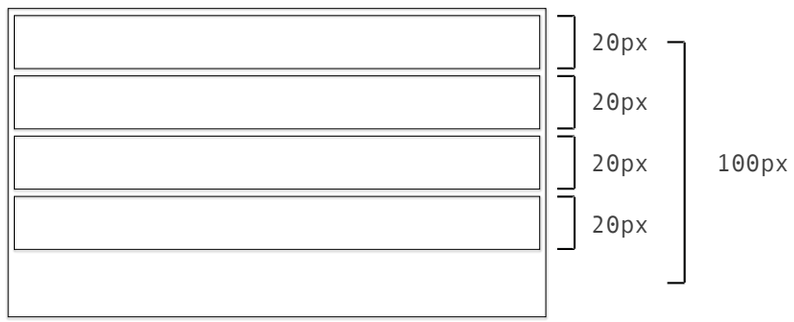
Boxes shrink and grow to fit their contents.



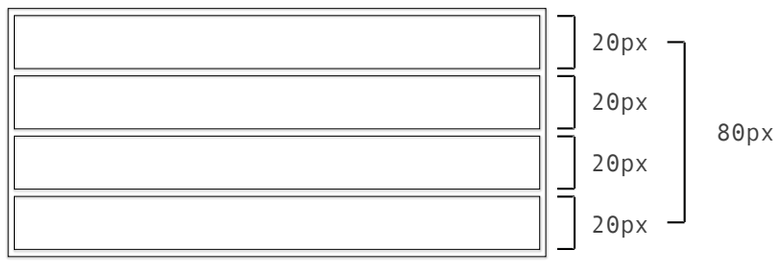
parent.height = Σ children.height



parent.height = Σ children.height

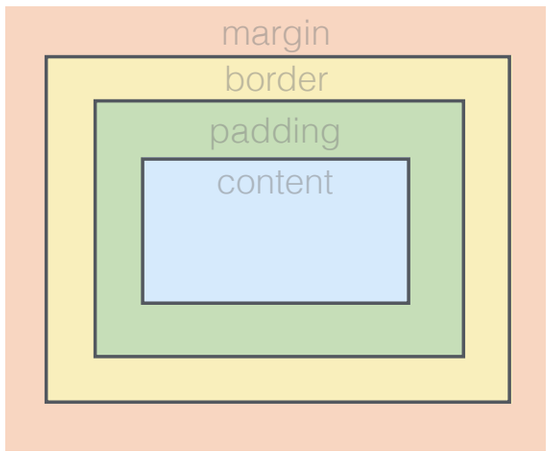


parent.height = Σ children.height



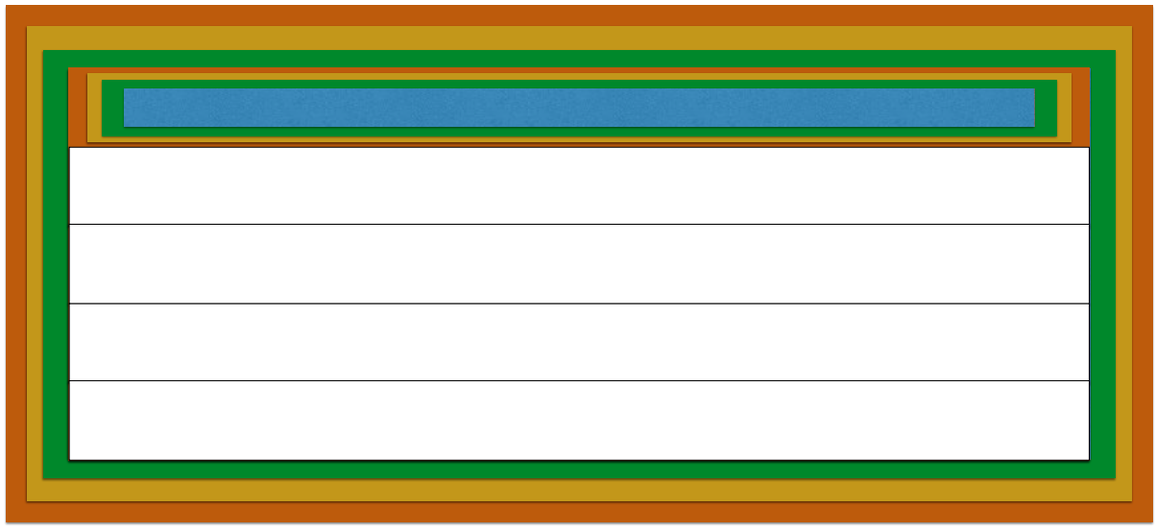
parent.height = Σ children.height

BOX MODEL

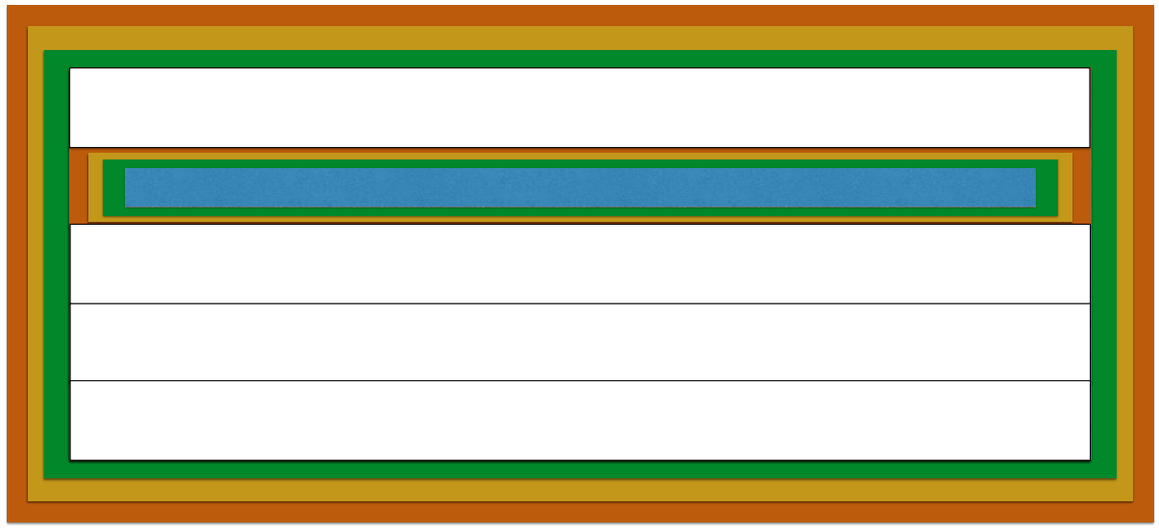


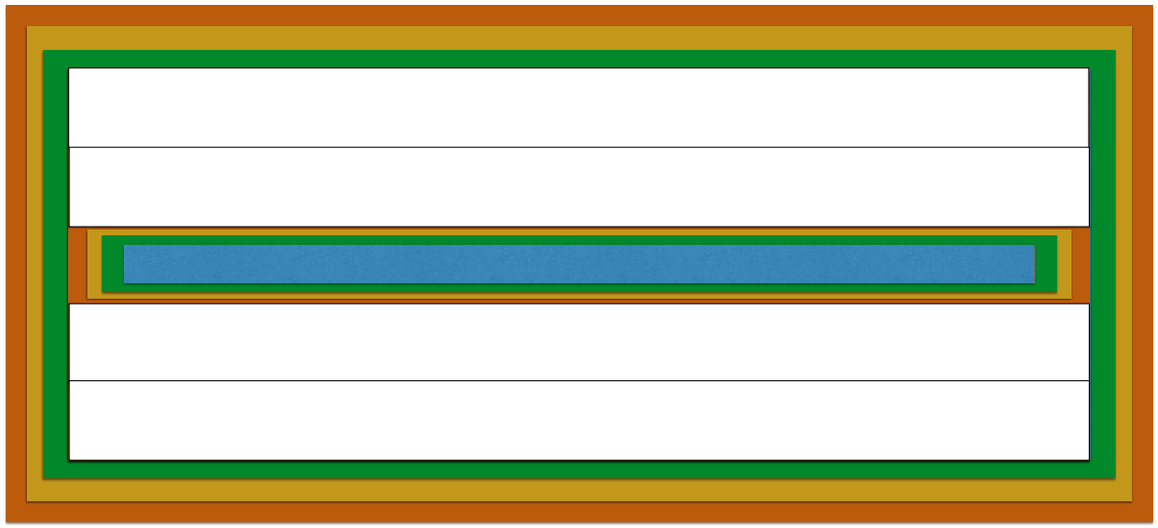
And remember, every box follows the box model.

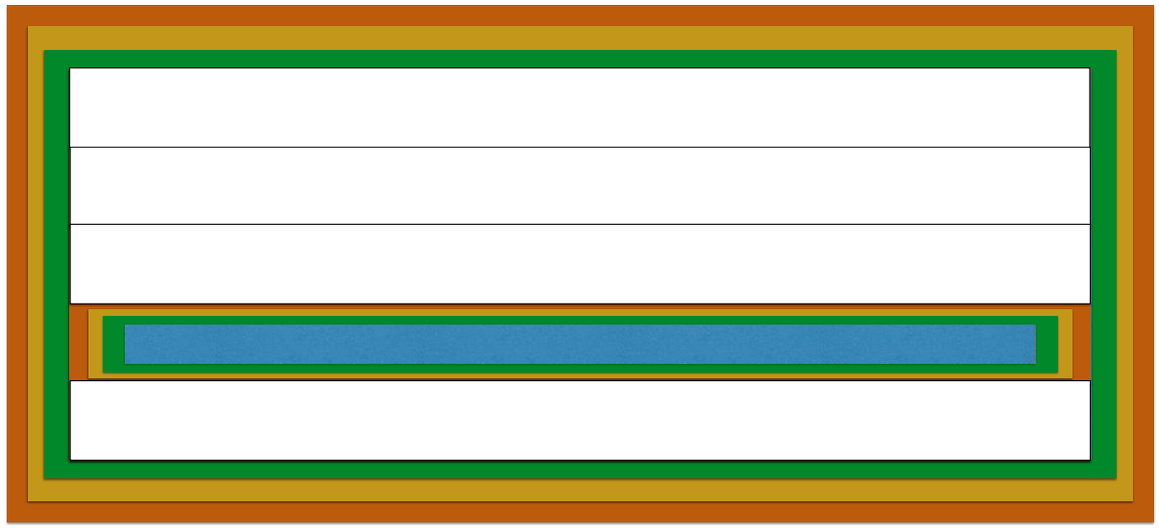
The parent

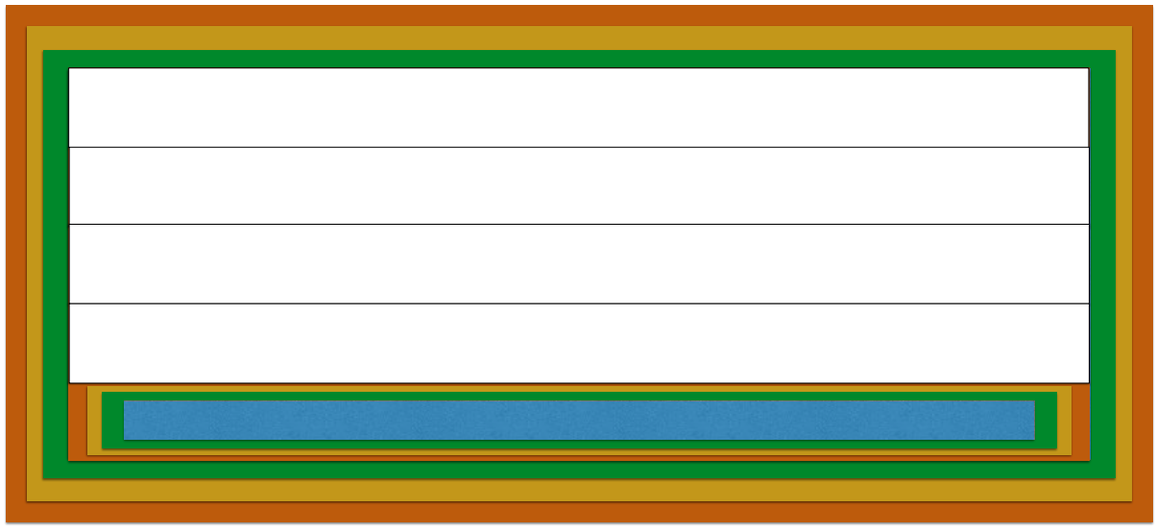


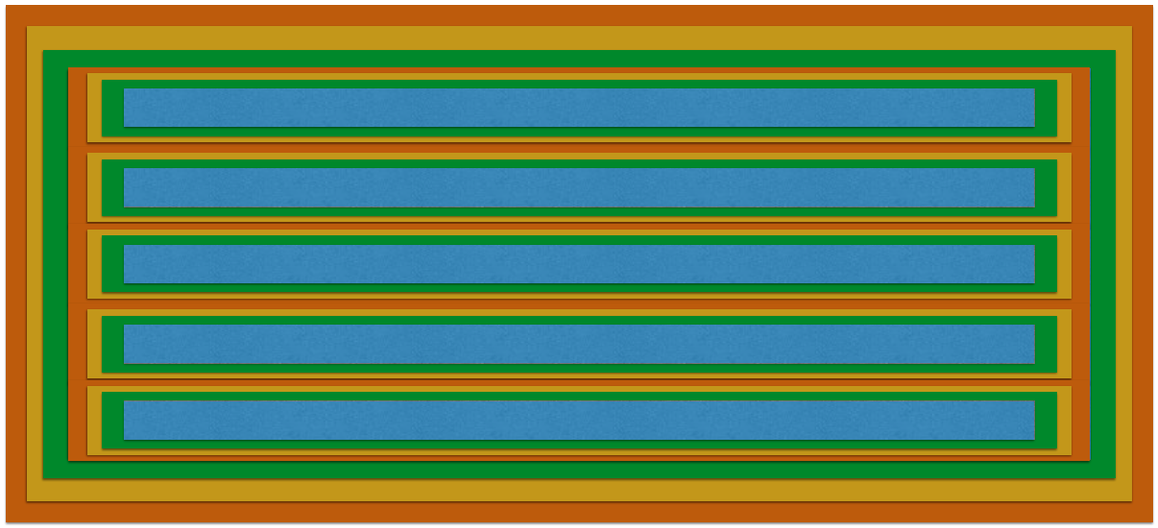
And every child.











Boxes all the way down.