Advanced CSS

Mufaro Makiwa

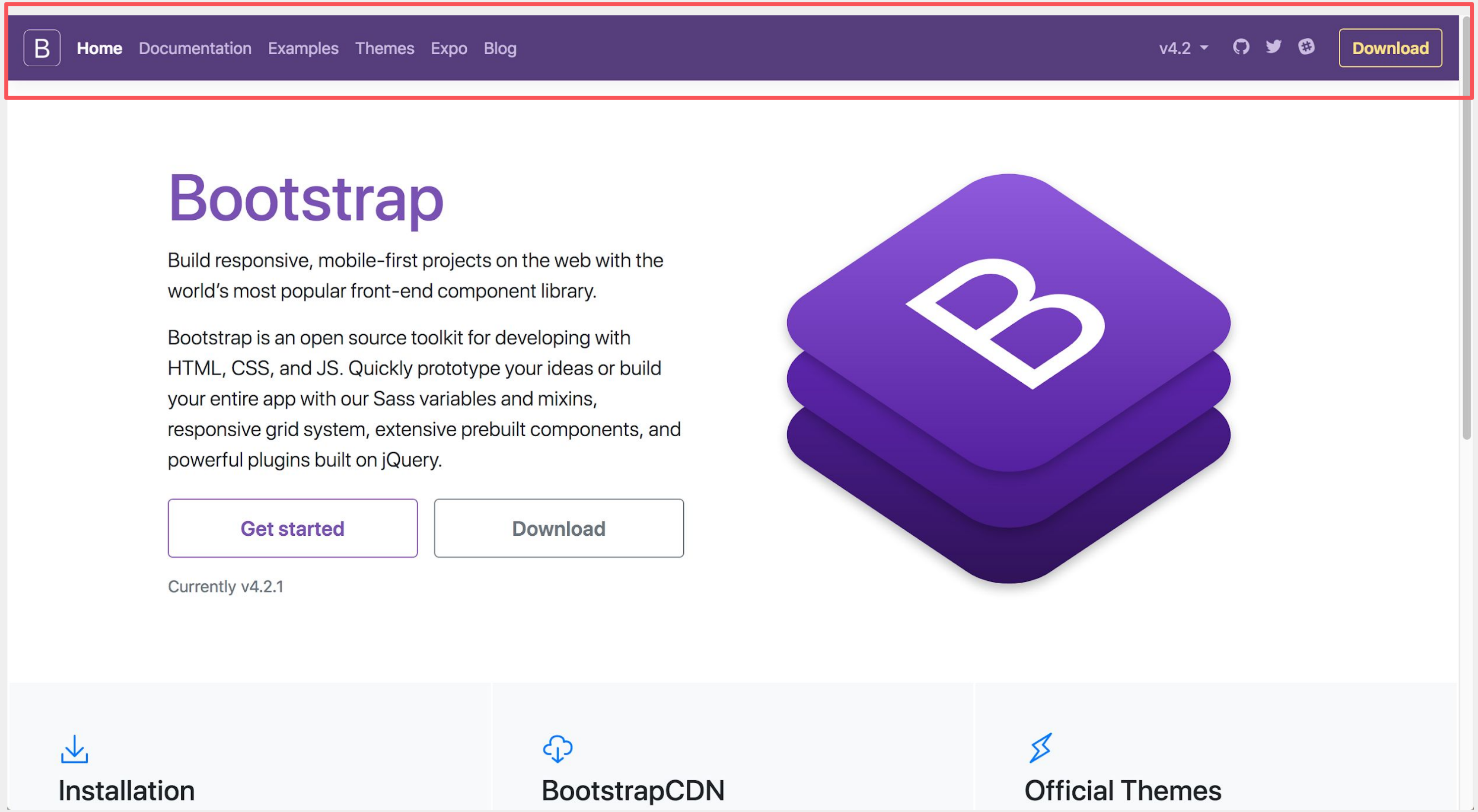
CSS

(cascading style sheets)

describes how our HTML boxes look

what are examples of things that CSS can do?

so far...



layouts

Bootstrap

Build responsive, mobile-first projects on the web with the world's most popular front-end component library.

Bootstrap is an open source toolkit for developing with HTML, CSS, and JS. Quickly prototype your ideas or build your entire app with our Sass variables and mixins, responsive grid system, extensive prebuilt components, and powerful plugins built on jQuery.

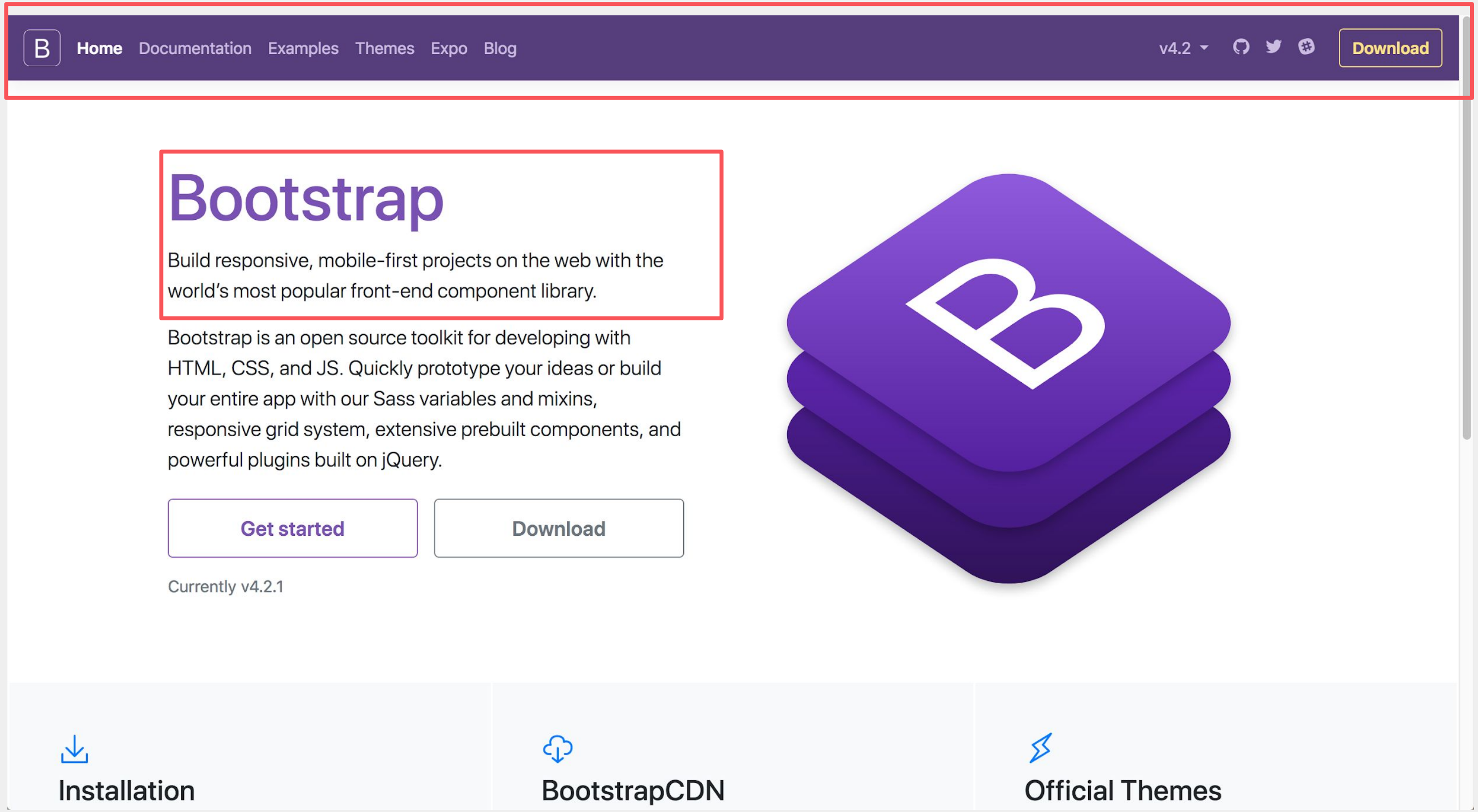
Currently v4.2.1

Installation

BootstrapCDN

Official Themes

so far...



Bootstrap

Build responsive, mobile-first projects on the web with the world's most popular front-end component library.

Bootstrap is an open source toolkit for developing with HTML, CSS, and JS. Quickly prototype your ideas or build your entire app with our Sass variables and mixins, responsive grid system, extensive prebuilt components, and powerful plugins built on jQuery.

Currently v4.2.1

Installation

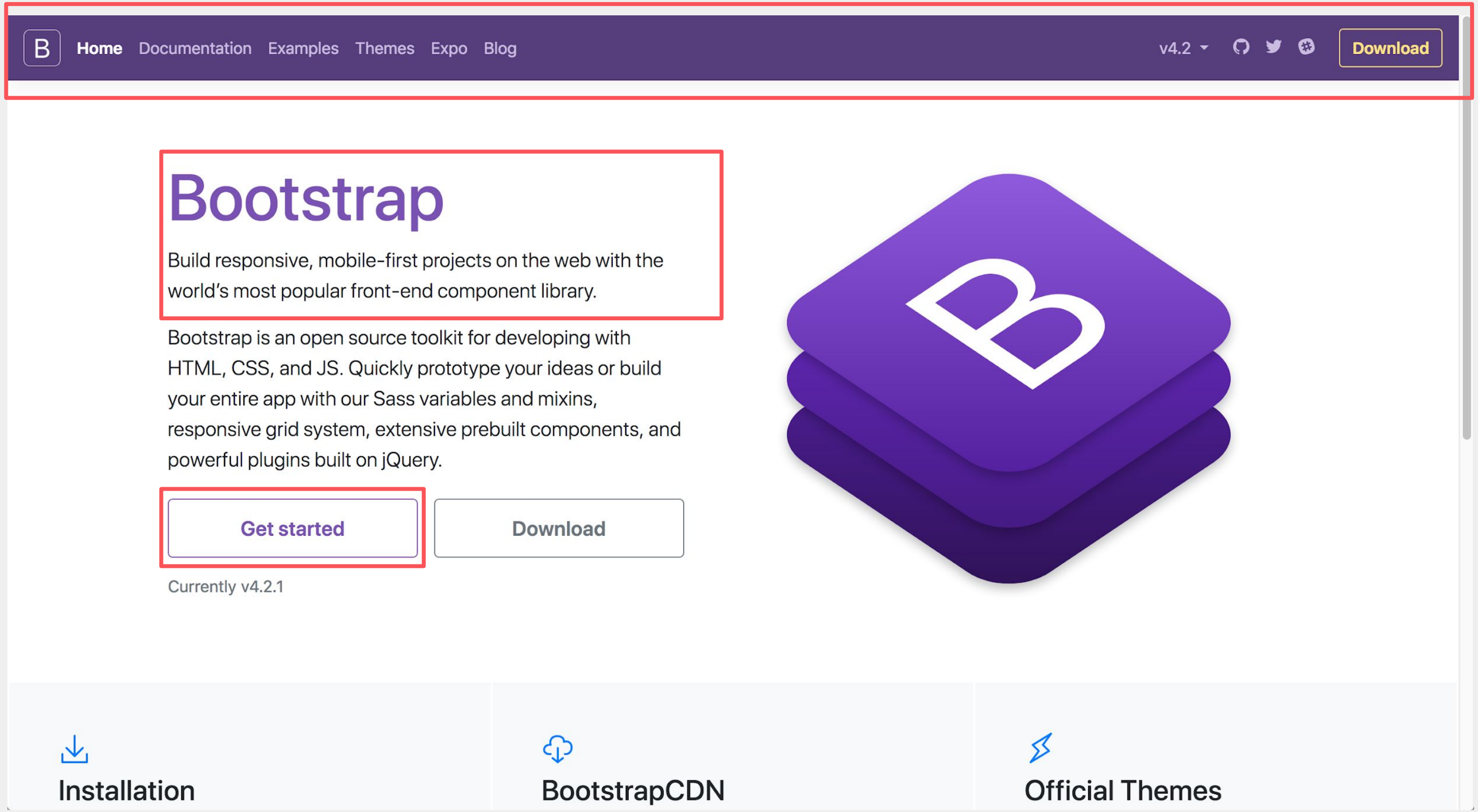
BootstrapCDN

Official Themes

layouts

fonts
text styles
colors

so far...



B Home Documentation Examples Themes Expo Blog v4.2 Download

Bootstrap

Build responsive, mobile-first projects on the web with the world's most popular front-end component library.

Bootstrap is an open source toolkit for developing with HTML, CSS, and JS. Quickly prototype your ideas or build your entire app with our Sass variables and mixins, responsive grid system, extensive prebuilt components, and powerful plugins built on jQuery.

Get started Download

Currently v4.2.1

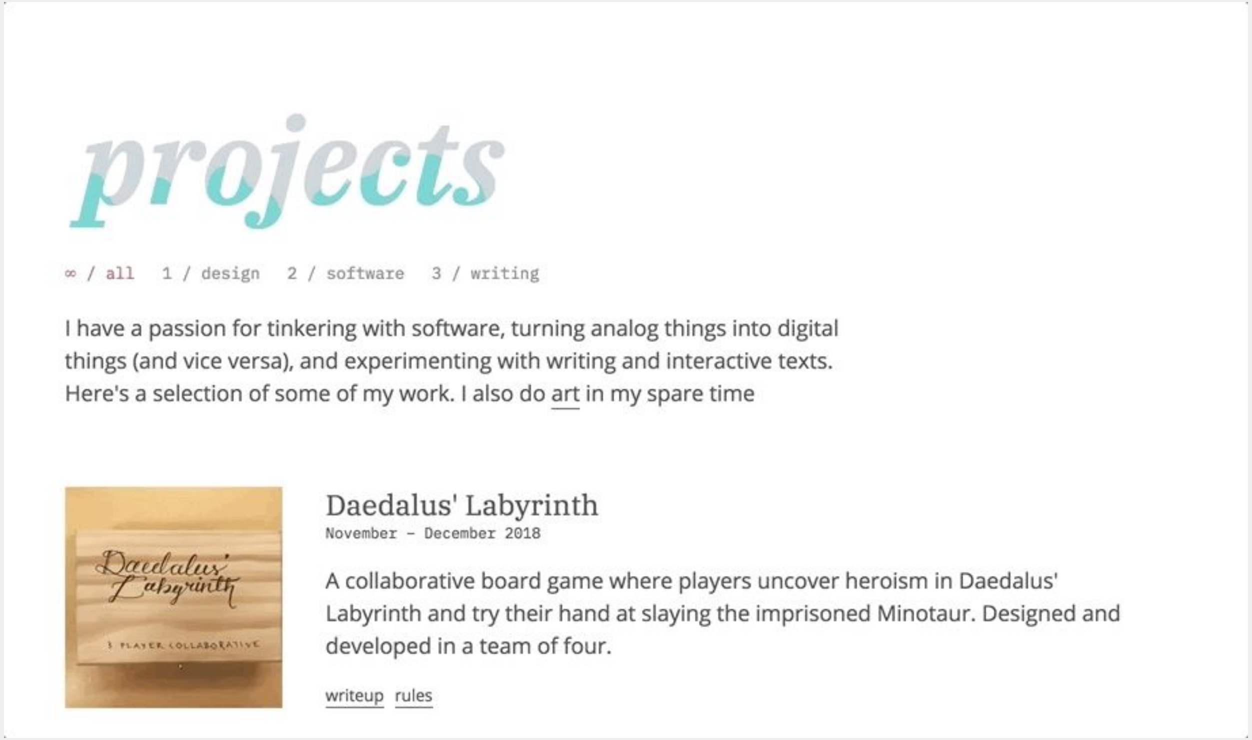
Installation BootstrapCDN Official Themes

layouts

fonts
text styles
colors

box styles
borders

but we can do so much more



projects

∞ / all   1 / design   2 / software   3 / writing

I have a passion for tinkering with software, turning analog things into digital things (and vice versa), and experimenting with writing and interactive texts. Here's a selection of some of my work. I also do art in my spare time

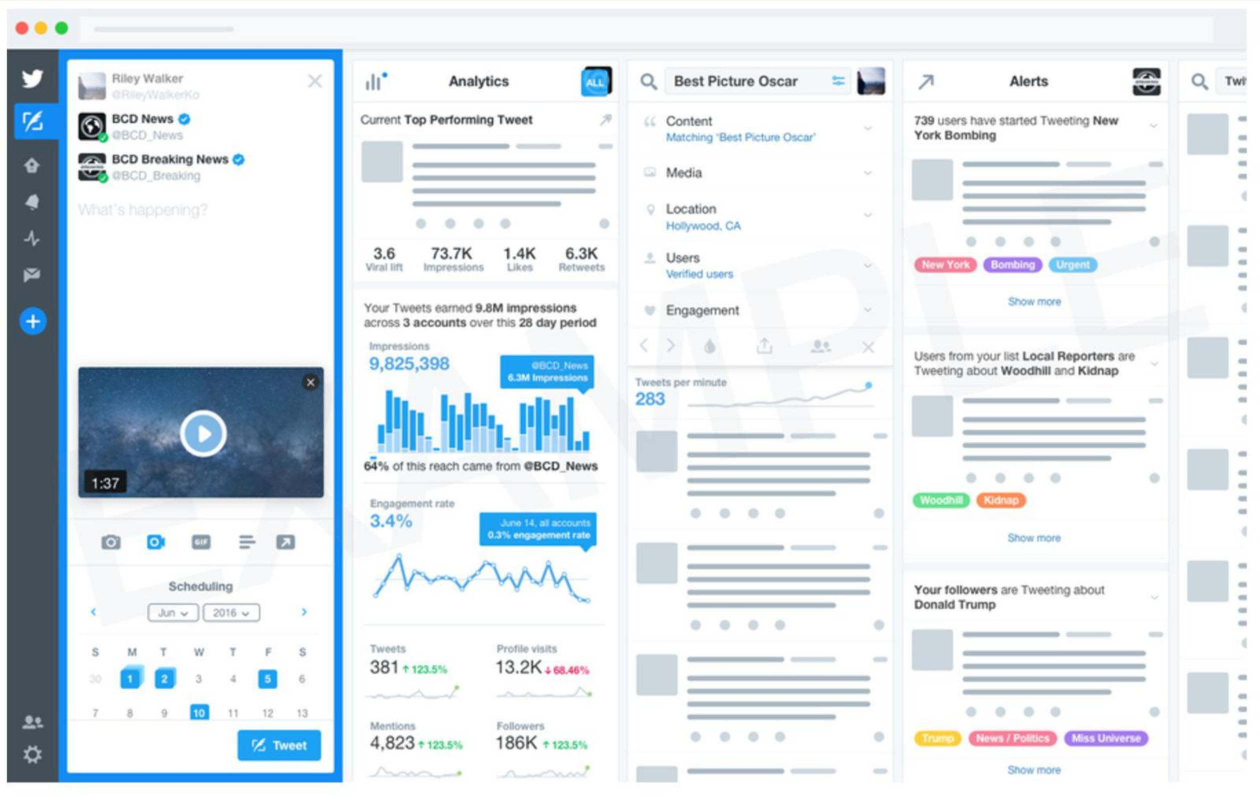
Daedalus' Labyrinth

November – December 2018

A collaborative board game where players uncover heroism in Daedalus' Labyrinth and try their hand at slaying the imprisoned Minotaur. Designed and developed in a team of four.

writeup rules

idle animations



Current Top Performing Tweet

Your Tweets earned 9.8M impressions across 3 accounts over this 28 day period

64% of this reach came from @BCD_News

Content: Matching 'Best Picture Oscar'

Media

Location: Hollywood, CA

Users: Verified users

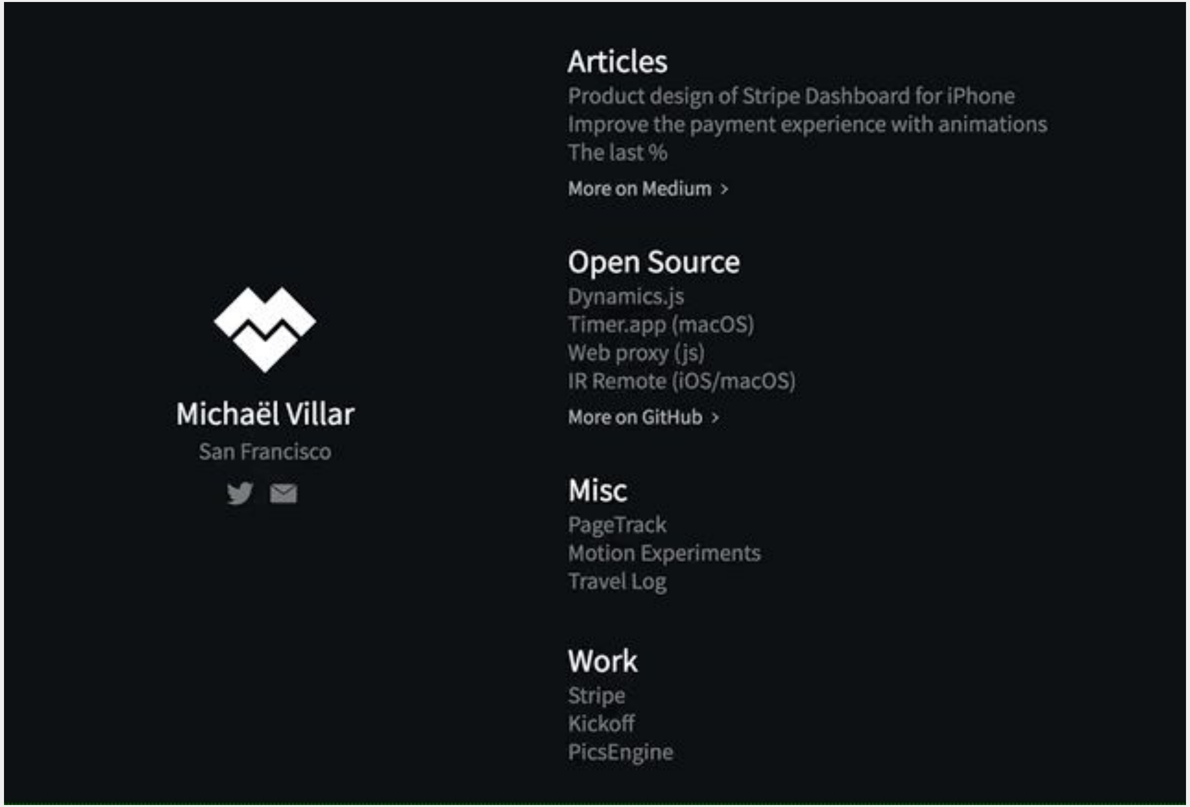
Engagement

739 users have started Tweeting New York Bombing

Users from your list Local Reporters are Tweeting about Woodhill and Kidnap

Your followers are Tweeting about Donald Trump

complex layouts



Michaël Villar

San Francisco

Articles

Product design of Stripe Dashboard for iPhone

Improve the payment experience with animations

The last %

More on Medium >

Open Source

Dynamics.js

Timer.app (macOS)

Web proxy (js)

IR Remote (iOS/macOS)

More on GitHub >

Misc

PageTrack

Motion Experiments

Travel Log

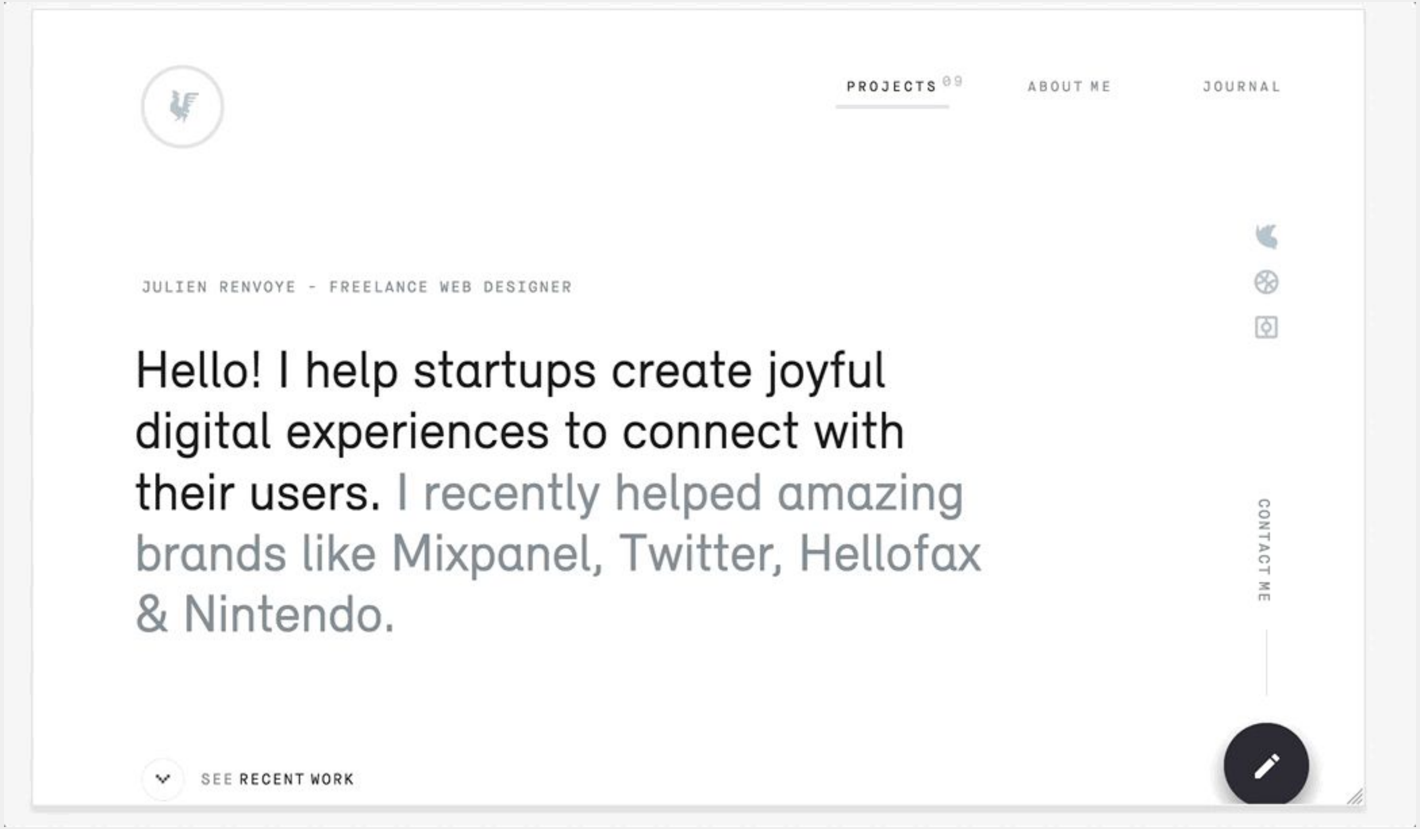
Work

Stripe

Kickoff

PicsEngine

hover animations



PROJECTS 09

ABOUT ME

JOURNAL

JULIEN RENVOYE - FREELANCE WEB DESIGNER

Hello! I help startups create joyful digital experiences to connect with their users. I recently helped amazing brands like Mixpanel, Twitter, Hellofax & Nintendo.

responsive design

why do we care?

mimic expected, intuitive behavior

FOOD GROUPS
FRUITS
GRAINS
DAIRY
VEGGIES
PROTEIN

why aren’t external libraries enough?

you want

custom design

so…
how to make good CSS?

today’s topics

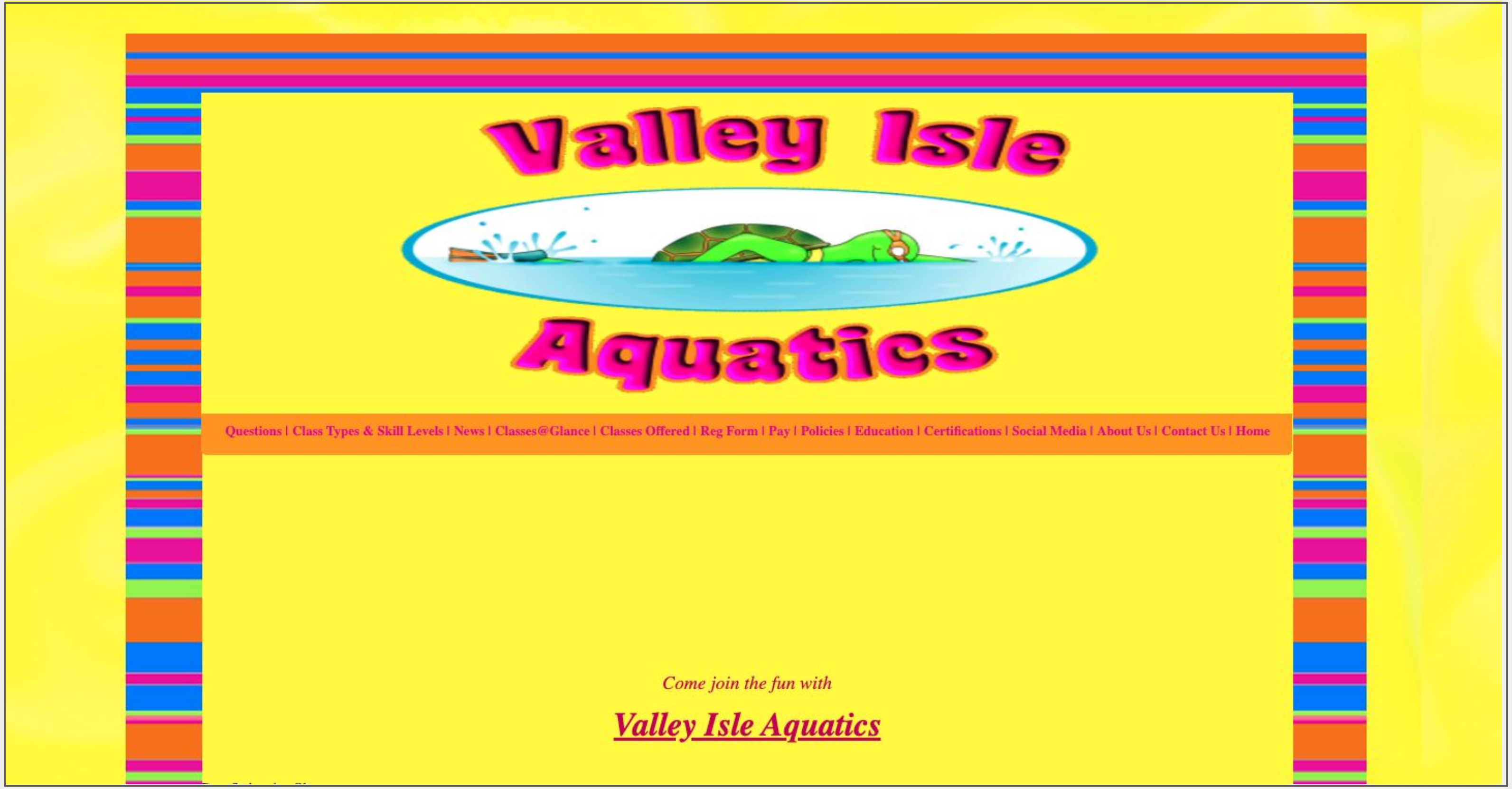
1. Palettes
2. Box model
3. Layouts
4. Transitions & animations
5. Responsive Layouts

today’s topics

1.	Palettes	=>	Colors
2.	Box model	=>	Cards
3.	Layouts	=>	Nice Cards
4.	Transitions & animations	=>	Buttons
5.	Responsive Layouts		

color palettes

why do we care?



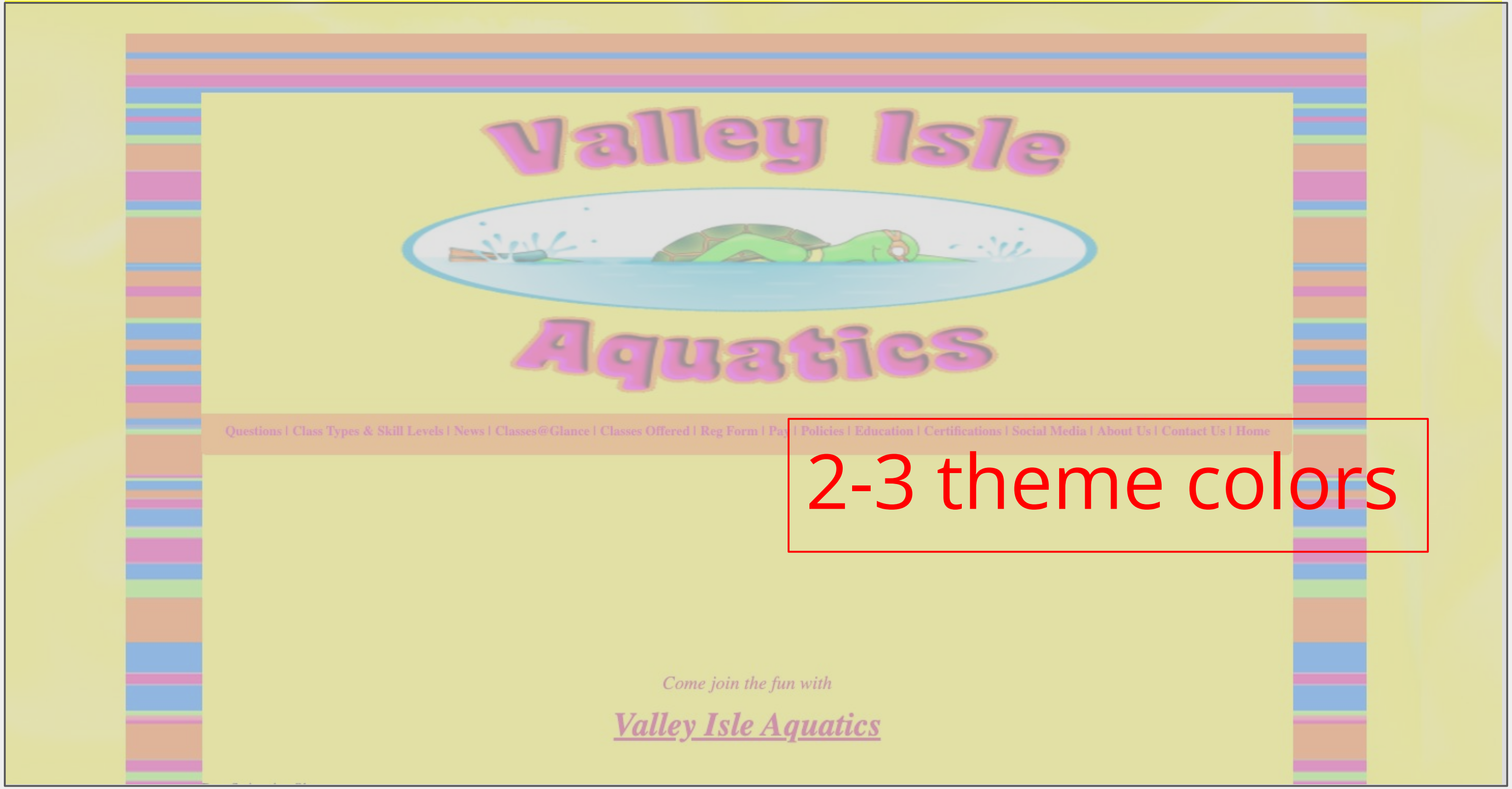
Valley Isle

Aquatics

Questions | Class Types & Skill Levels | News | Classes@Glance | Classes Offered | Reg Form | Pay | Policies | Education | Certifications | Social Media | About Us | Contact Us | Home

Come join the fun with

Valley Isle Aquatics



Valley Isle Aquatics

Questions | Class Types & Skill Levels | News | Classes@Glance | Classes Offered | Reg Form | Pay | Policies | Education | Certifications | Social Media | About Us | Contact Us | Home

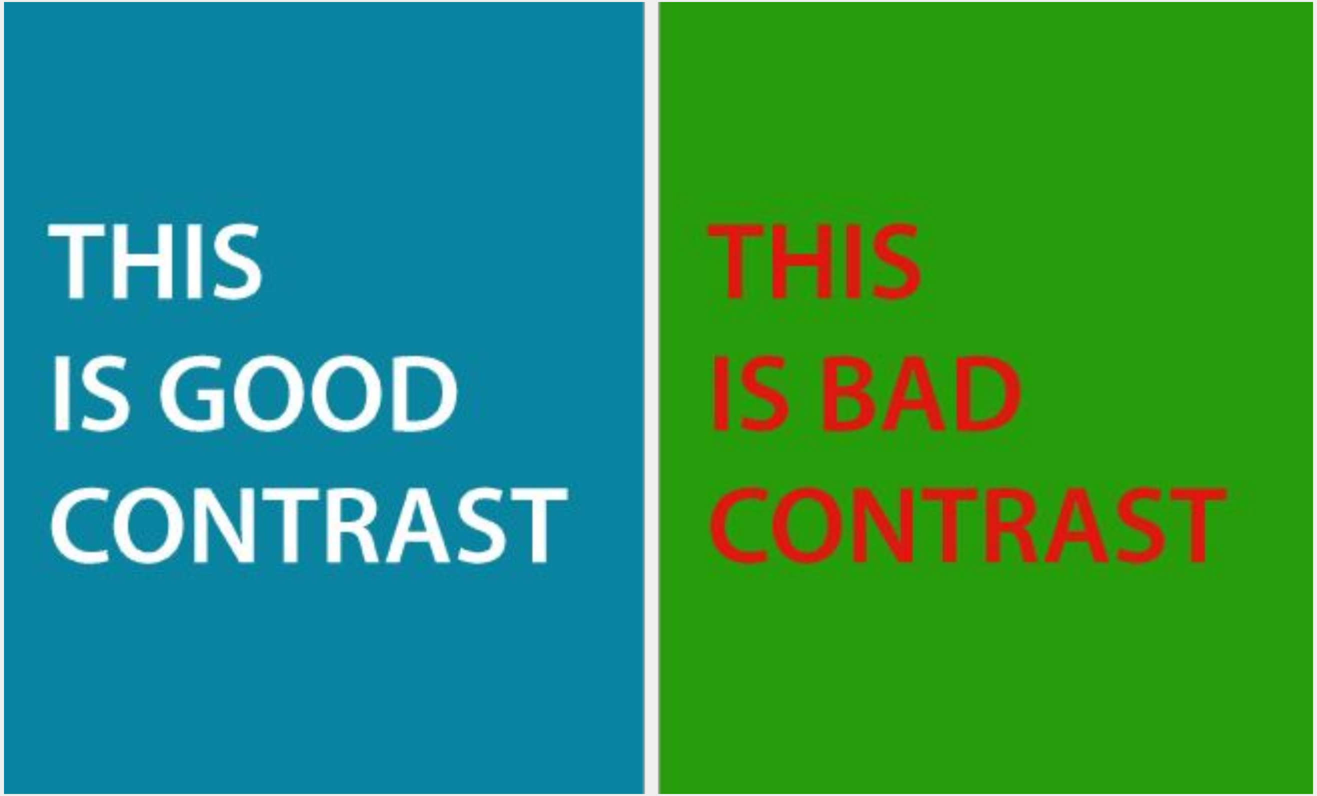
Come join the fun with

Valley Isle Aquatics

2-3 theme colors

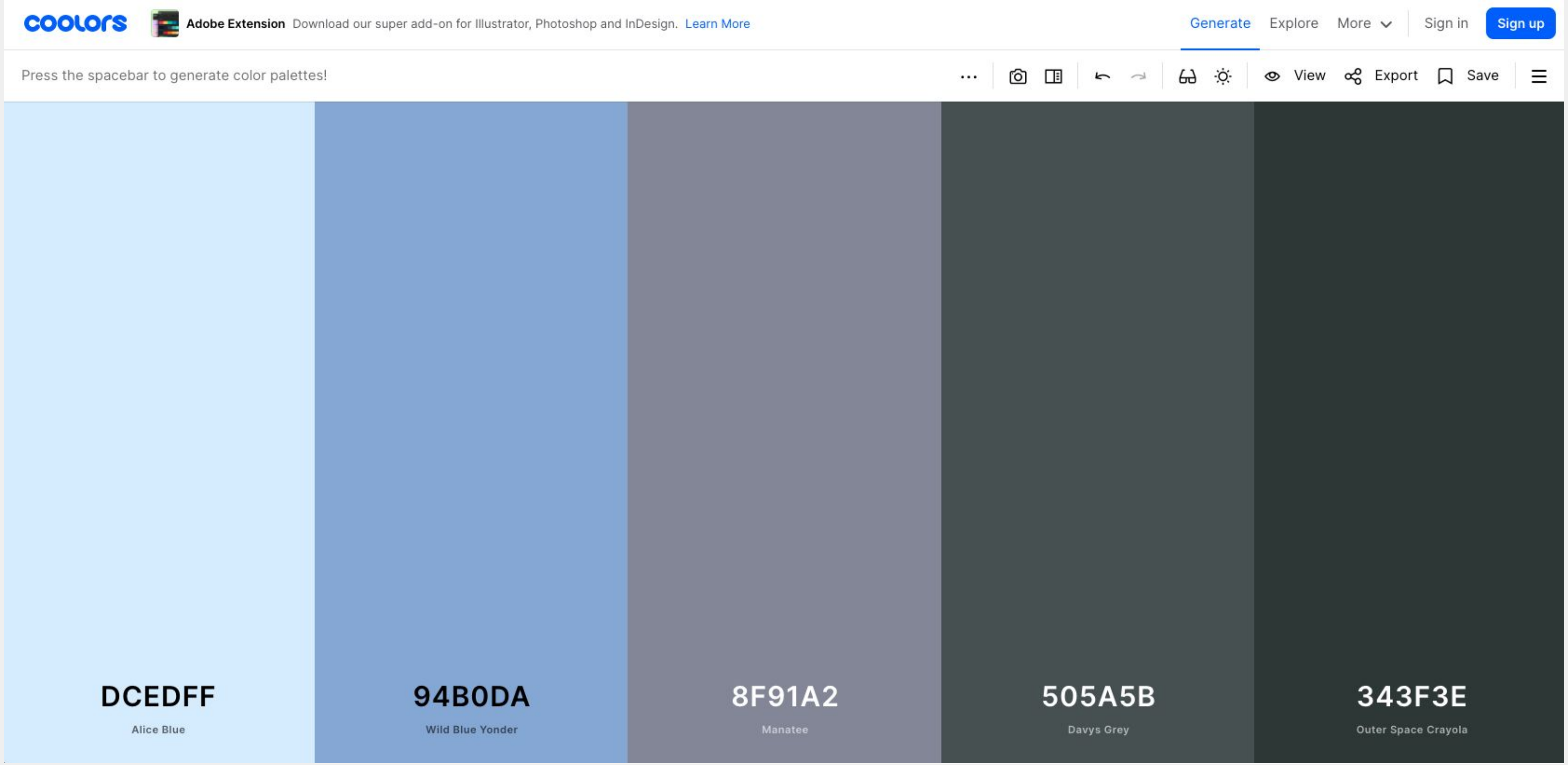
how many theme colors?

2-3 theme colors



THIS
IS GOOD
CONTRAST

THIS
IS BAD
CONTRAST



Download our super add-on for Illustrator, Photoshop and InDesign.

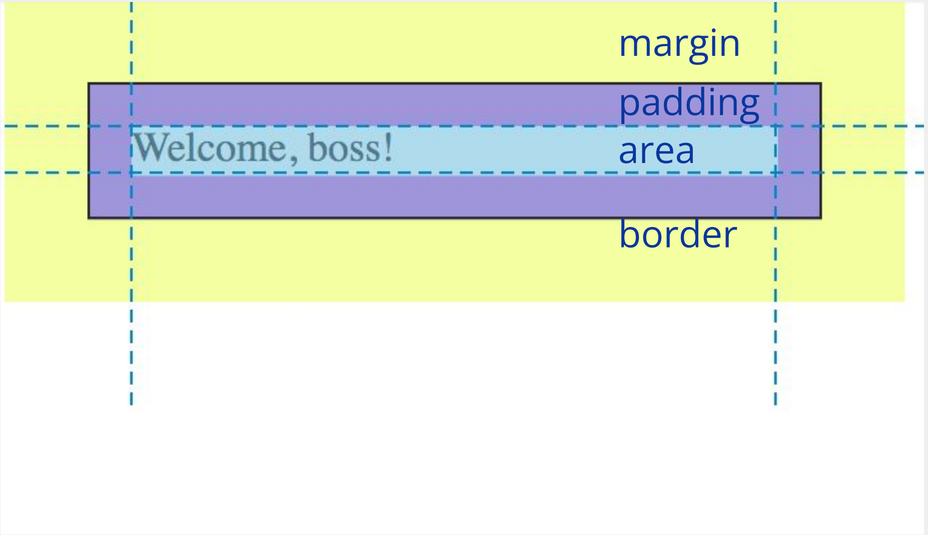
Press the spacebar to generate color palettes!

DCEDFF	94B0DA	8F91A2	505A5B	343F3E
Alice Blue	Wild Blue Yonder	Manatee	Davys Grey	Outer Space Crayola

coolors.co

box model

Box model



layouts

two major properties:

- display
- position

display determines how elements size & sit with/in each other

display: block

big blocks that stretch across, always sit on new lines

generally useful, default property for div

(i.e. <div>, <section>, <ul>, <p>, <h1-6>, <header>)

Lorem Ipsum is simply

dummy

text of the printing and typesetting industry. Lorem Ipsum has been the industry's standard dummy text ever since the 1500s, when an unknown printer took a galley of type and scrambled it to make a type specimen book.

display: inline

an element that is part of text, size is always proportional to text

they do NOT accept width/height properties and top/bottom margins

(i.e. <span>, <a>, <img>)

Lorem Ipsum is simply dummy text of the printing and typesetting industry. Lorem Ipsum has been the industry's standard dummy text ever since the 1500s, when an unknown printer took a galley of type and scrambled it to make a type specimen book.

display: inline-block

stays within text, but can have its own size independent from text

you CAN set width and height

useful for including non-text elements within text

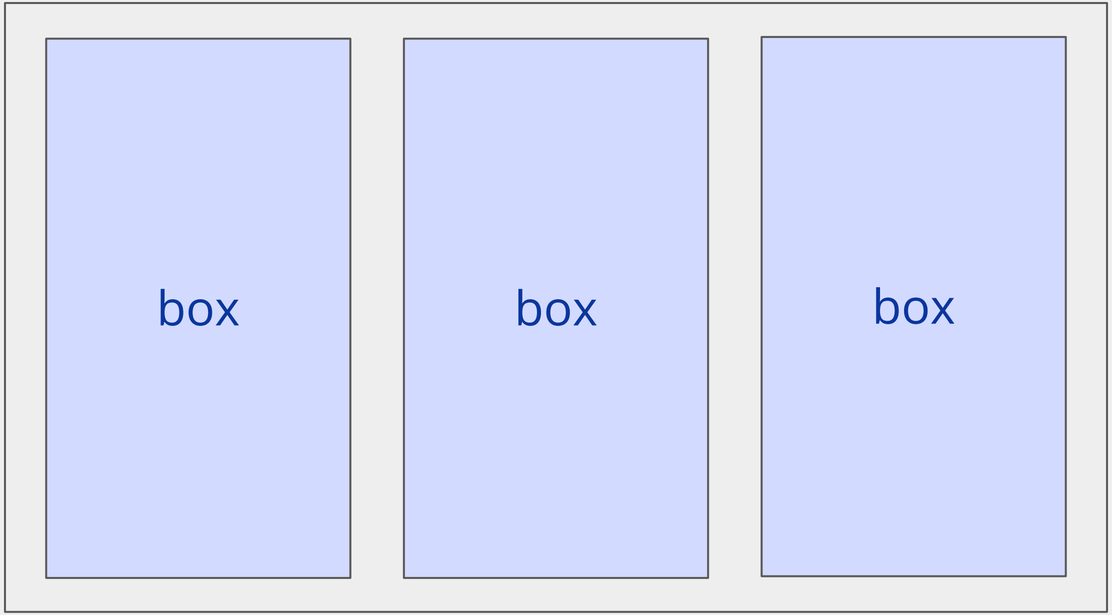
Lorem Ipsum is simply dummy text of the printing and typesetting industry. Lorem Ipsum has been the industry's standard dummy text ever since the 1500s, when an unknown printer took a galley of type and scrambled it to make a type specimen book.

display: none

it’s literally not there

Lorem Ipsum is simply text of the

printing and typesetting industry. Lorem Ipsum has been the industry's standard dummy text ever since the 1500s, when an unknown printer took a galley of type and scrambled it to make a type specimen book.



now… how to do this?

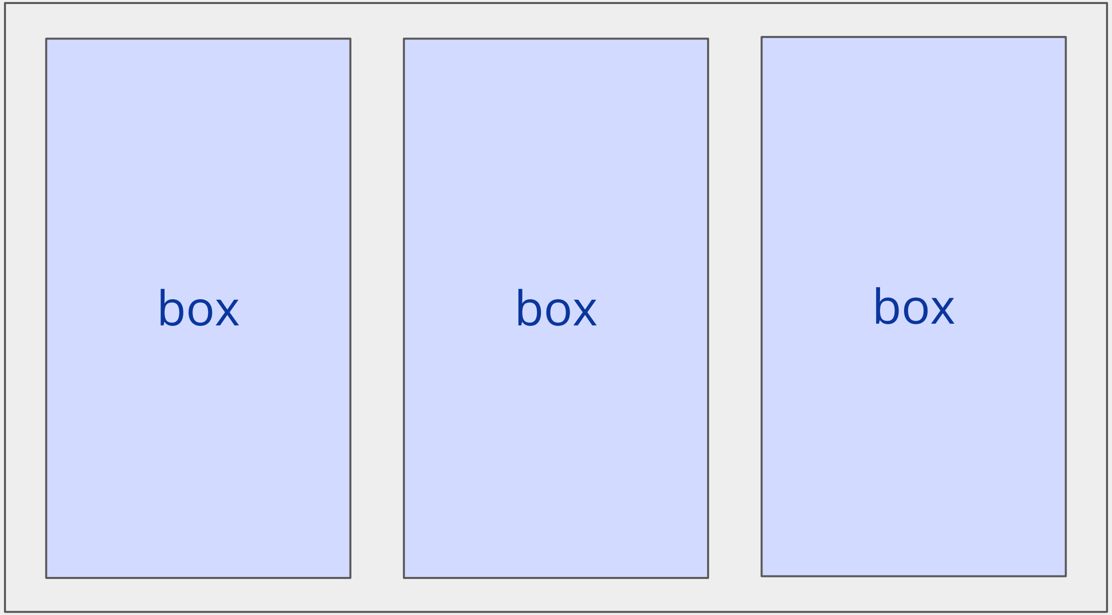
multiple solutions to display elements side-by-side

display: flex

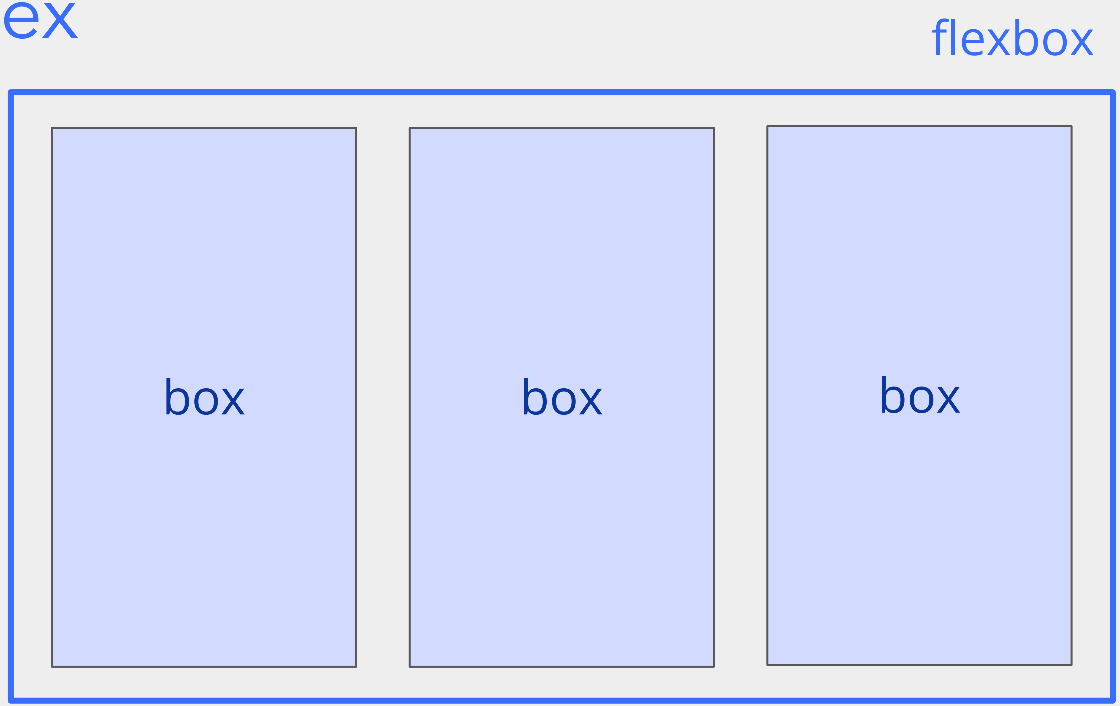
flexbox attempts to *flexibly* contain and distribute elements *inside* itself

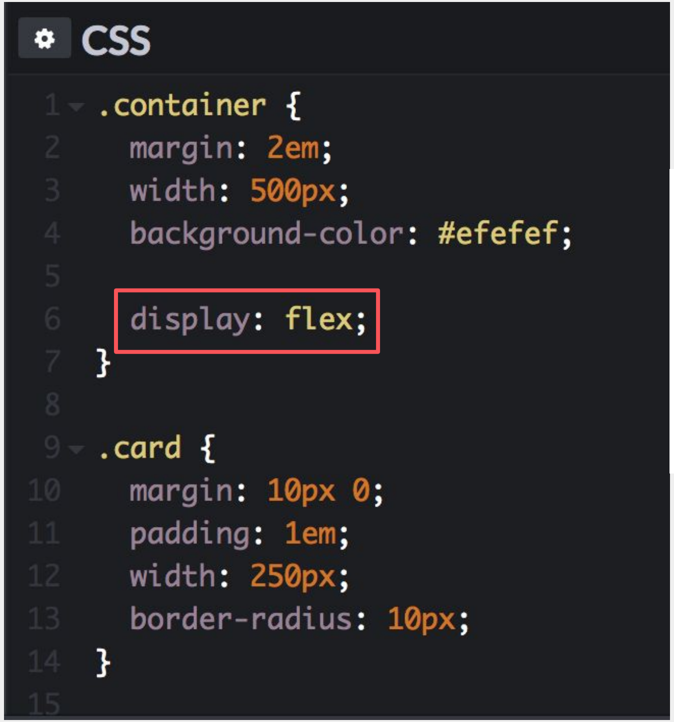
flexible and useful for space distribution

display: flex



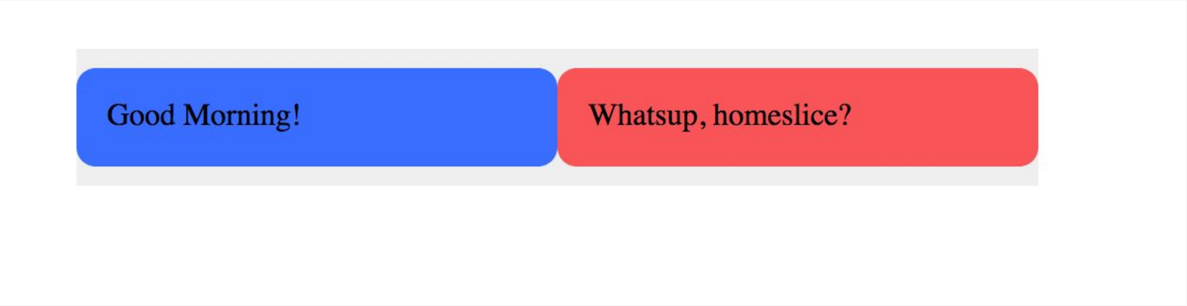
display: flex

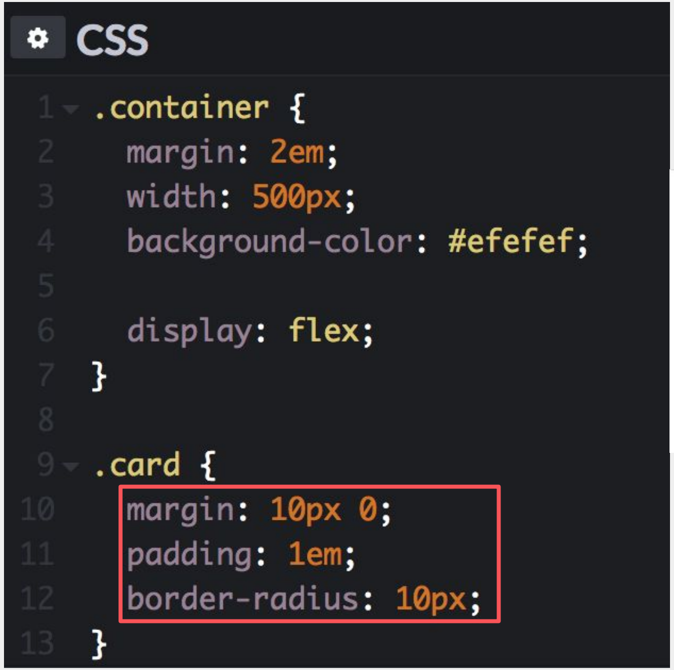




CSS

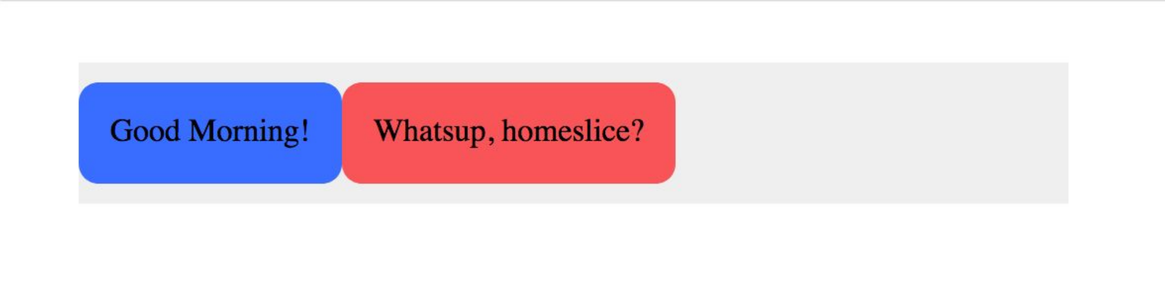
```css
.container {
  margin: 2em;
  width: 500px;
  background-color: #efefef;

  display: flex;
}

.card {
  margin: 10px 0;
  padding: 1em;
  width: 250px;
  border-radius: 10px;
}
```

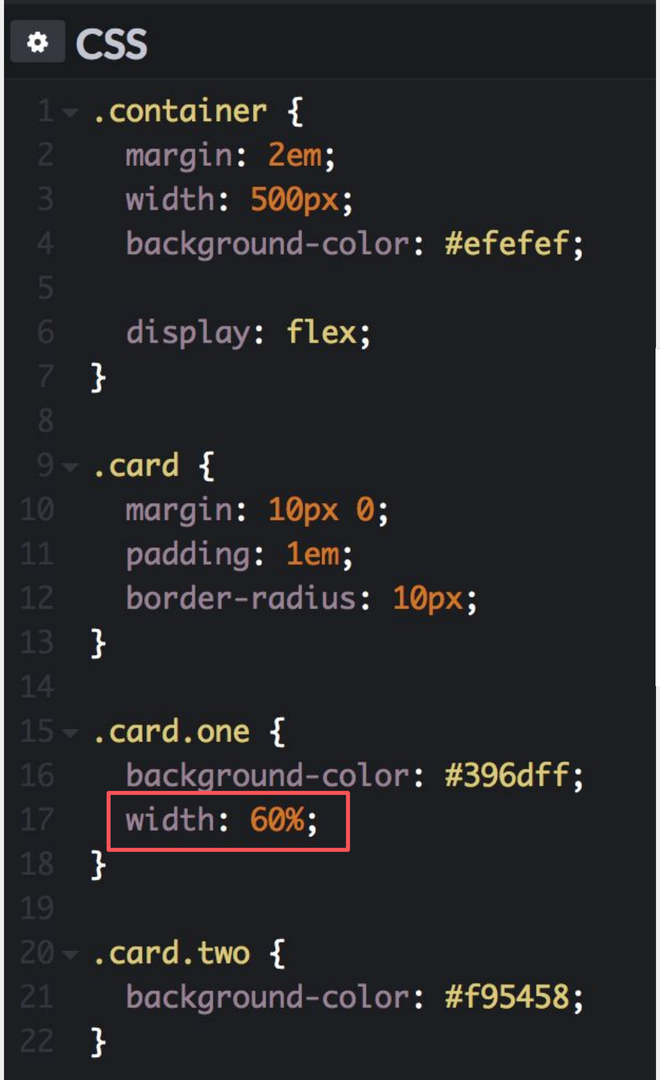




CSS

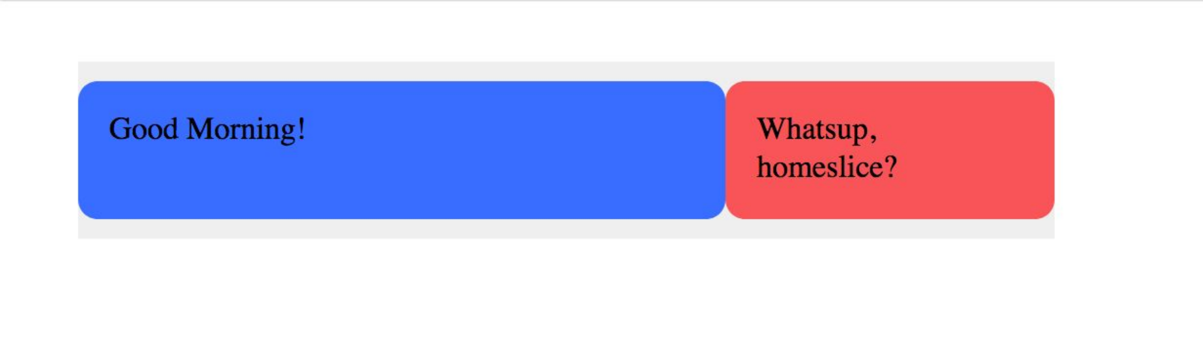
```css
.container {
  margin: 2em;
  width: 500px;
  background-color: #efefef;

  display: flex;
}

.card {
  margin: 10px 0;
  padding: 1em;
  border-radius: 10px;
}
```





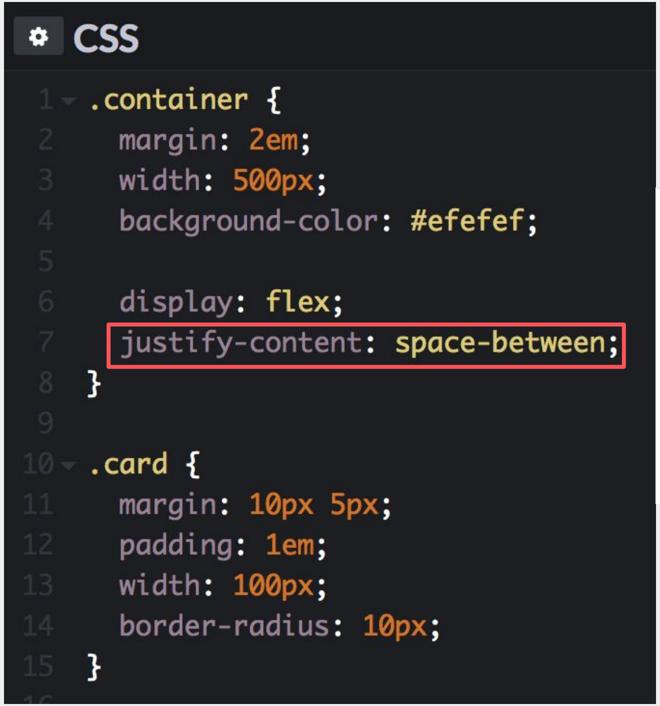
CSS

```css
.container {
  margin: 2em;
  width: 500px;
  background-color: #efefef;

  display: flex;
}

.card {
  margin: 10px 0;
  padding: 1em;
  border-radius: 10px;
}

.card.one {
  background-color: #396dff;
  width: 60%;
}

.card.two {
  background-color: #f95458;
}
```

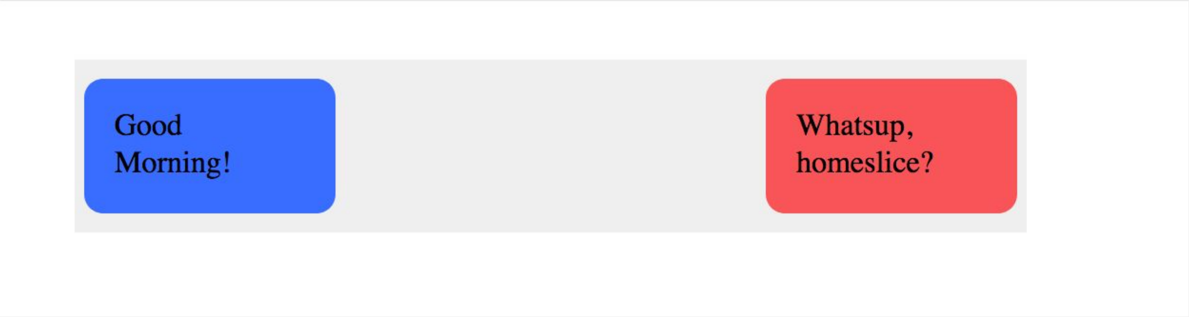


Good Morning!

Whatsup, homeslice?



```css
.container {
  margin: 2em;
  width: 500px;
  background-color: #efefef;

  display: flex;
  justify-content: space-between;
}

.card {
  margin: 10px 5px;
  padding: 1em;
  width: 100px;
  border-radius: 10px;
}
```



‘center’, ‘start’, ‘space-evenly’, etc.

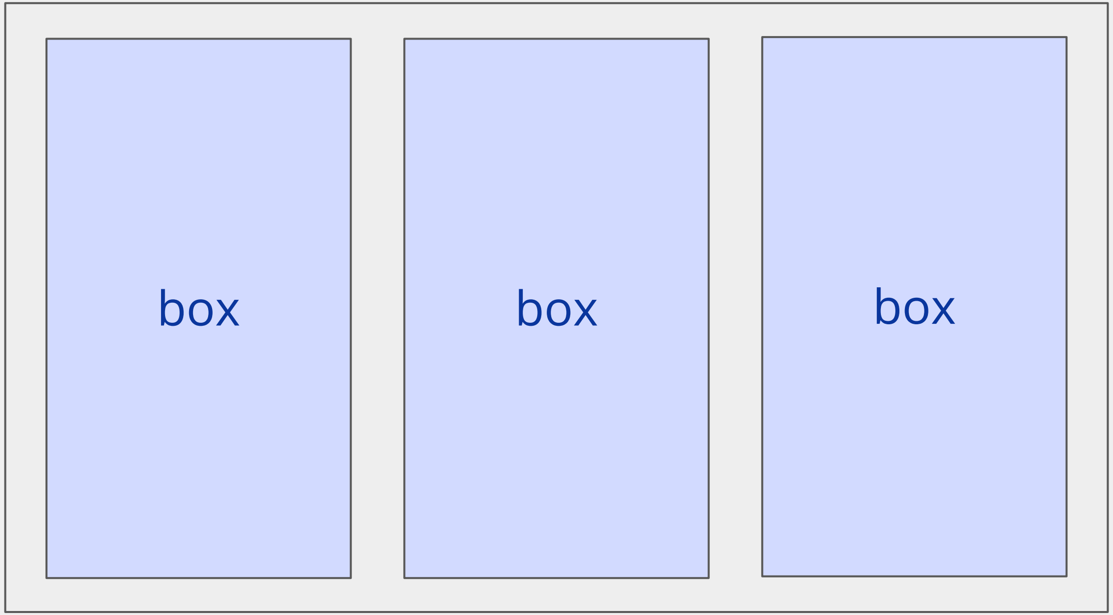
display: grid

creates a “grid” of the child elements

good for managing 2D attributes (sizing of rows & columns)

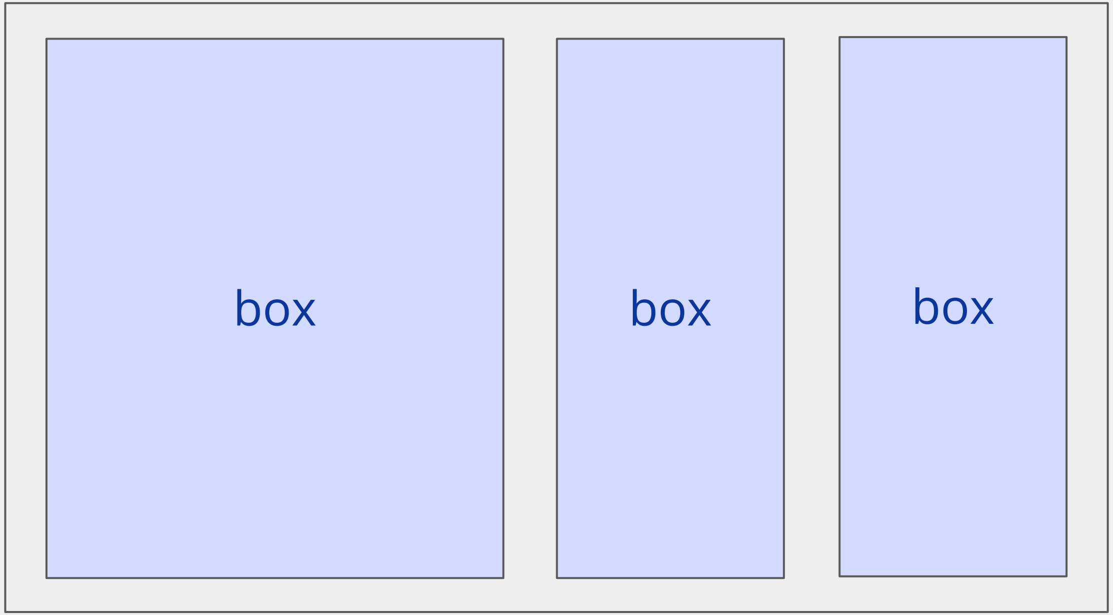
display: grid

```
.container {
  display: grid;
  grid-template-columns: 1fr 1fr 1fr;
}
```



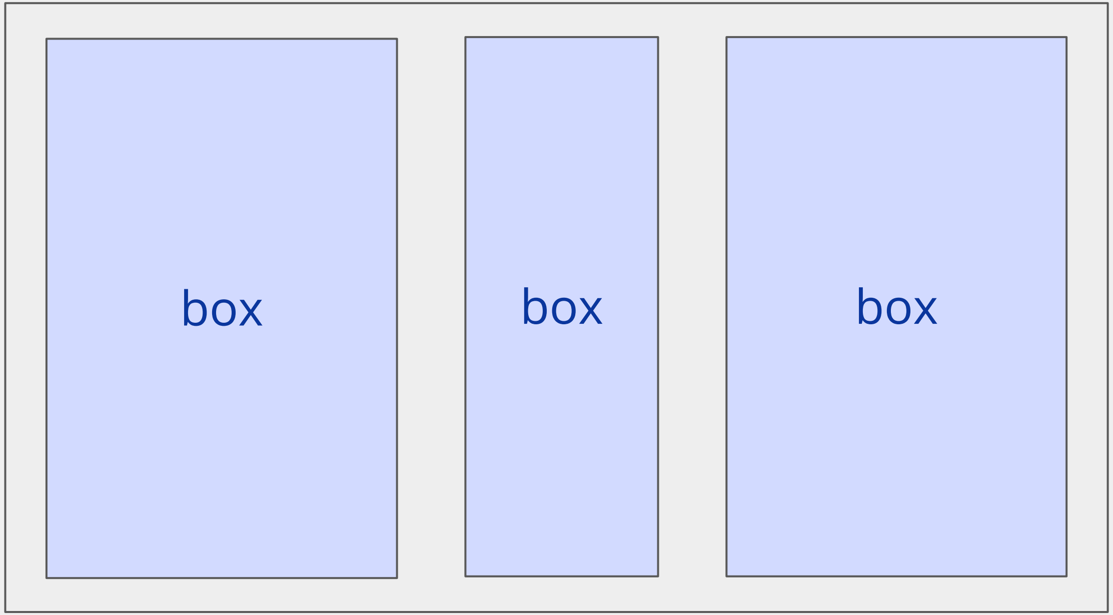
display: grid

```
.container {
  display: grid;
  grid-template-columns: 2fr 1fr 1fr;
}
```



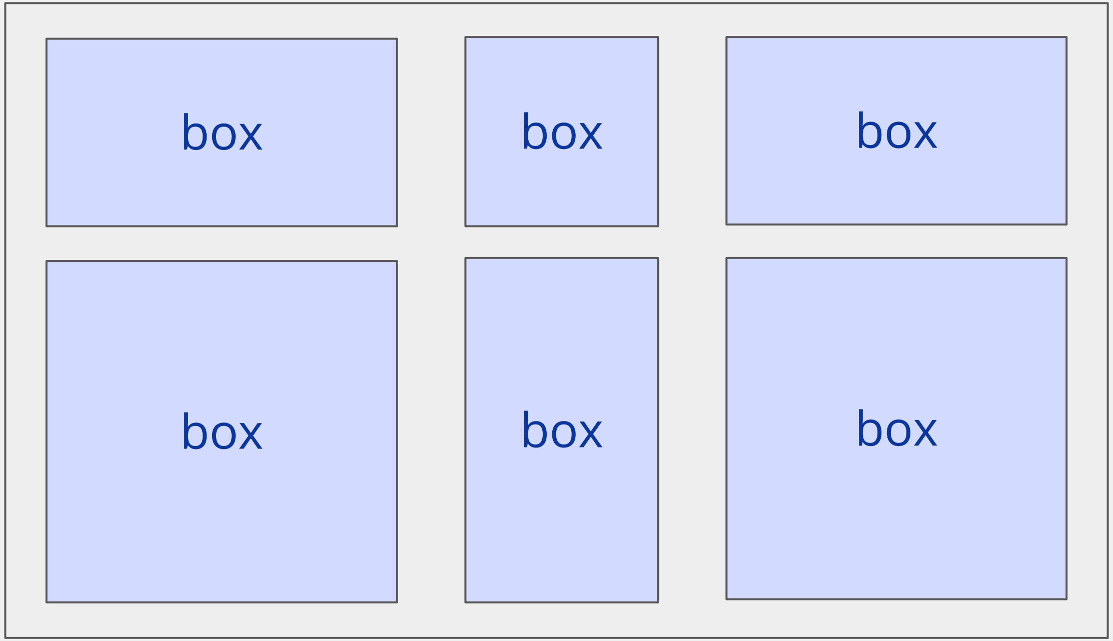
display: grid

```
.container {
  display: grid;
  grid-template-columns: 2fr 1fr 2fr;
}
```



display: grid

```
.container {
  display: grid;
  grid-template-columns: 2fr 1fr 2fr;
  grid-template-rows: 100px 200px;
}

// 2D controls!!
```

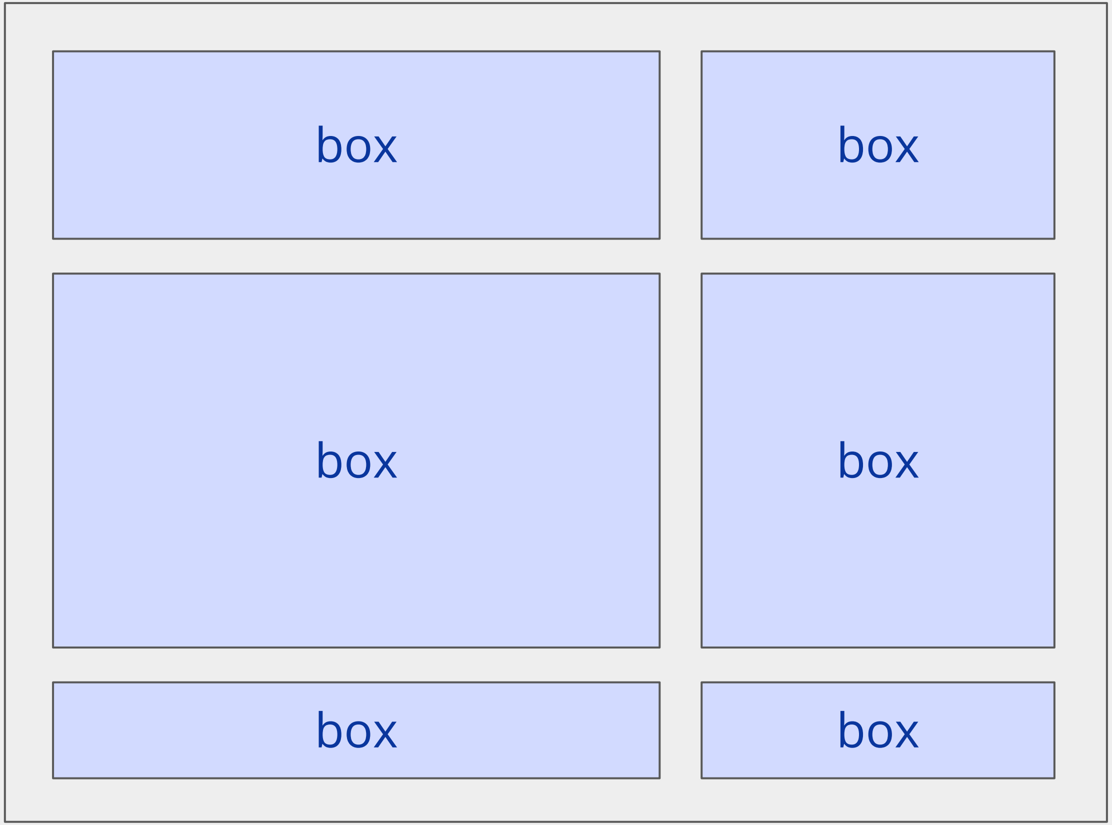


box box box

box box box

display: grid

```
.container {
  display: grid;
  grid-template-columns: 2fr 1fr;
  grid-template-rows: 100px 200px 50px;
}

// 2D controls!!
```



position determines
where an element sits
based on other elements

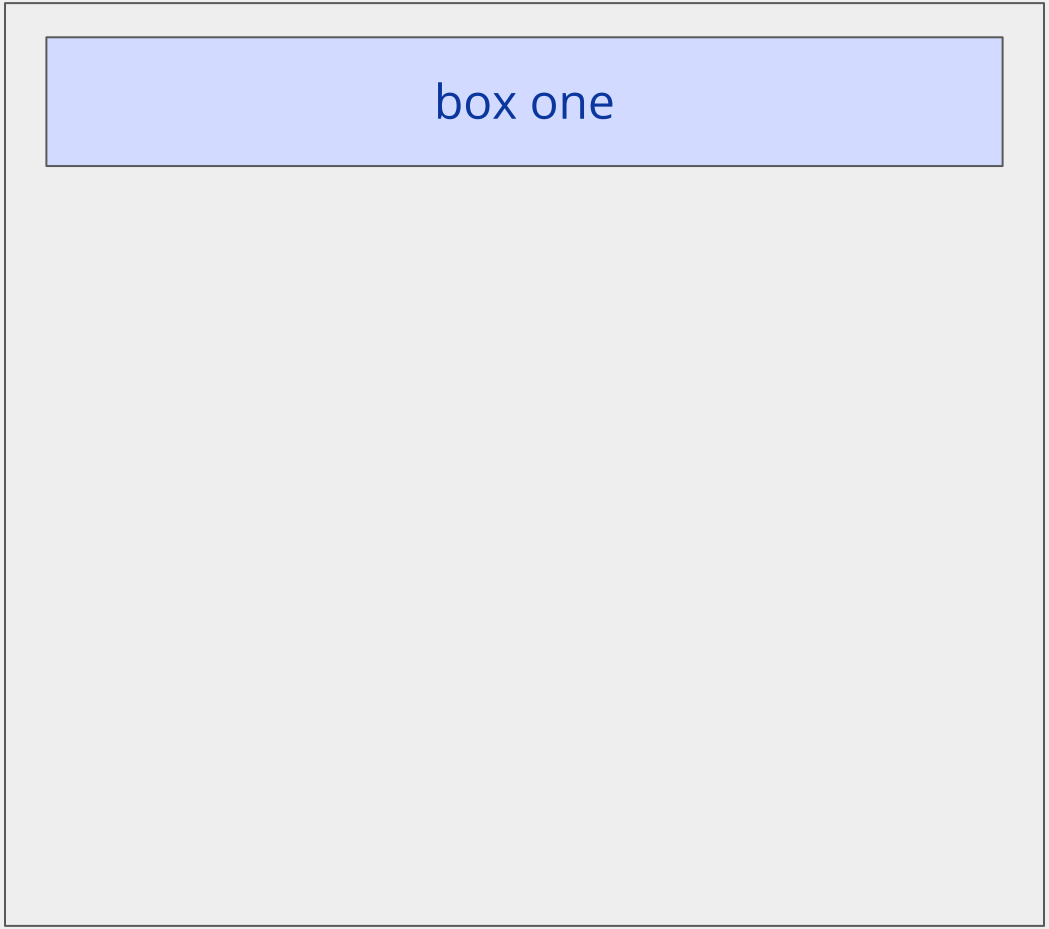
position: static

renders boxes position based on order in document

default property value for div

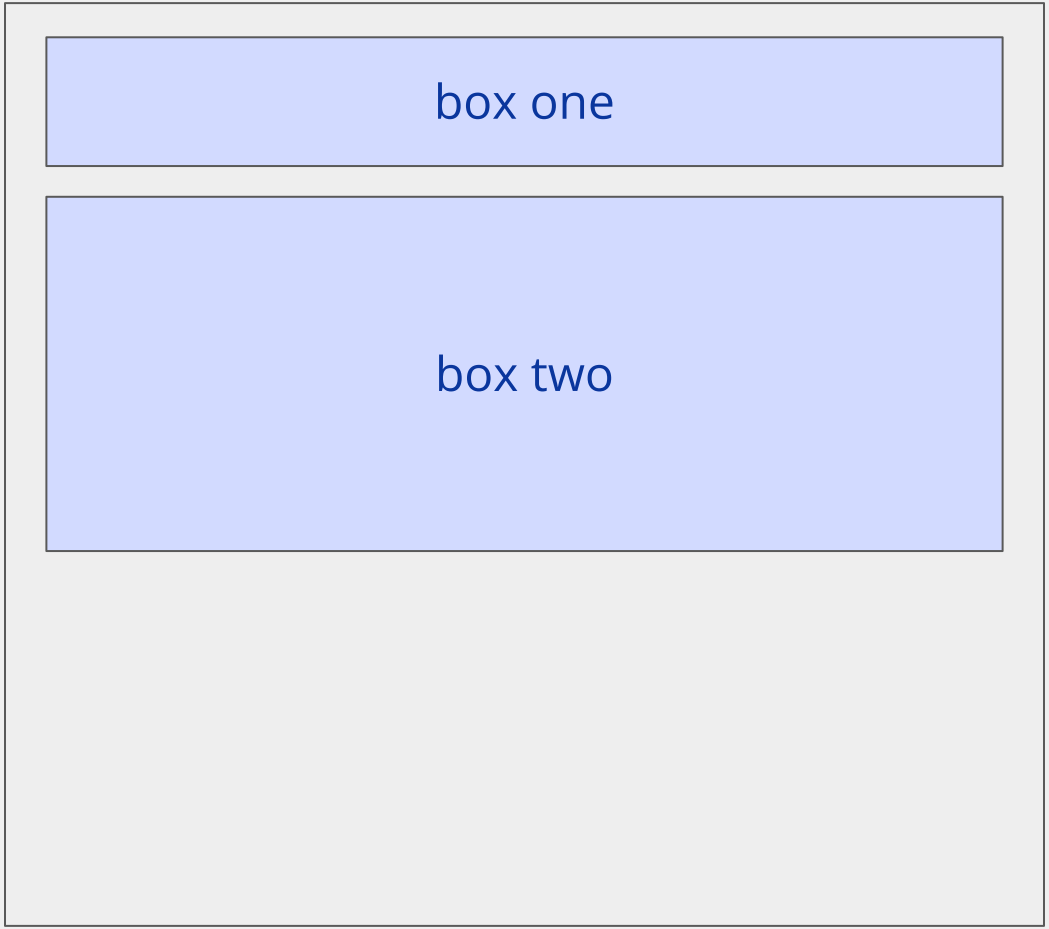
HTML

```
<div className=b1>
  box one
</div>
```



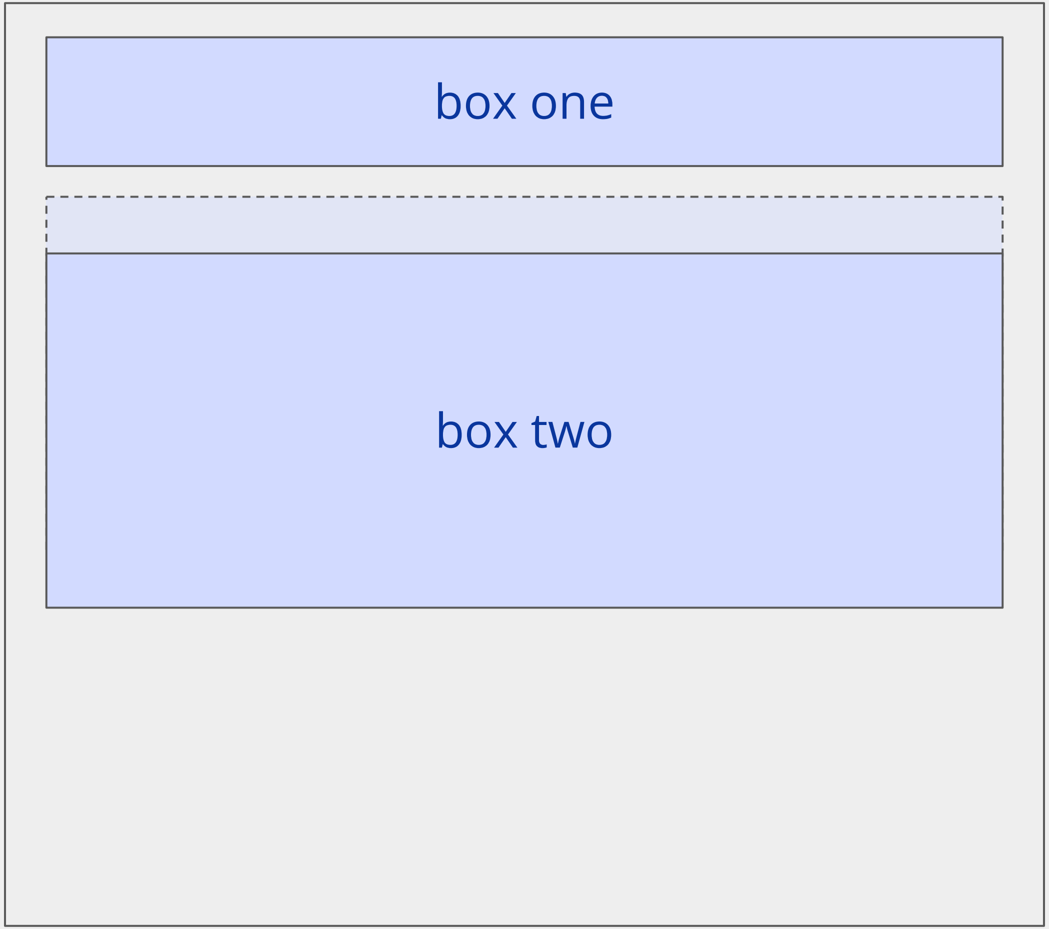
HTML

```
<div className=b1>
  box one
</div>
<div className=b2>
  box two
</div>
```



HTML

```
<div className=b1>
  box one
</div>
<div className=b2>
  box two
</div>
```



position: relative

positions the element “relative” to where it would be if static

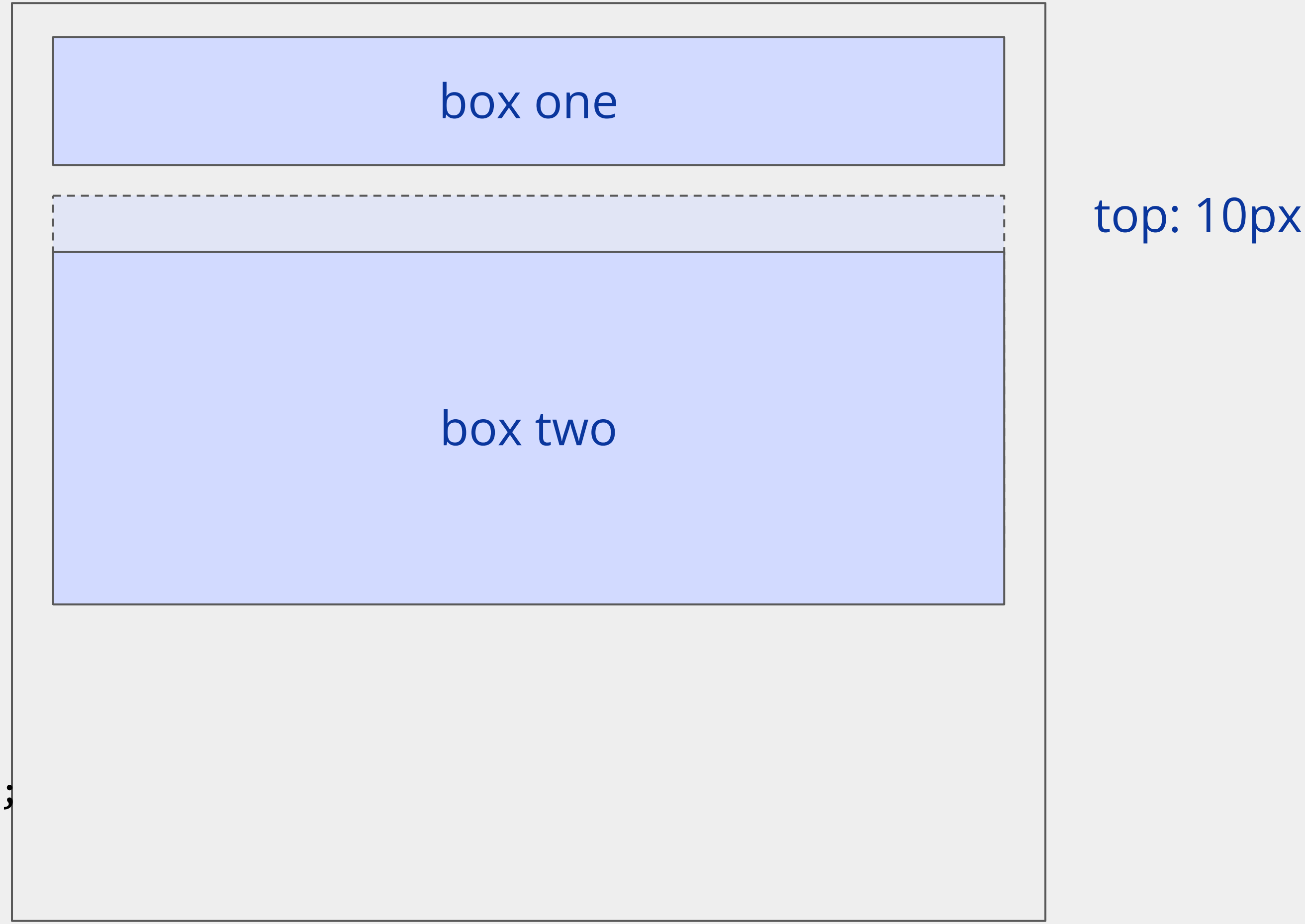
useful for slight modifications in position

HTML

```
<div className=b1>
  box one
</div>
<div className=b2>
  box two
</div>
```

CSS

```
.b1 {
  //rules
}
.b2 {
  position: relative;
  top: 10px;
}
```

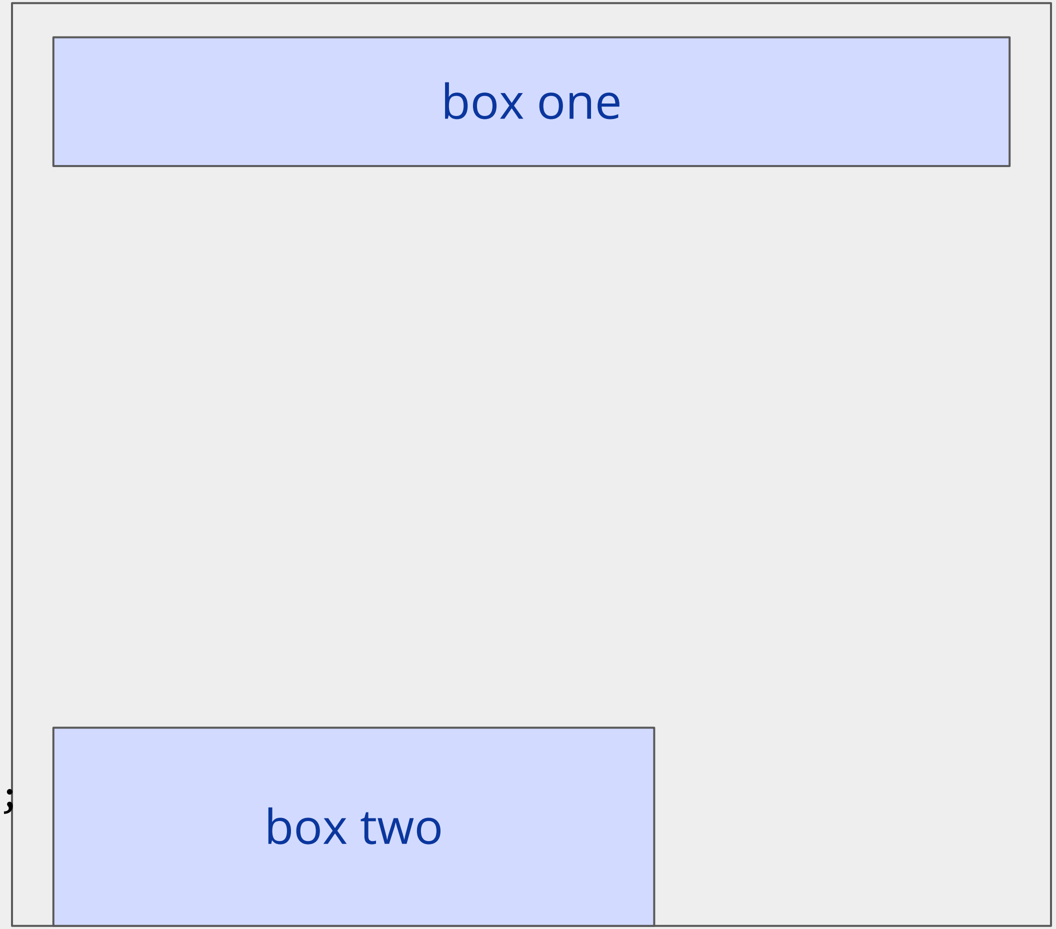


HTML

```
<div className=b1>
  box one
</div>
<div className=b2>
  box two
</div>
```

CSS

```
.b1 {
  //rules
}
.b2 {
  position: relative;
  top: ??????
}
```



position: absolute

positions the element relative to first ancestor positioned non-statically

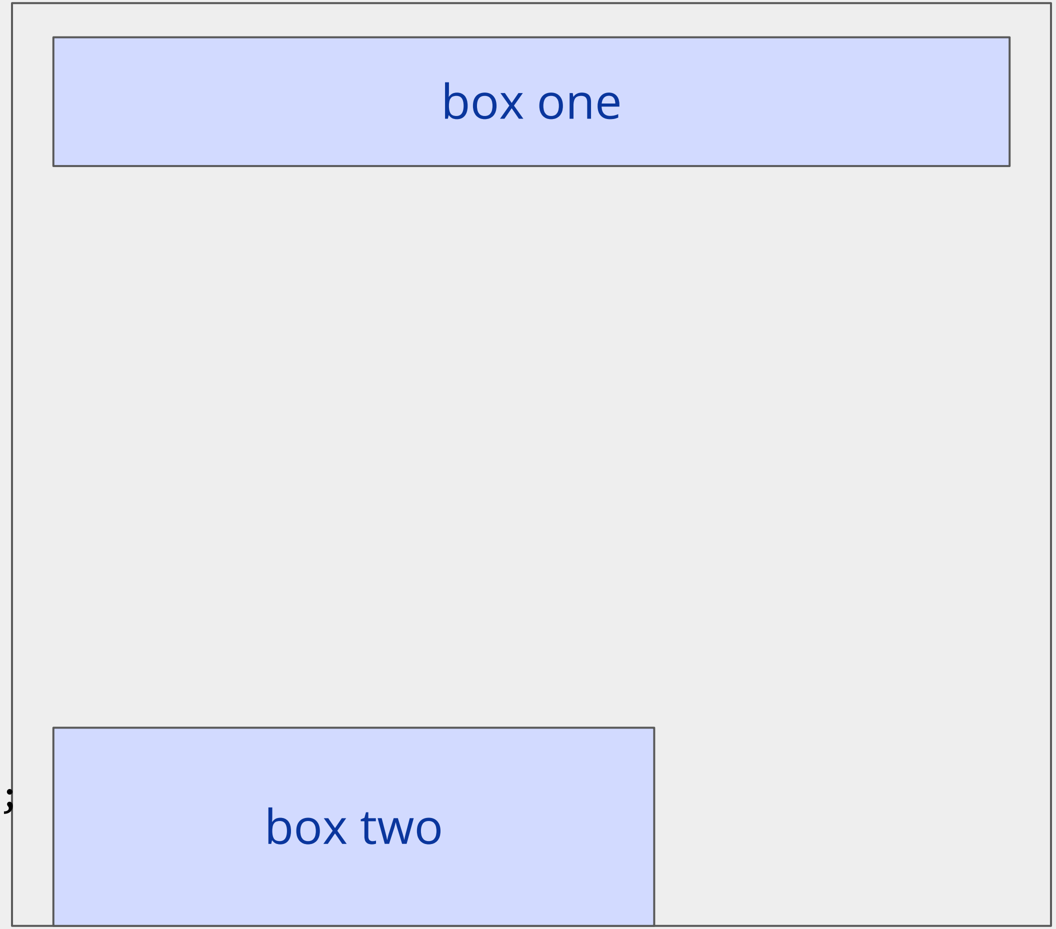
useful for navbars and sidebars

HTML

```
<div className=b1>
  box one
</div>
<div className=b2>
  box two
</div>
```

CSS

```
.b1 {
  //rules
}
.b2 {
  position: absolute;
  bottom: 0px;
}
```



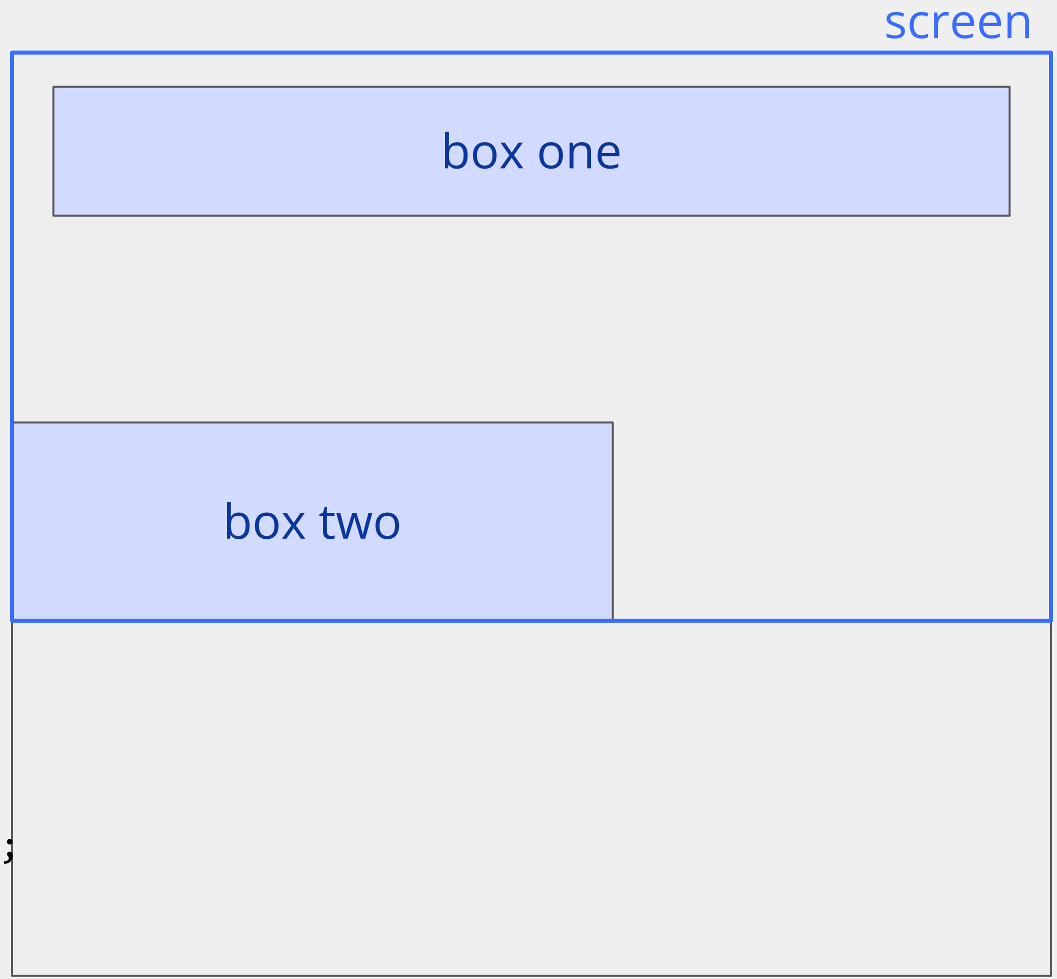
bottom: 0px

HTML

```
<div className=b1>
  box one
</div>
<div className=b2>
  box two
</div>
```

CSS

```
.b1 {
  //rules
}
.b2 {
  position: absolute;
  bottom: 0px;
}
```

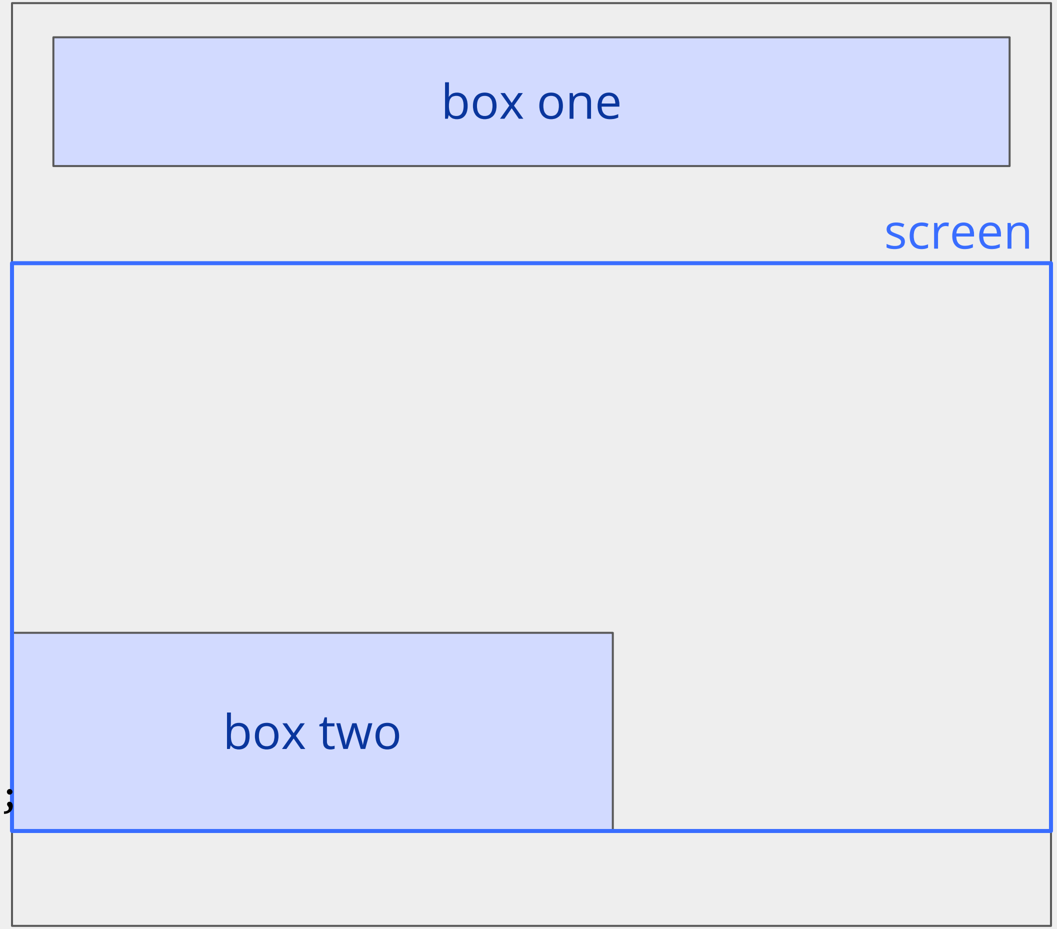


HTML

```
<div className=b1>
  box one
</div>
<div className=b2>
  box two
</div>
```

CSS

```
.b1 {
  //rules
}
.b2 {
  position: absolute;
  bottom: -1239px??
}
```



position: fixed

positions the element relative to the screen

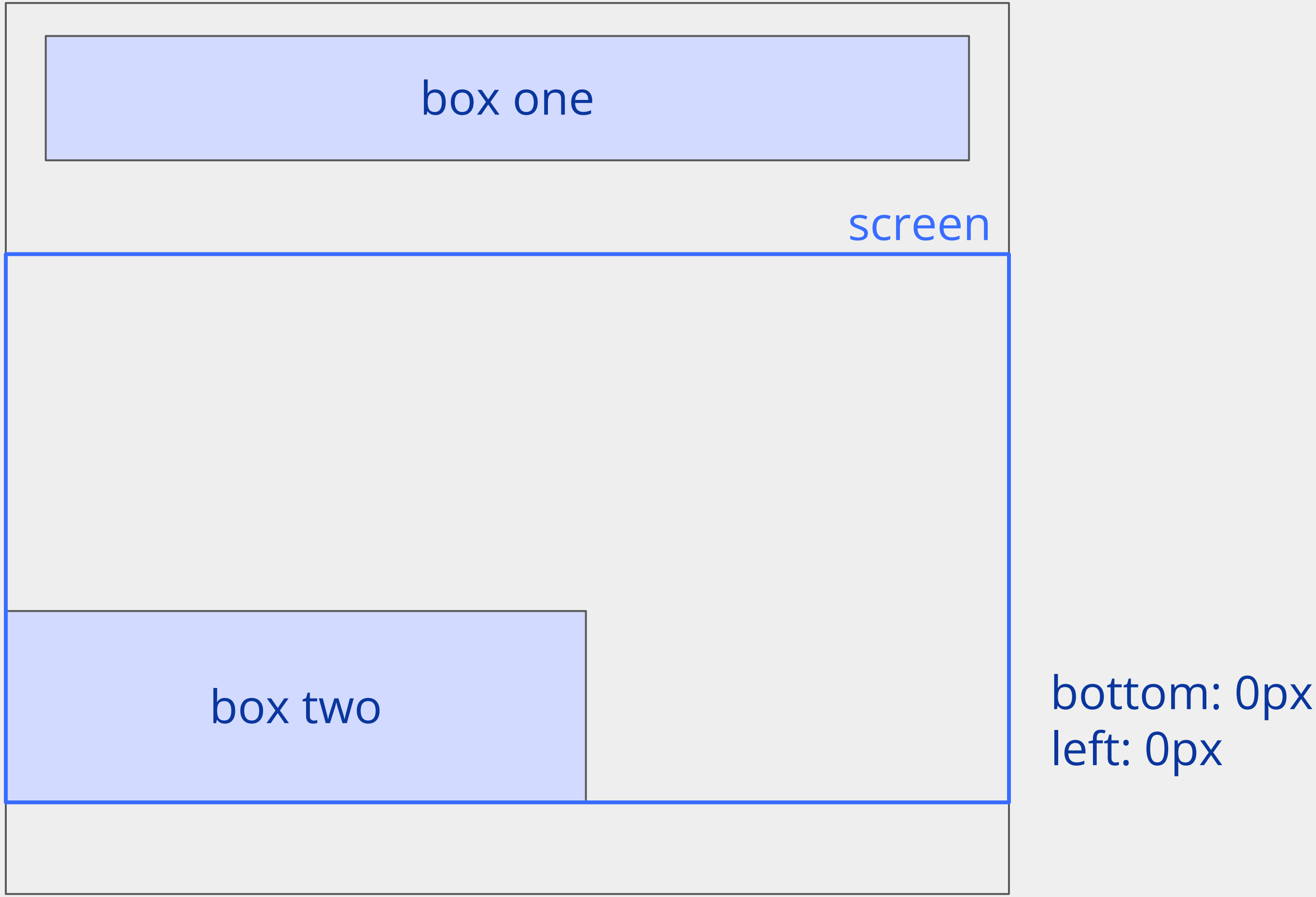
useful for navbars and bulletins

HTML

```
<div className=b1>
  box one
</div>
<div className=b2>
  box two
</div>
```

CSS

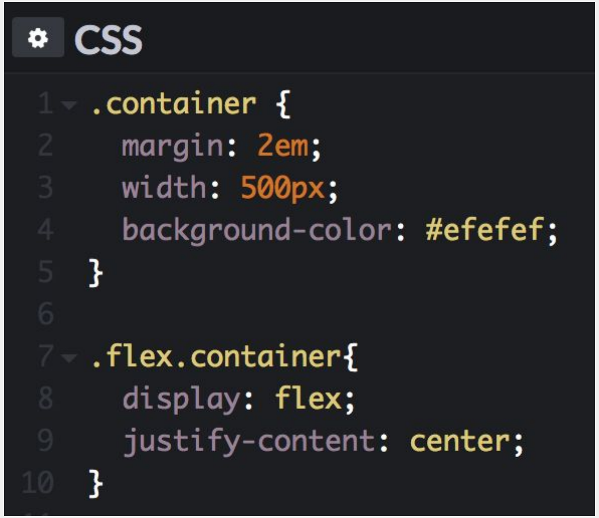
```
.b1 {
  //rules
}
.b2 {
  position: fixed;
  bottom: 0px;
  left: 0px;
}
```



challenge!

center a div horizontally

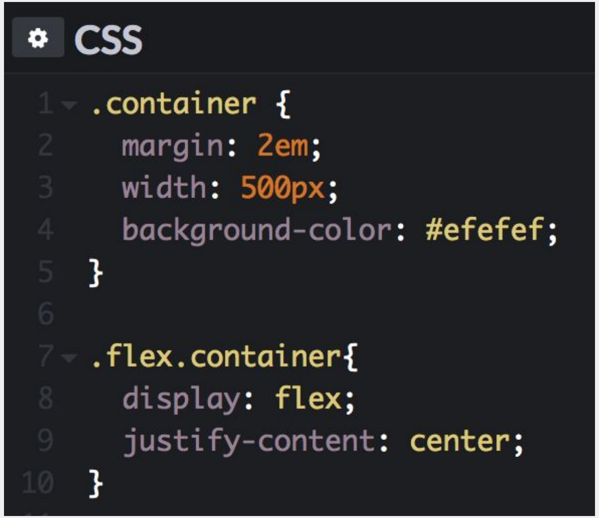
flexbox



```css
.container {
  margin: 2em;
  width: 500px;
  background-color: #efefef;
}

.flex.container{
  display: flex;
  justify-content: center;
}
```

parent container has flex display

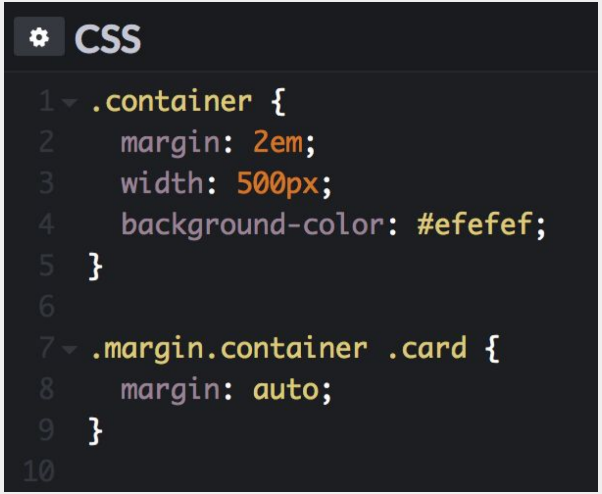
flexbox



```css
.container {
  margin: 2em;
  width: 500px;
  background-color: #efefef;
}

.flex.container{
  display: flex;
  justify-content: center;
}
```

parent container has flex display

margins



```css
.container {
  margin: 2em;
  width: 500px;
  background-color: #efefef;
}

.margin.container .card {
  margin: auto;
}

```

child element has auto margins

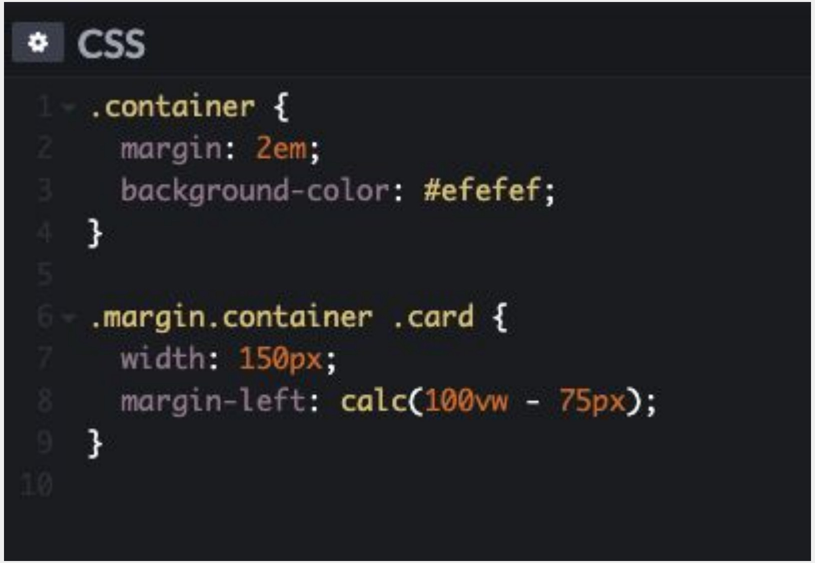
using calc and vw

100vw

→ 100% viewport width (full width of window)

calc

→ calculate math thingies



```
.container {
  margin: 2em;
  background-color: #efefef;
}

.margin.container .card {
  width: 150px;
  margin-left: calc(100vw - 75px);
}
```

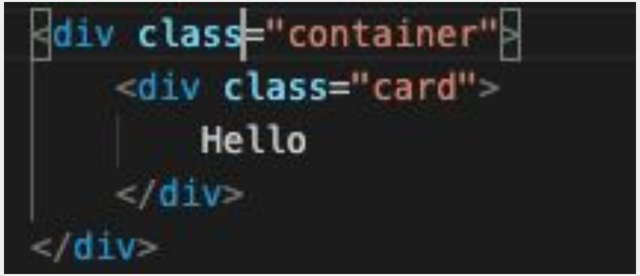
other relative length units like vh (viewport height), em (current font size) are useful there are many solutions

let’s look at some code

transitions & animations

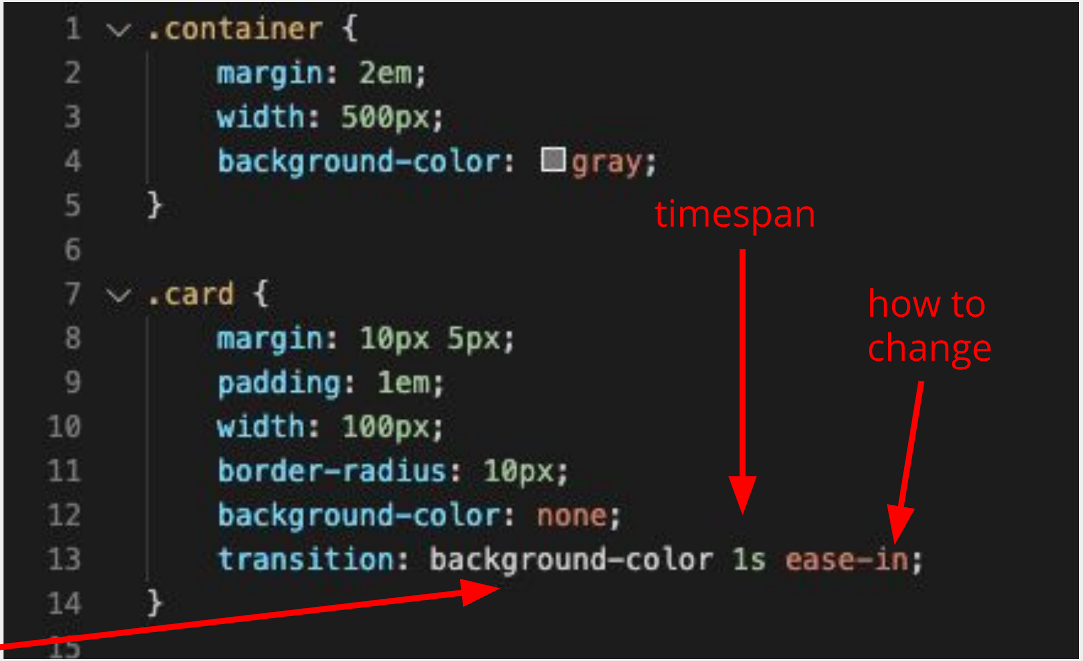
transition determines how changes in properties show up on your screen!

HTML



```
<div class="container">
    <div class="card">
        Hello
    </div>
</div>
```

CSS



```
.container {
    margin: 2em;
    width: 500px;
    background-color: gray;
}

.card {
    margin: 10px 5px;
    padding: 1em;
    width: 100px;
    border-radius: 10px;
    background-color: none;
    transition: background-color 1s ease-in;
}
```

timespan

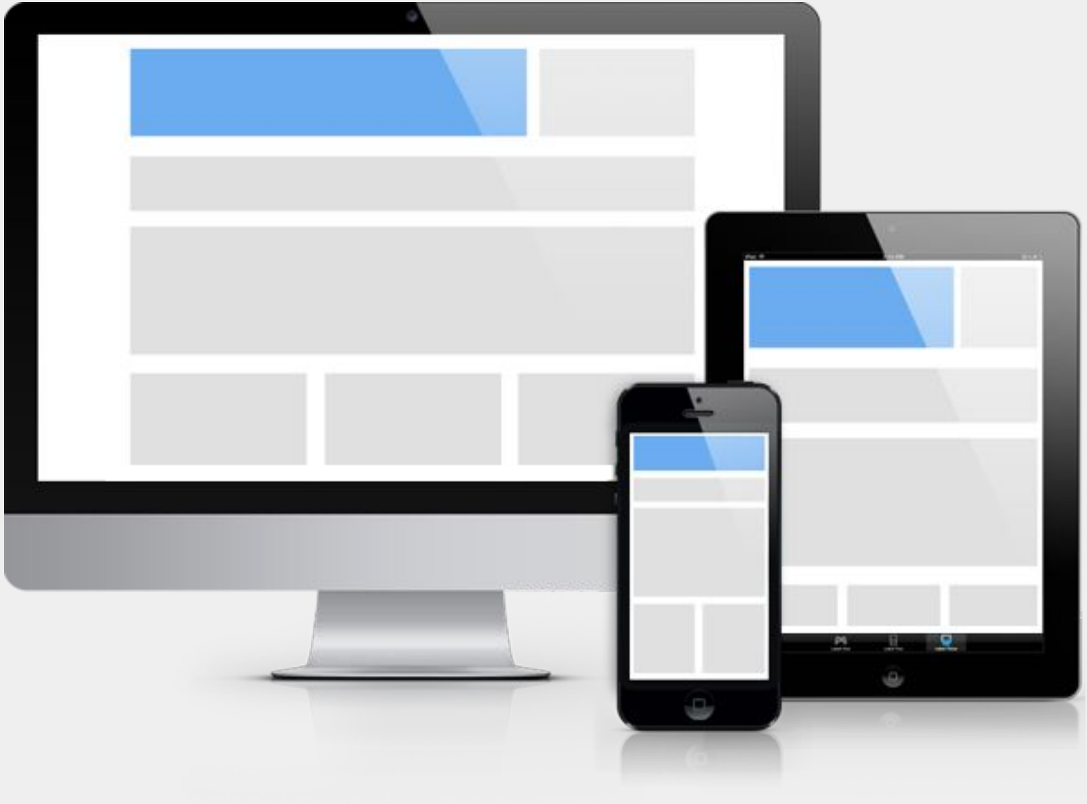
how to change

property that we are changing

animation defines keyframes to transition through

let’s look at some code

responsive layouts



Media queries

Media queries allow you to run certain CSS only if:

- The device screen is at least certain size
- The device supports hovering (with a mouse)
- The device supports colors
- … Speech input, aspect ratio, etc

let’s see it in action

weblab.mit.edu

code time

Special CSS Selectors

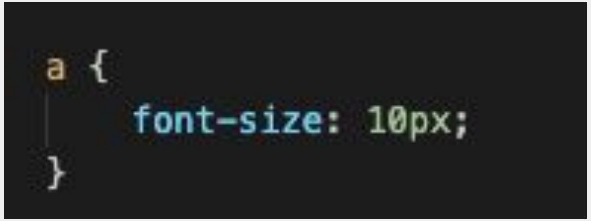
Blatantly ripped from docs: https://developer.mozilla.org/en-US/docs/Web/CSS/CSS_Selectors

Recap of Selectors

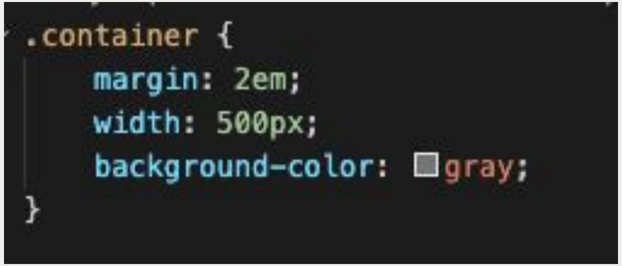
- Type selectors: a, p, h1, div, etc

- Class selectors: .classname

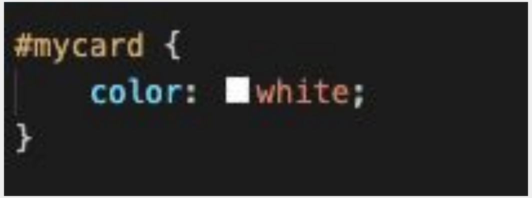
- ID selectors: #idname



```css
a {
    font-size: 10px;
}
```



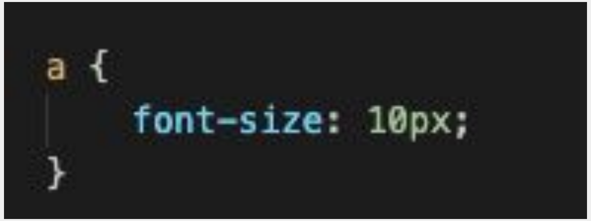
```css
.container {
    margin: 2em;
    width: 500px;
    background-color: ■gray;
}
```



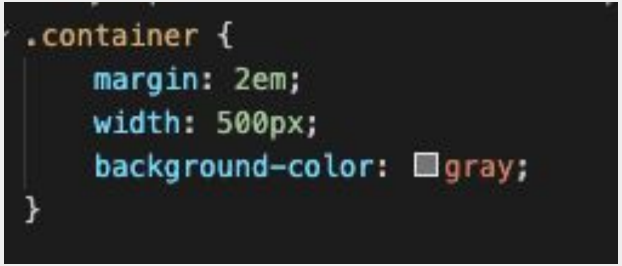
```css
#mycard {
    color: □white;
}
```

Recap of Selectors

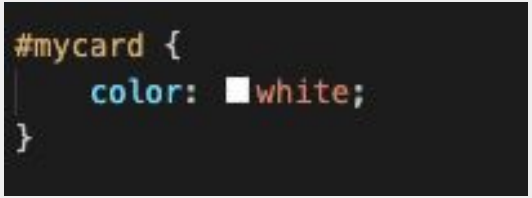
- Type selectors: a, p, h1, div, etc
- Class selectors: .classname
- ID selectors: #idname
- Attribute selectors: [attr=value]



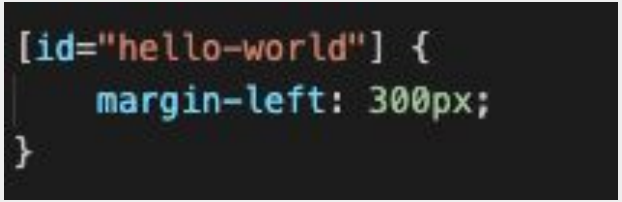
```
a {
    font-size: 10px;
}
```



```
.container {
    margin: 2em;
    width: 500px;
    background-color: ▨gray;
}
```



```
#mycard {
    color: □white;
}
```



```
[id="hello-world"] {
    margin-left: 300px;
}
```

Attribute Selectors

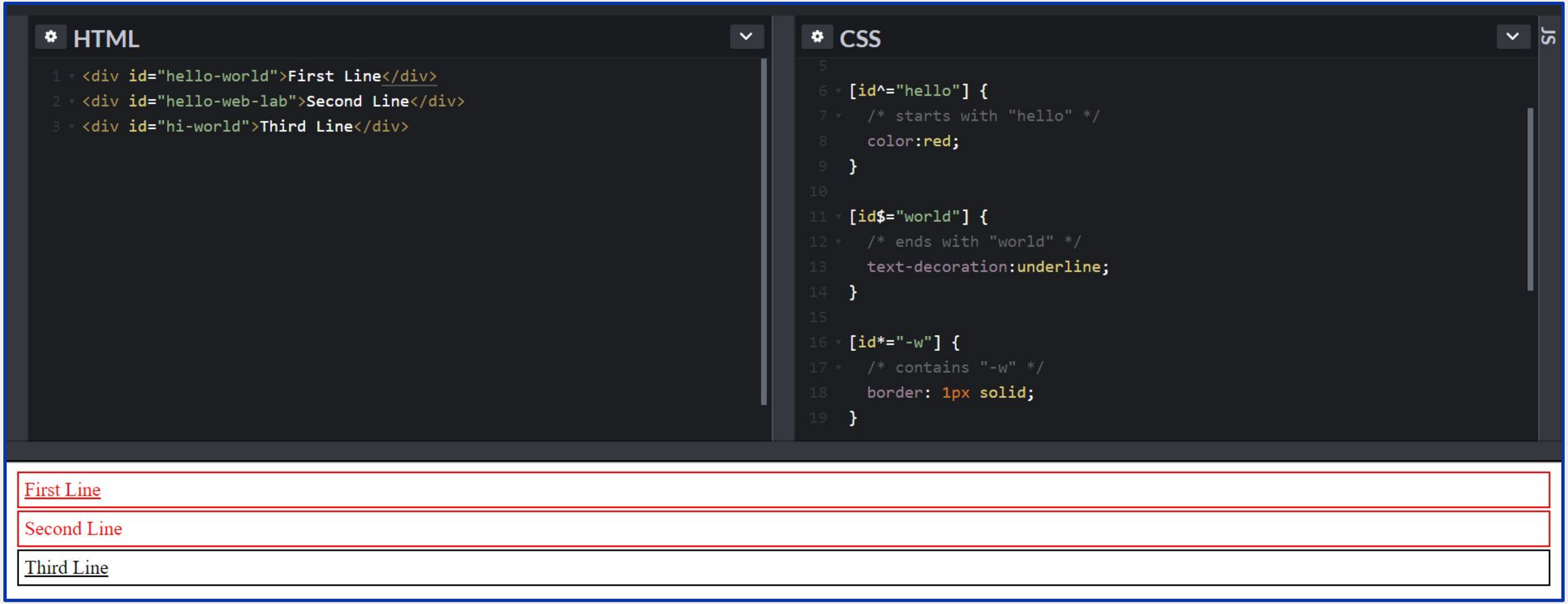
Can find elements based on any attribute (not just class and ID):

```
[attr] { // rules }

[attr=value] { // rules }
```

Can also filter with attributes that start with, end with, or contain a value!

https://developer.mozilla.org/en-US/docs/Web/CSS/Attribute_selectors

Attribute Selectors



```html
<div id="hello-world">First Line</div>
<div id="hello-web-lab">Second Line</div>
<div id="hi-world">Third Line</div>
```

```css

[id^="hello"] {
  /* starts with "hello" */
  color:red;
}

[id$="world"] {
  /* ends with "world" */
  text-decoration:underline;
}

[id*="-w"] {
  /* contains "-w" */
  border: 1px solid;
}
```

First Line

Second Line

Third Line

CSS Combinators

Used to select elements in relation to other elements

- Siblings
- Children
- Descendants

CSS Combinators - Siblings

Adjacent siblings: +

```
h2 + p { // selects all <p> that directly follow an <h2> }
```

General siblings: ~

```
p ~ span { // selects all <span> that follow an <p> }
```

CSS Combinators - Child/Descendant

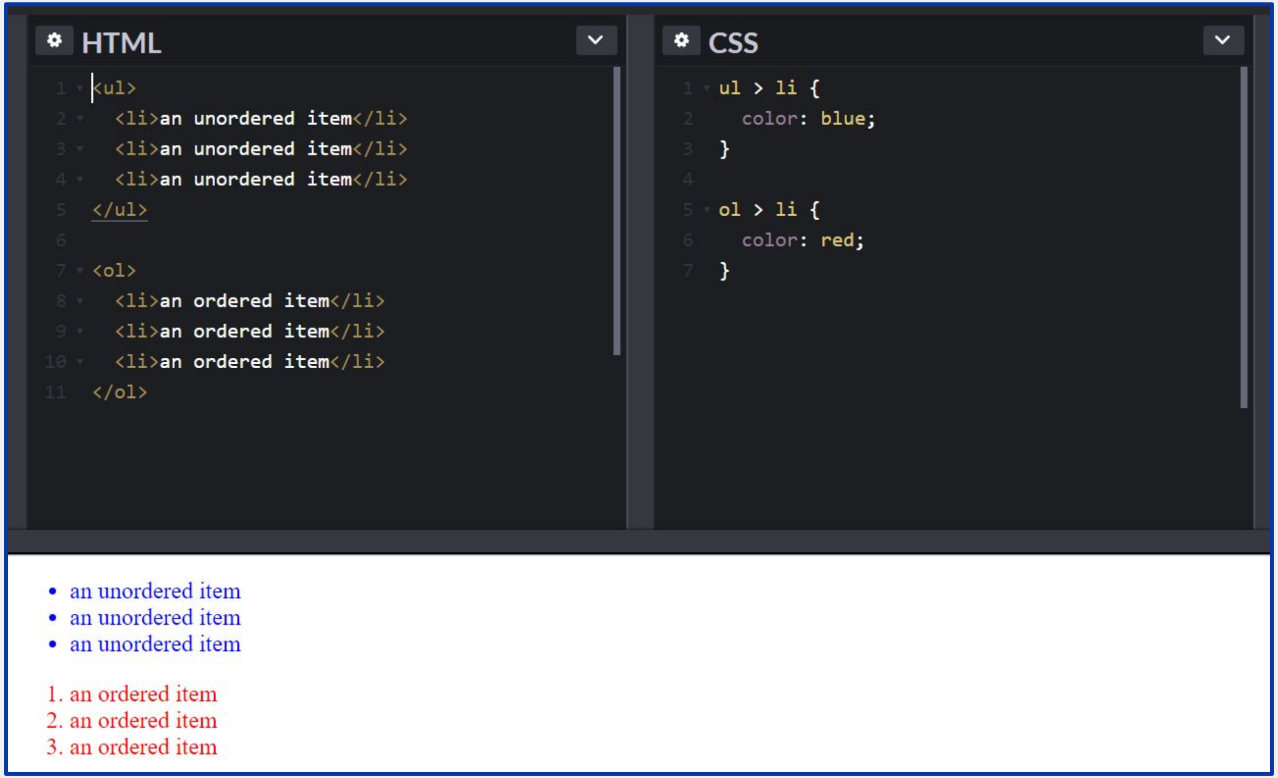
Child: >

```
ul > li { // selects all <li> directly inside a <ul> }
```

Descendant: ‘ ‘ (space)

```
div span { // selects all <span> anywhere inside a <div> }
```

CSS Combinators



HTML

```
<ul>
  <li>an unordered item</li>
  <li>an unordered item</li>
  <li>an unordered item</li>
</ul>

<ol>
  <li>an ordered item</li>
  <li>an ordered item</li>
  <li>an ordered item</li>
</ol>
```

CSS

```
ul > li {
  color: blue;
}

ol > li {
  color: red;
}
```

- an unordered item
- an unordered item
- an unordered item

1. an ordered item
2. an ordered item
3. an ordered item

CSS Pseudo-classes

Used to specify a selector that is not directly represented in the HTML.

You’ve seen one already!

```
element:hover {
    // cool hover rules
}
```

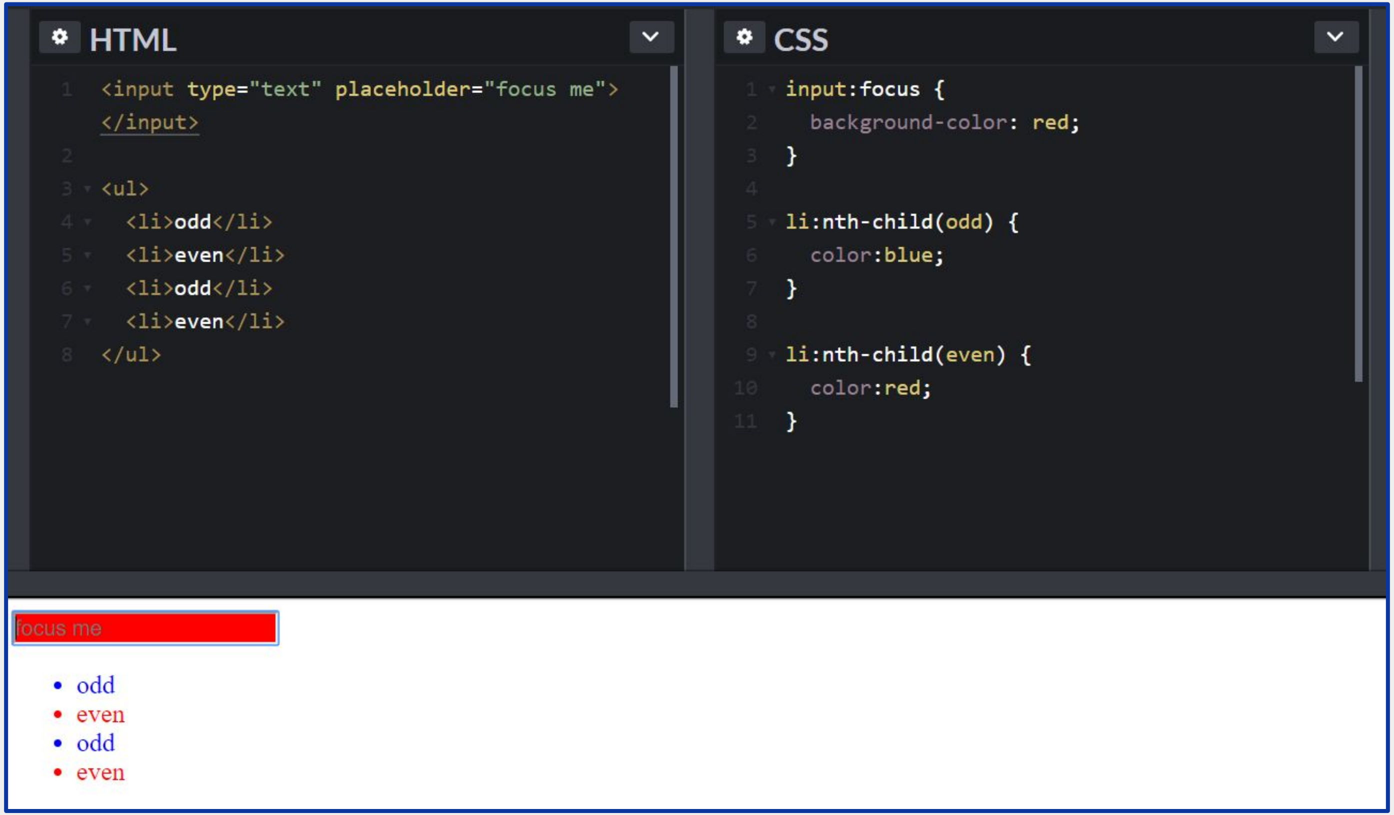
CSS Pseudo-classes

```
a:active { // clicking on a link }

input:focus { // tabbing/clicking on an input }

p:nth-child(4n) { // every 4th <p> }
```

See more at:

https://developer.mozilla.org/en-US/docs/Web/CSS/Pseudo-classes

CSS Pseudo-classes



```html
<input type="text" placeholder="focus me">
</input>

<ul>
  <li>odd</li>
  <li>even</li>
  <li>odd</li>
  <li>even</li>
</ul>
```

```css
input:focus {
  background-color: red;
}

li:nth-child(odd) {
  color:blue;
}

li:nth-child(even) {
  color:red;
}
```

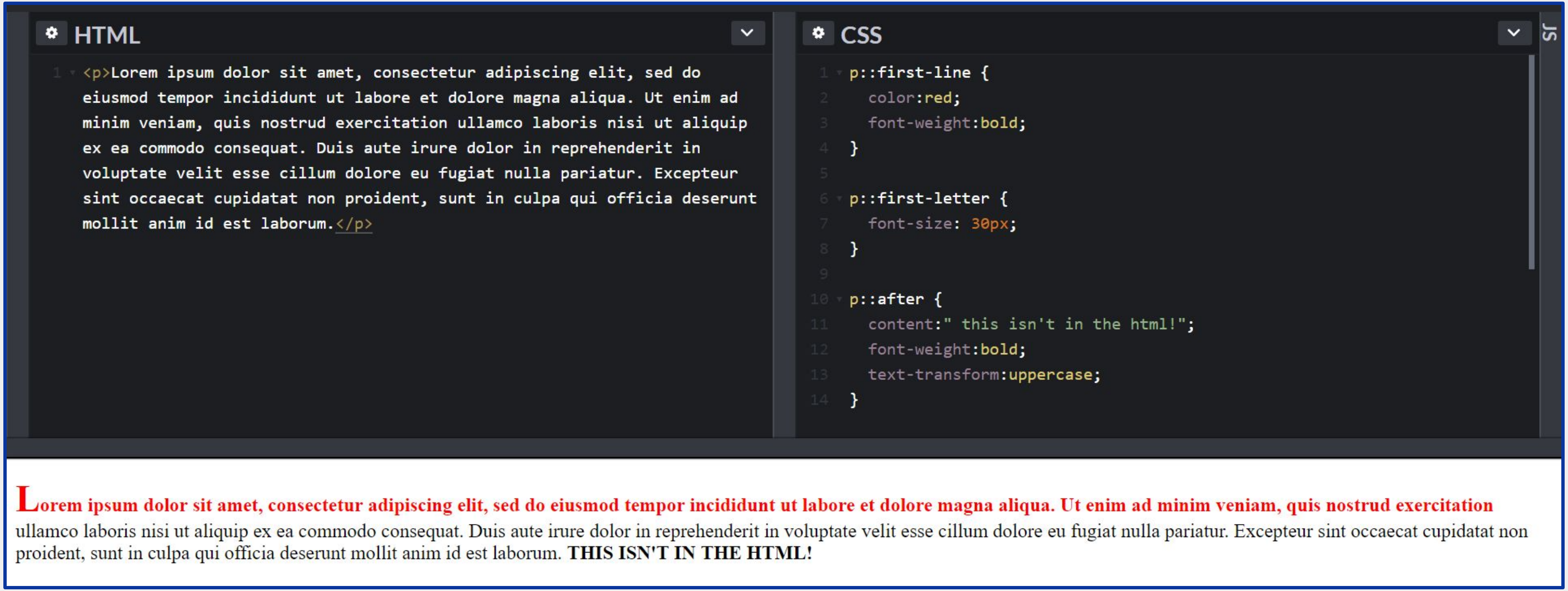
```
focus me

• odd
• even
• odd
• even
```

CSS Pseudo-elements

Used to style specific parts of selected elements.

```
p::first-line { // first line of a <p> }

::selection { // user highlighted text }



a::after {
   // styling placed in a generated element after every <a>
}
```

CSS Pseudo-elements



```html
<p>Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat. Duis aute irure dolor in reprehenderit in voluptate velit esse cillum dolore eu fugiat nulla pariatur. Excepteur sint occaecat cupidatat non proident, sunt in culpa qui officia deserunt mollit anim id est laborum.</p>
```

```css
p::first-line {
  color:red;
  font-weight:bold;
}

p::first-letter {
  font-size: 30px;
}

p::after {
  content:" this isn't in the html!";
  font-weight:bold;
  text-transform:uppercase;
}
```

Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat. Duis aute irure dolor in reprehenderit in voluptate velit esse cillum dolore eu fugiat nulla pariatur. Excepteur sint occaecat cupidatat non proident, sunt in culpa qui officia deserunt mollit anim id est laborum. **THIS ISN'T IN THE HTML!**

let’s look at some code

is that all?

Nope. but that’s all for today <3