

Practical CSS Transforms

Jing Jin

Co-founder, Konsult

jing@konsu.it

github.com/jingidy/practical-css-transforms-2014

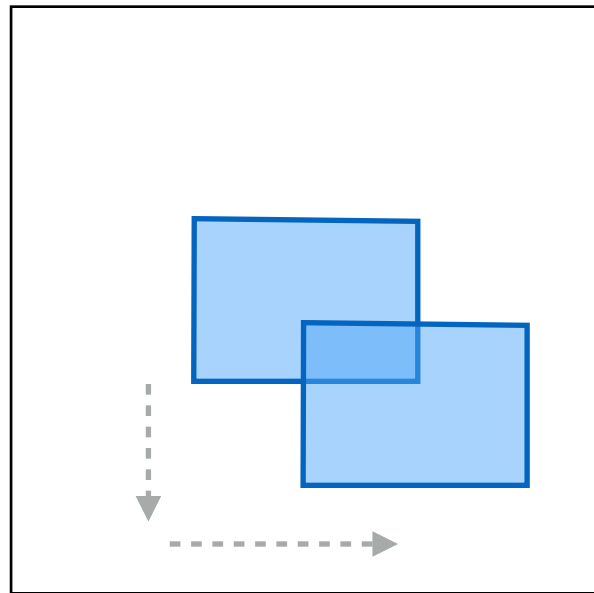
What
Support
Benefits
Code!
Gotchas

What

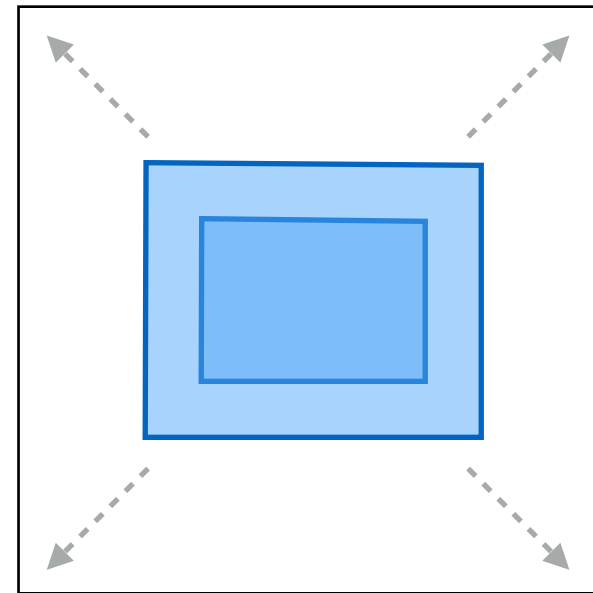
2D & 3D

2D

translate

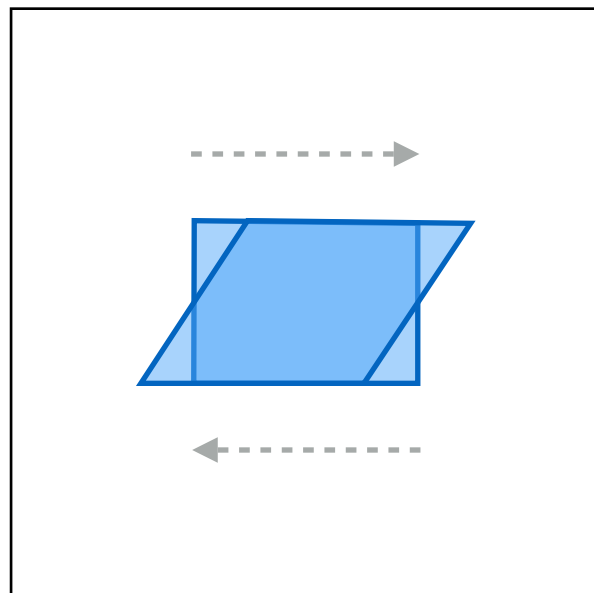


scale

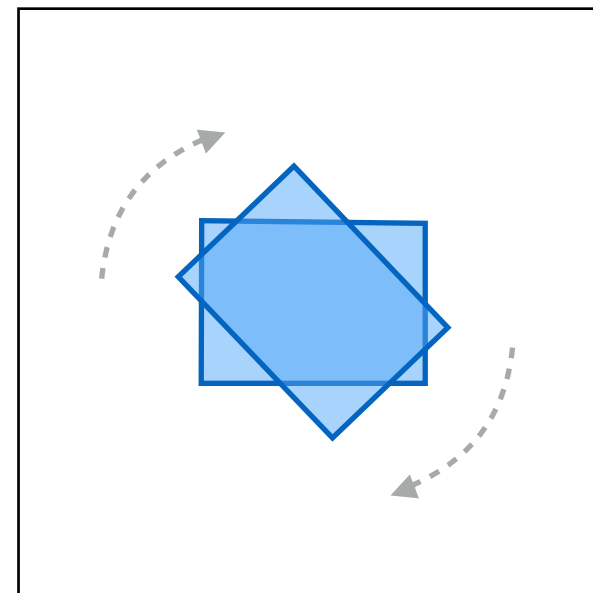


2D

skew



rotate



The Simpsons in CSS

Below are some Simpsons characters made in pure CSS - Made by [Chris Pattle](#).

(Best viewed in Chrome)

[View on Github](#)

[Follow me on Twitter](#)



Homer

[View CSS](#)



Bart

[View CSS](#)



Marge

[View CSS](#)



Lisa

[View CSS](#)



SITE of
the DAY

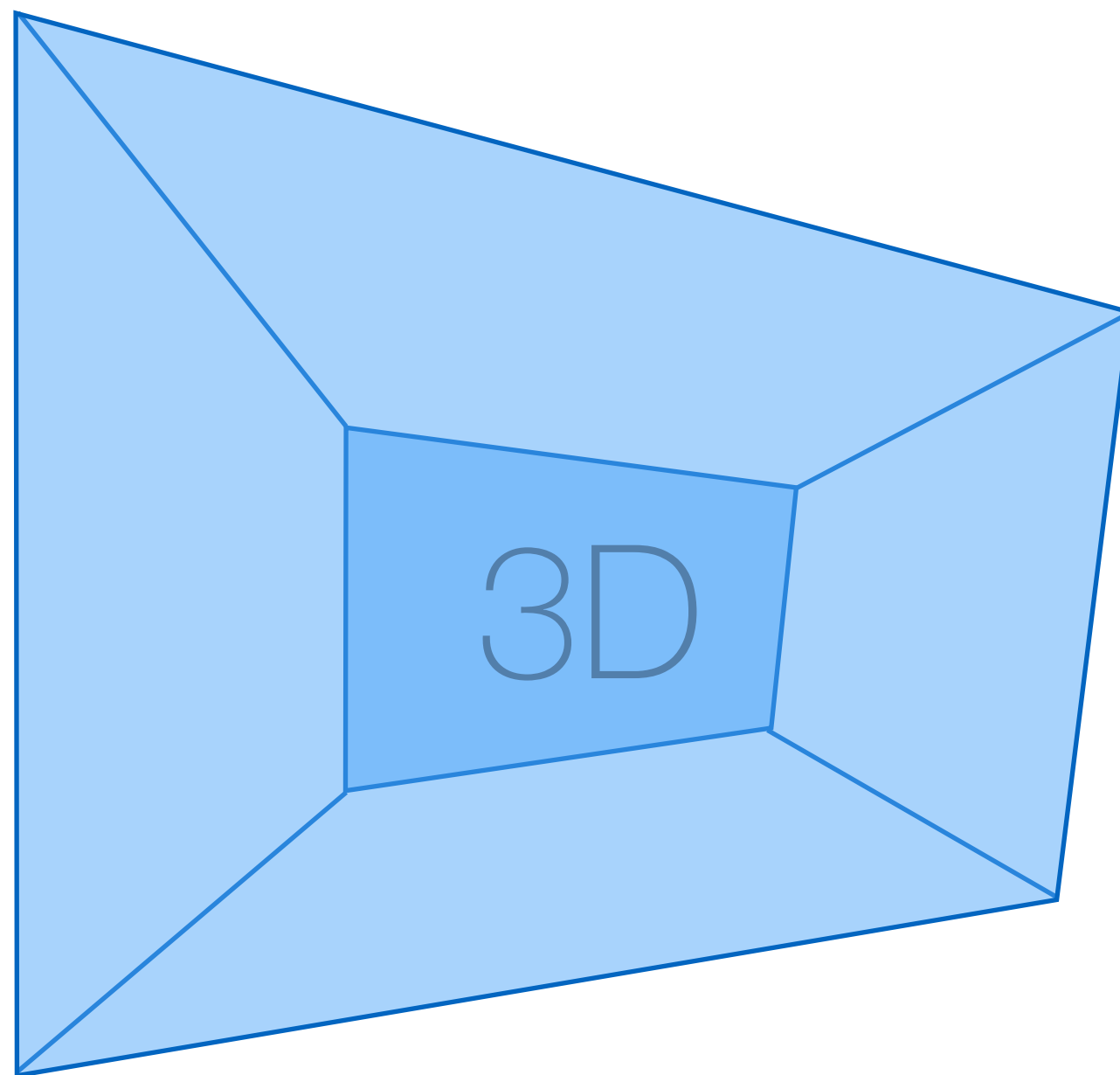
parallax.js



<http://matthew.wagerfield.com/parallax/>

Like 433 Tweet 5,452

3D



Transitions

Cube
Rotate In
Horizontal Flip
Multi-Flip
Dissolve
Unfold
Toss
Slide In
Iris
Fade Through



Watch it again

FWB  THE CUTTING
EDGE AWARD

WELCOME TO THE DIGITAL CREATIVITY VILLAGE.

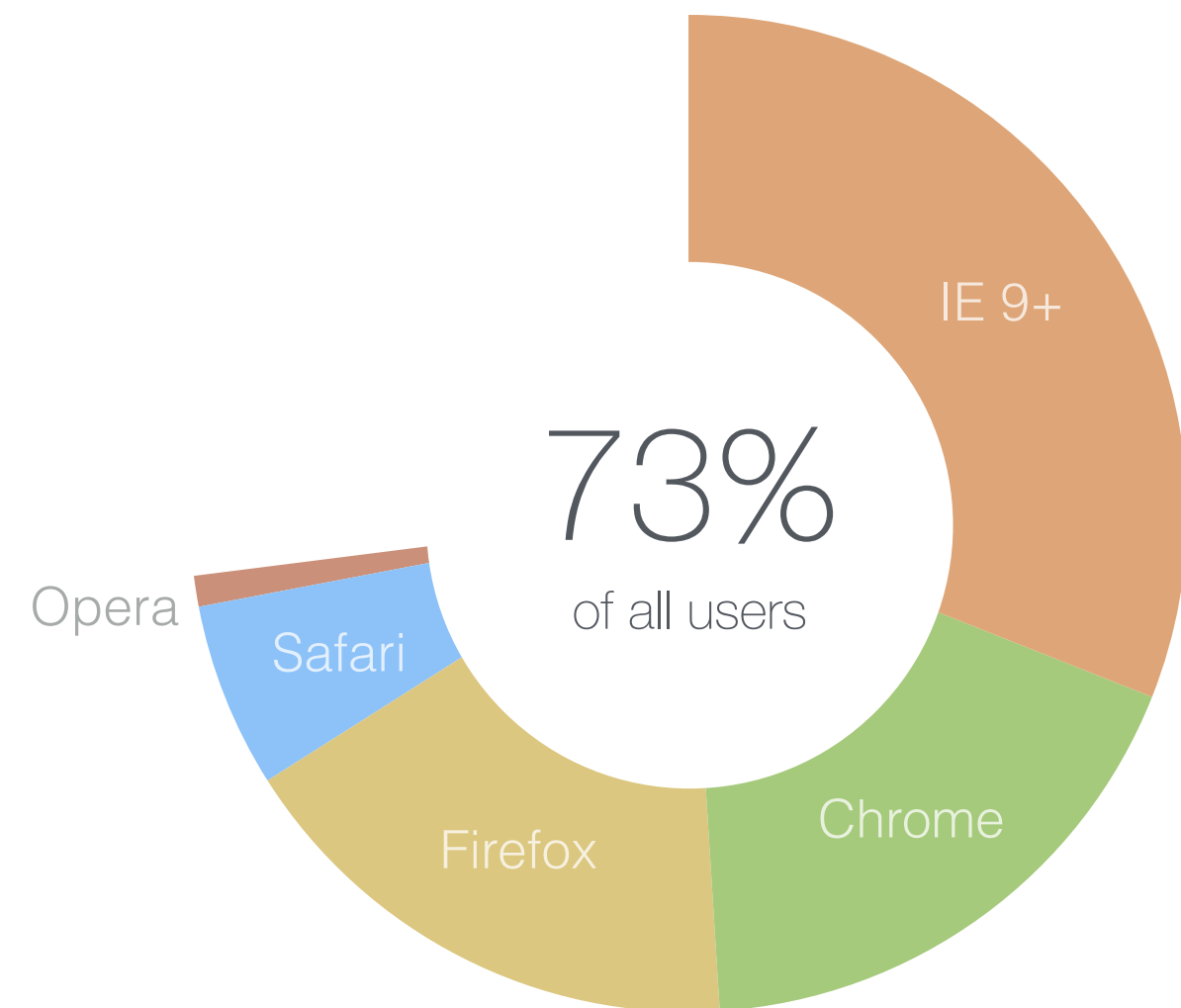
CLICK. DRAG. PULL.
FIND NEW CREATIVE OPPORTUNITIES FROM GOOGLE



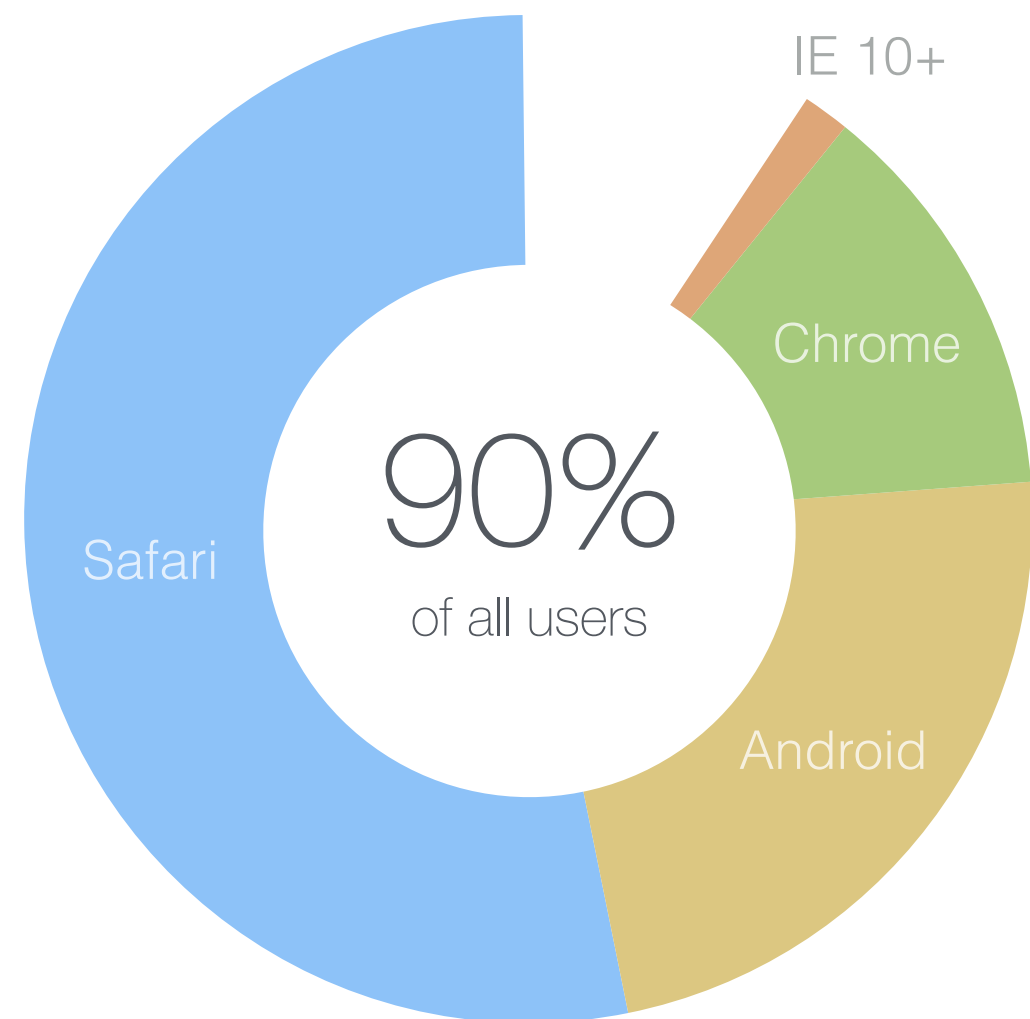
Browser Support

2D Transform Support

Desktop

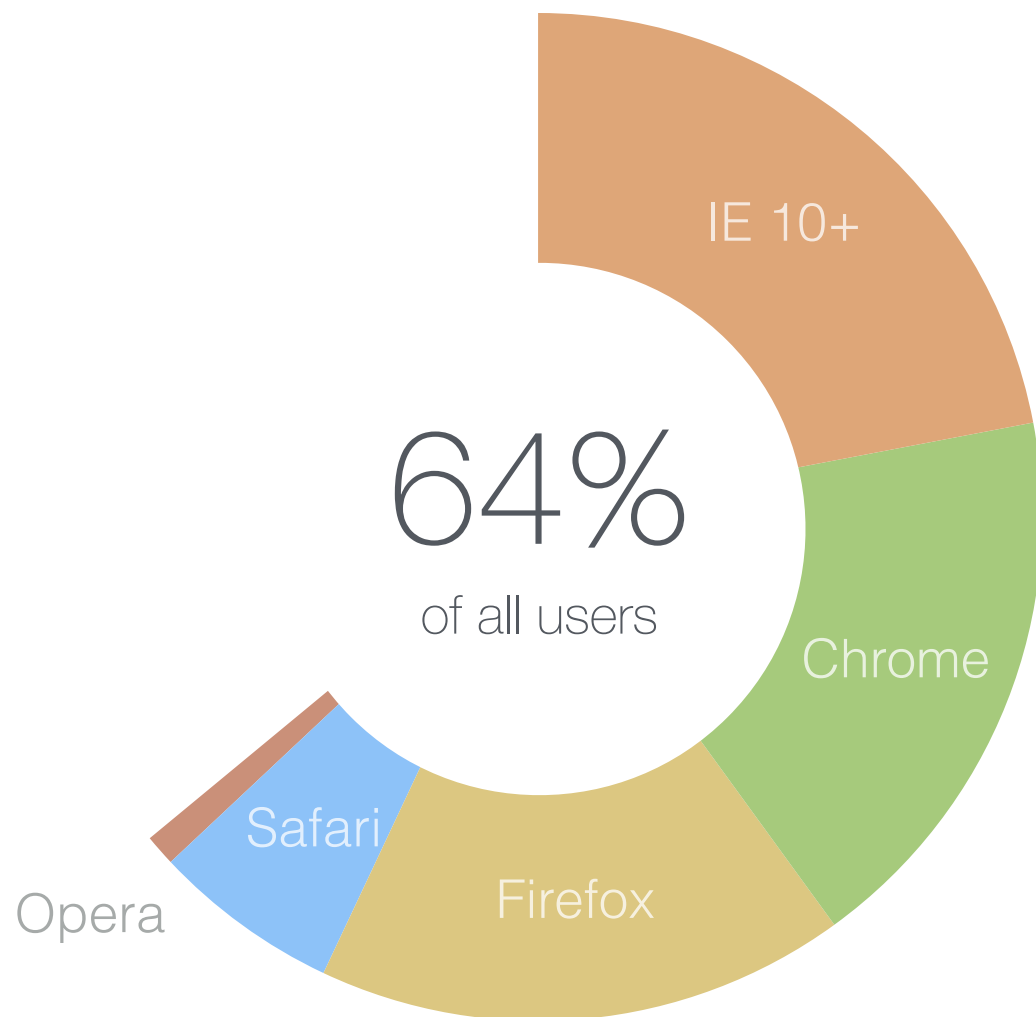


Mobile

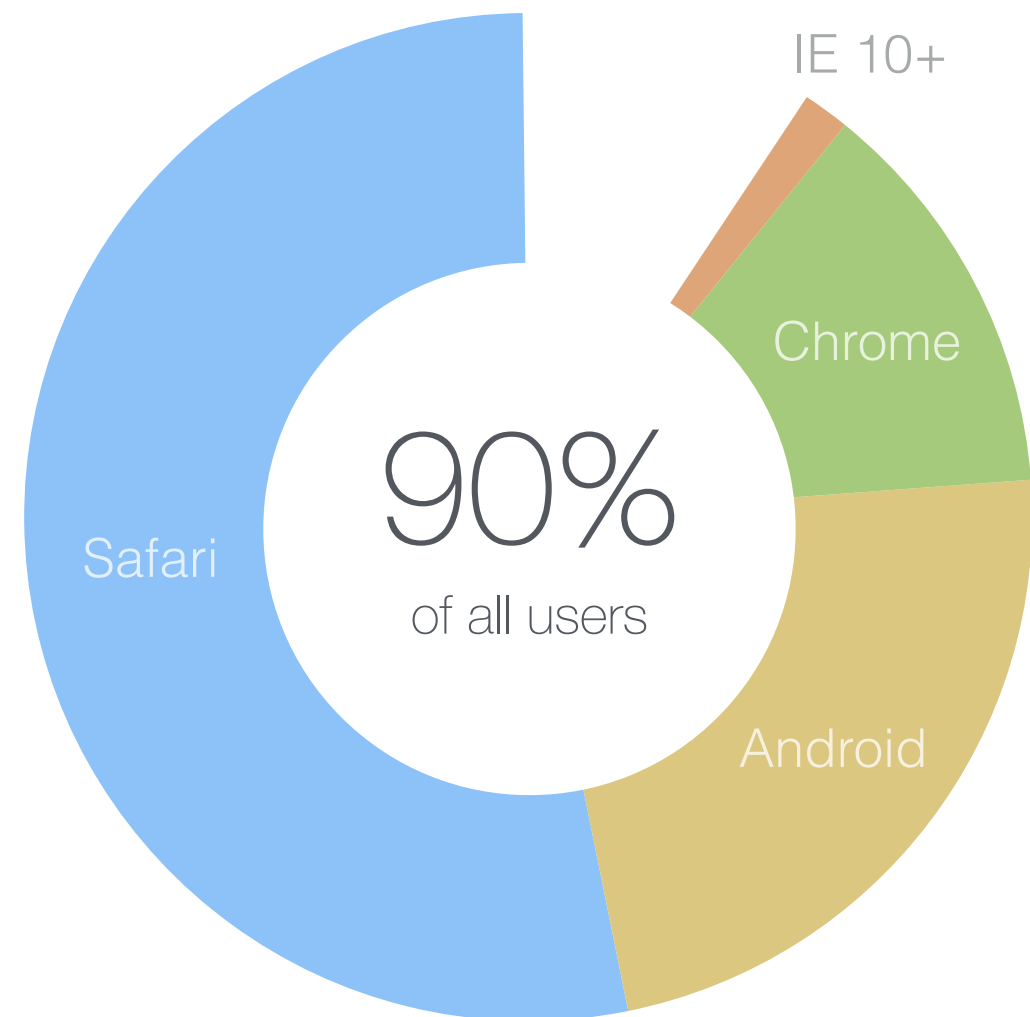


3D Transform Support

Desktop



Mobile



3D Transform Support

Rule of thumb: use with graceful degradation

Benefits

Resolution Independent

2x Image

Send Message

1x

Send Message

2x Transforms

Send Message

Smooth Animations

```
@keyframes pos {  
  0% {  
    left: 0;  
  }  
  100% {  
    left: 50px;  
  }  
}
```

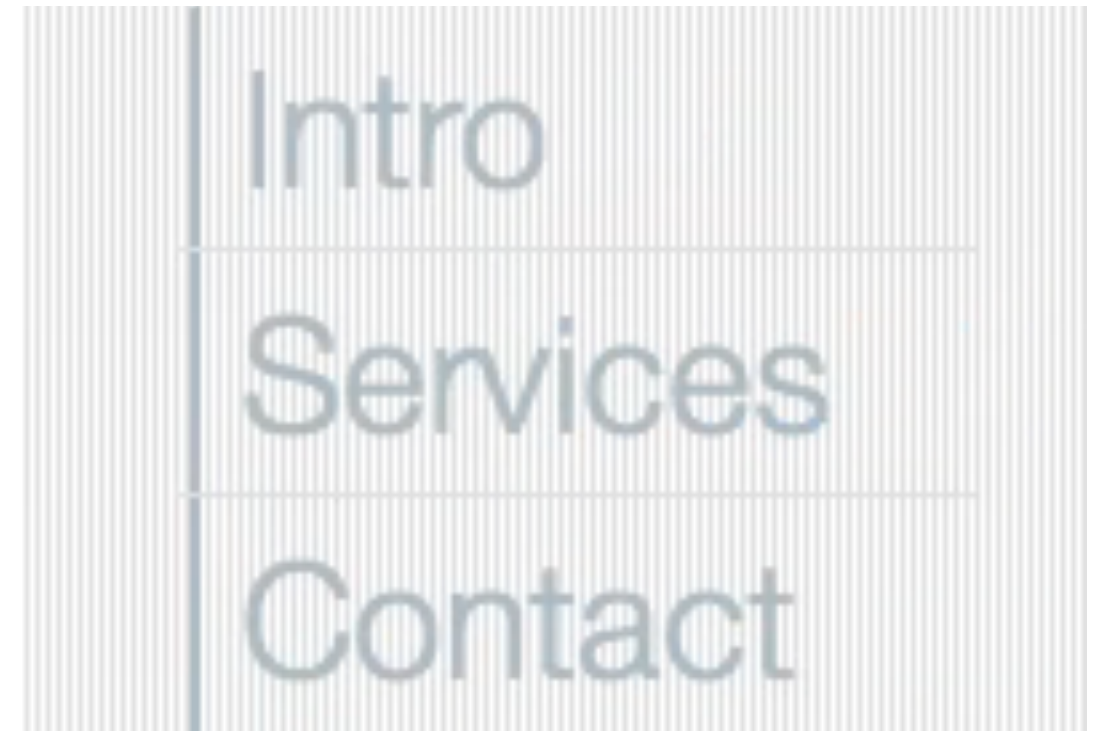
```
@keyframes pos {  
  0% {  
    left: 0;  
  }  
  100% {  
    left: 50px;  
  }  
}
```

```
@keyframes trans {  
  0% {  
    transform: none;  
  }  
  100% {  
    transform:  
      translateX(50px);  
  }  
}
```

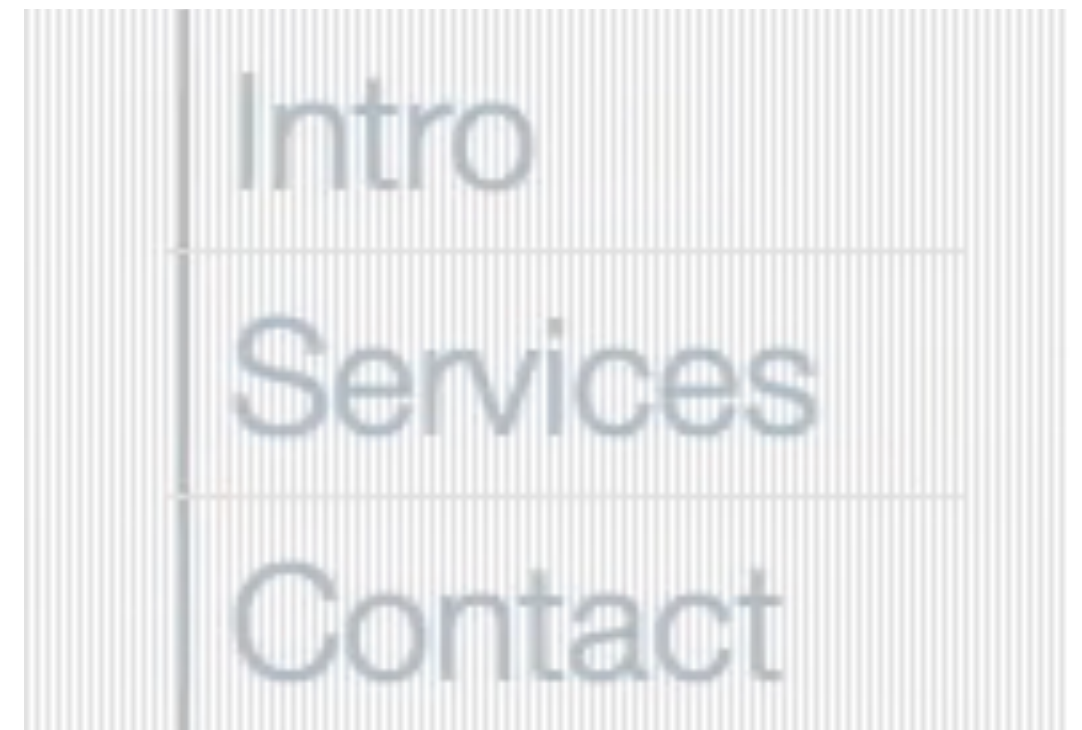
```
@keyframes pos {  
  0% {  
    left: 0;  
  }  
  100% {  
    left: 50px;  
  }  
}
```

```
@keyframes trans {  
  0% {  
    transform: none;  
  }  
  100% {  
    transform:  
      translateX(50px);  
  }  
}
```


Snapped to pixel boundaries



Sub-pixel interpolation

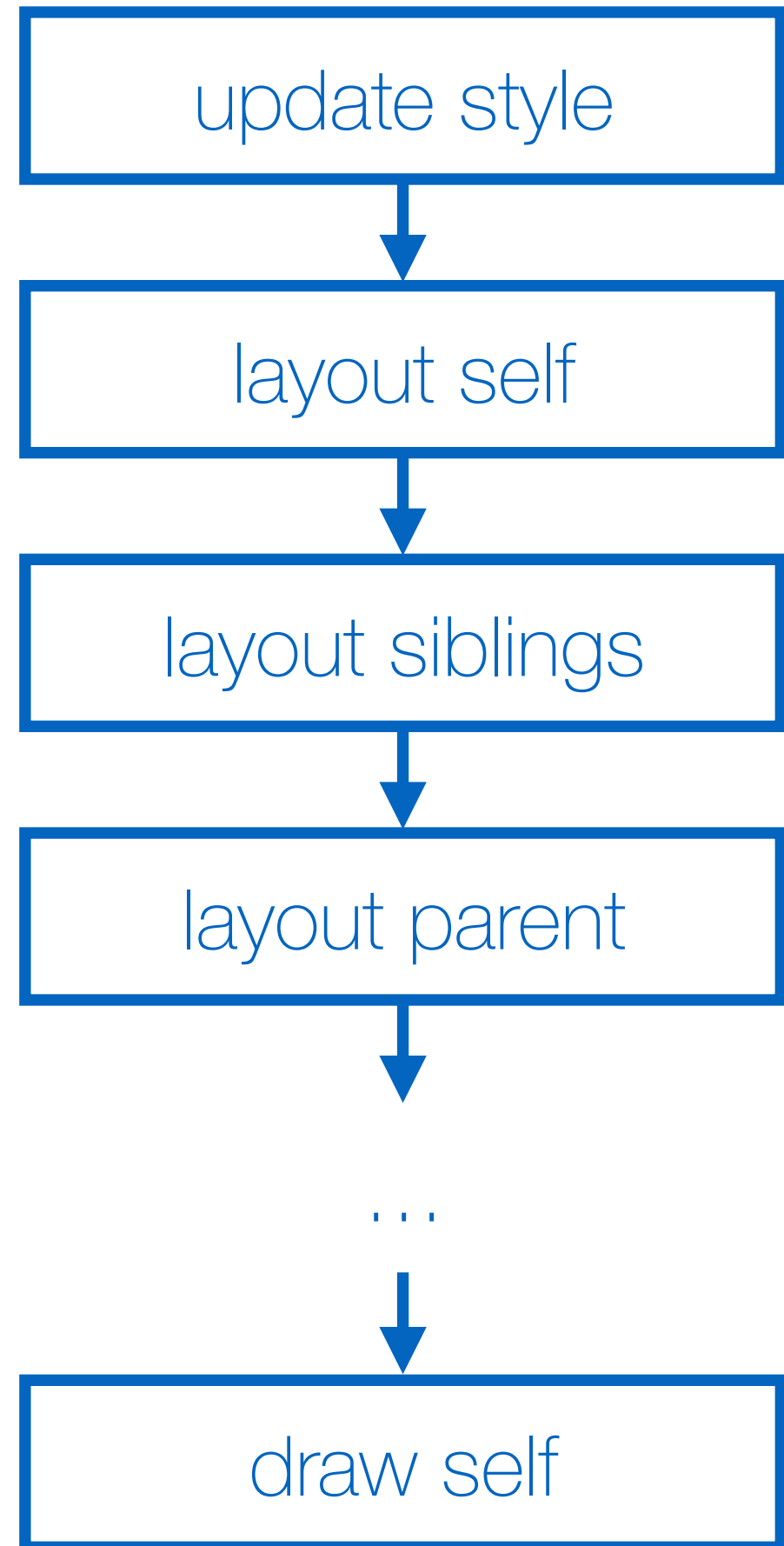


```
@keyframes pos {  
  0% {  
    left: 0;  
  }  
  100% {  
    left: 50px;  
  }  
}
```

```
@keyframes trans {  
  0% {  
    transform: none;  
  }  
  100% {  
    transform:  
      translateX(50px);  
  }  
}
```

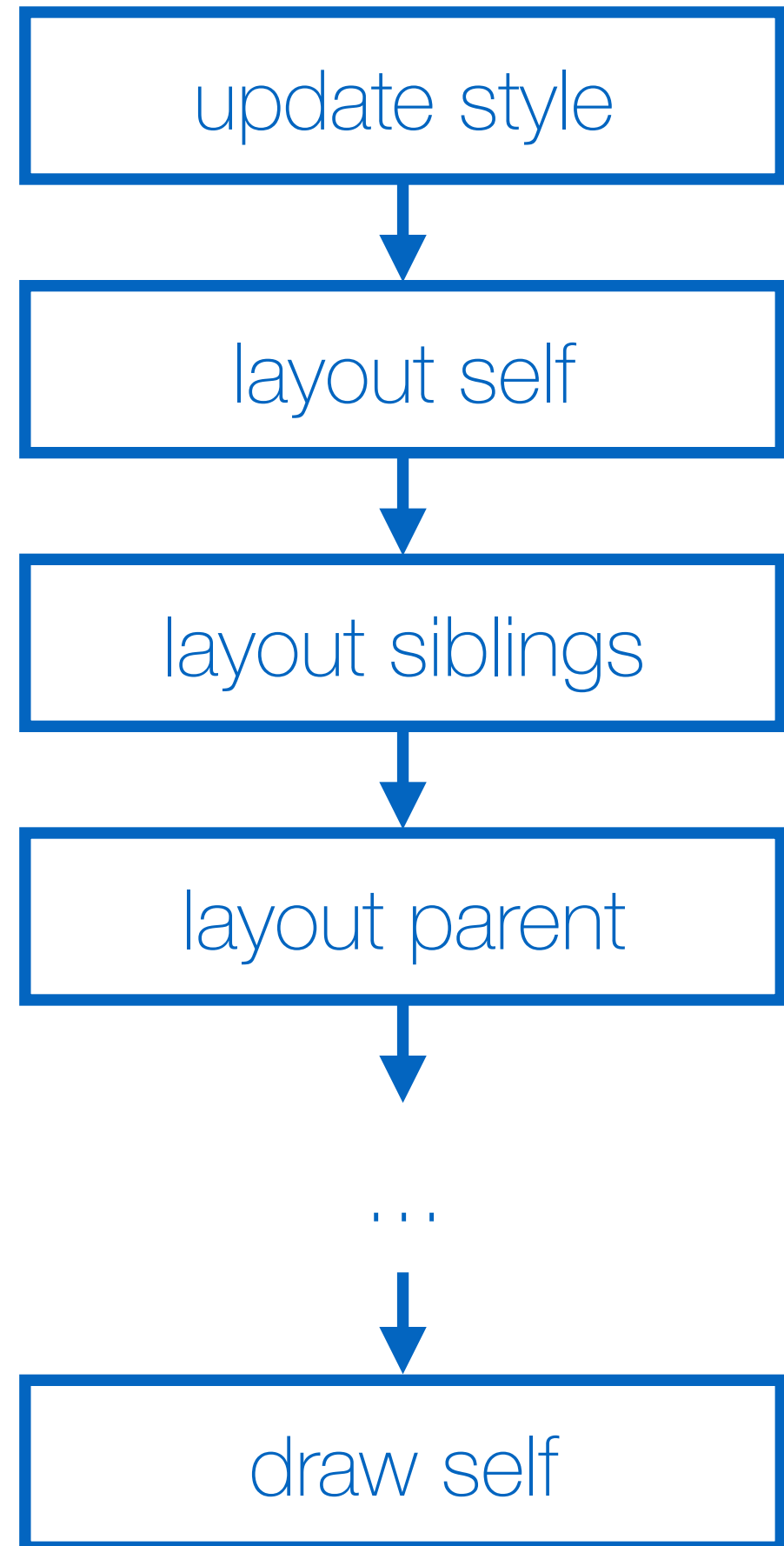
```
@keyframes pos {  
  0% {  
    left: 0;  
  }  
  100% {  
    left: 50px;  
  }  
}
```

```
@keyframes trans {  
  0% {  
    transform: none;  
  }  
  100% {  
    transform:  
      translateX(50px);  
  }  
}
```



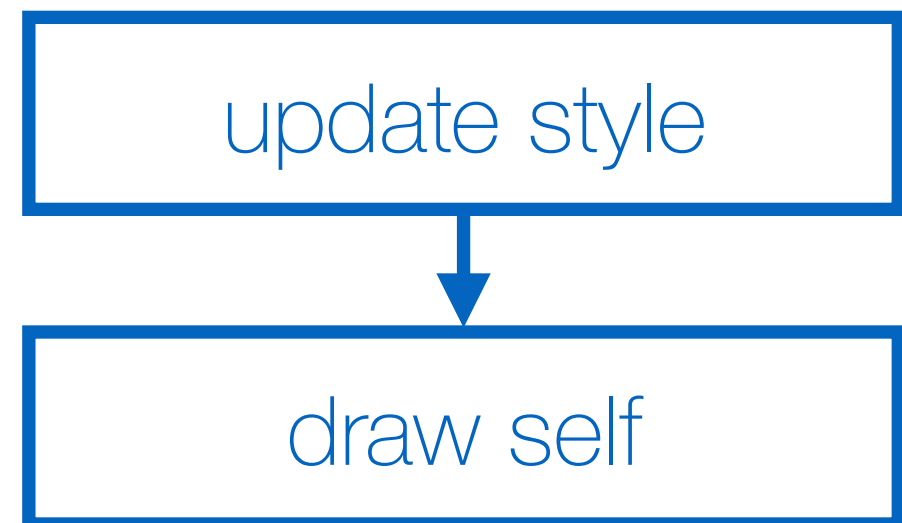
```
@keyframes pos {  
  0% {  
    left: 0;  
  }  
  100% {  
    left: 50px;  
  }  
}
```

```
@keyframes trans {  
  0% {  
    transform: none;  
  }  
  100% {  
    transform:  
      translateX(50px);  
  }  
}
```



```
@keyframes pos {  
  0% {  
    left: 0;  
  }  
  100% {  
    left: 50px;  
  }  
}
```

```
@keyframes trans {  
  0% {  
    transform: none;  
  }  
  100% {  
    transform:  
      translateX(50px);  
  }  
}
```



```
@keyframes pos {
```

```
  0% {  
    left: 0;  
  }
```

```
  100% {  
    left: 50px;  
  }
```

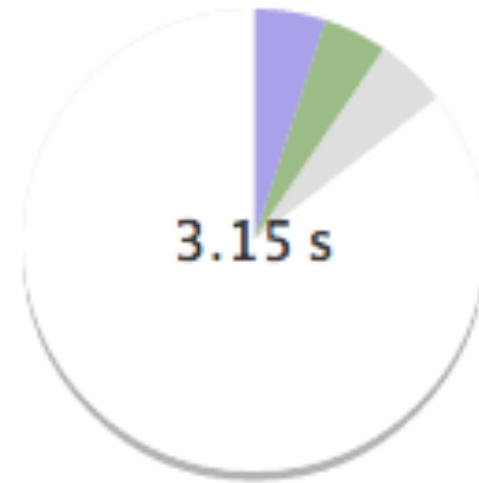
```
}
```

```
@keyframes trans {
```

```
  0% {  
    transform: none;  
  }
```

```
  100% {  
    transform:  
      translateX(50px);  
  }
```

```
}
```

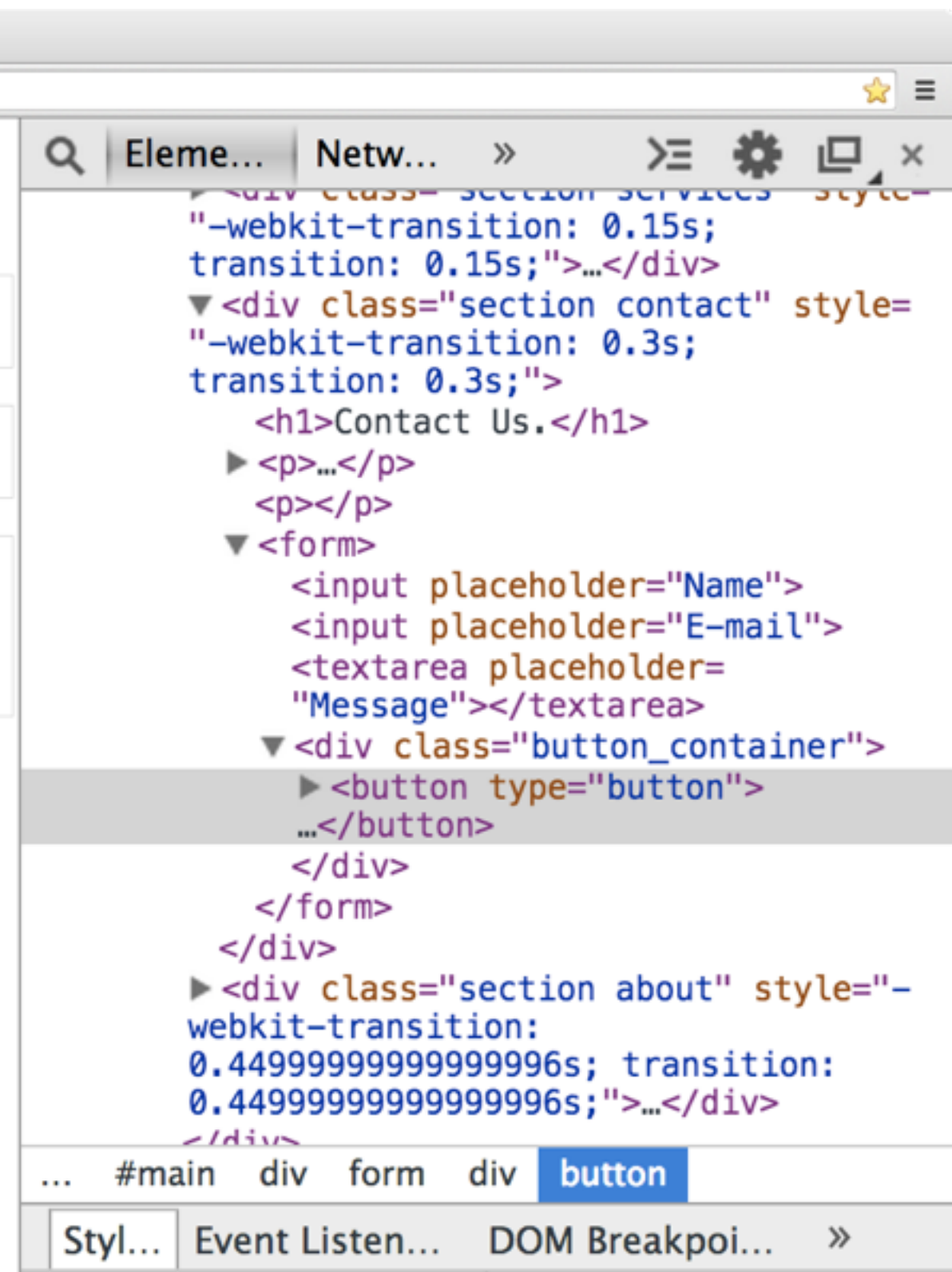
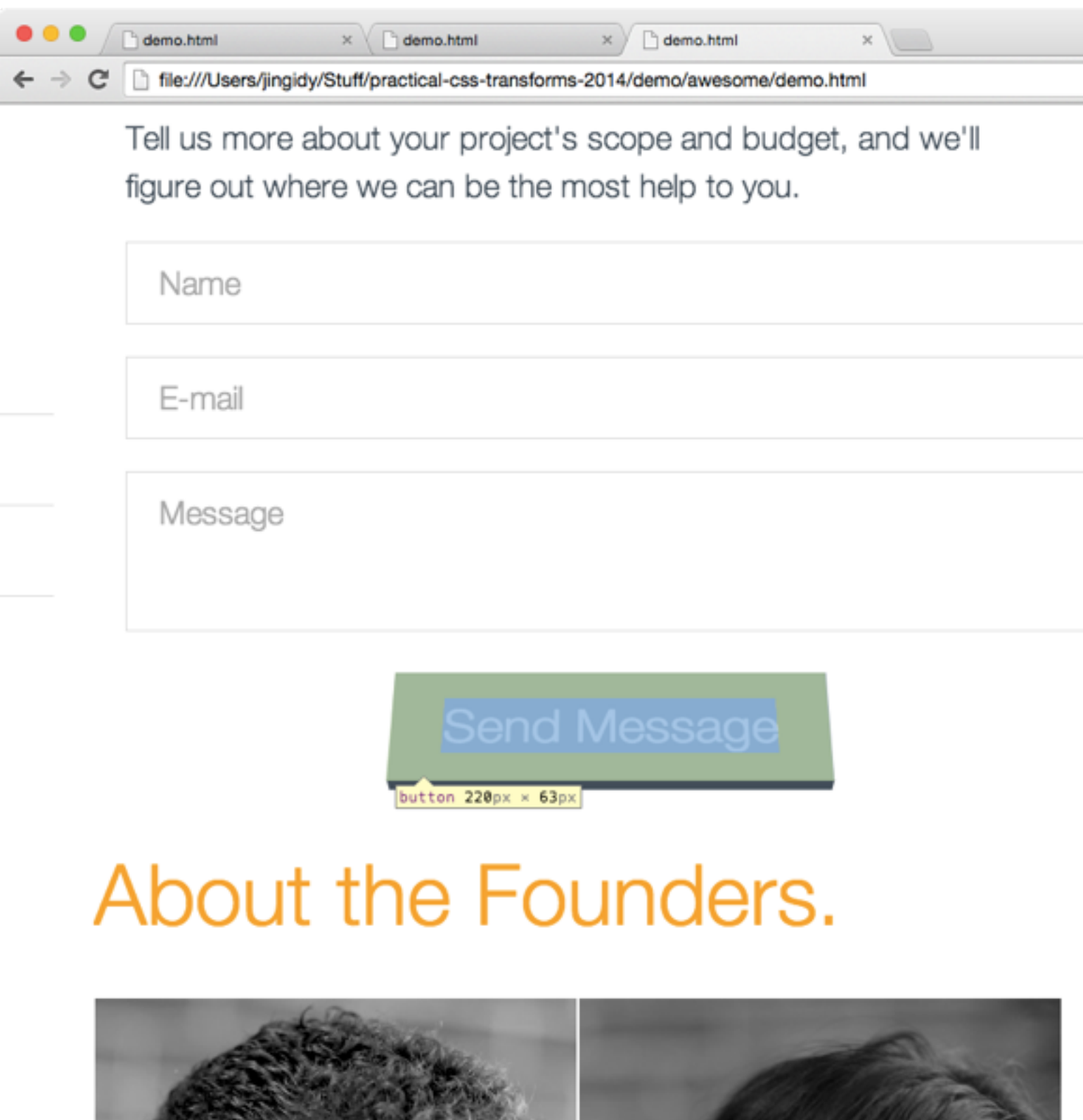


157.934 ms Rendering
141.581 ms Painting
157.215 ms Other
2.70 s Idle



8.415 ms Scripting
2.198 ms Rendering
3.350 ms Painting
15.885 ms Other
3.12 s Idle

Hit-testing & Accessibility

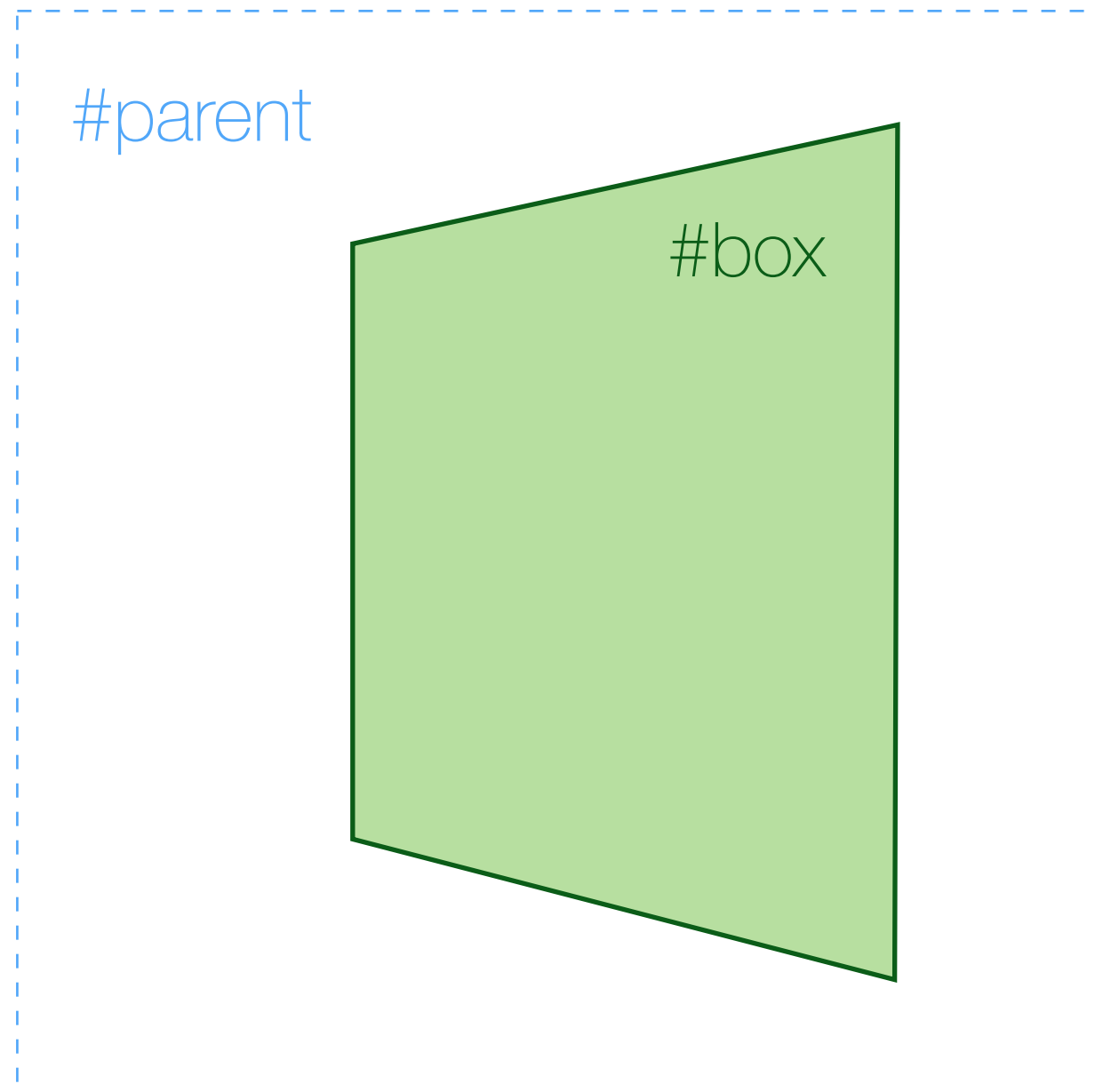


Why

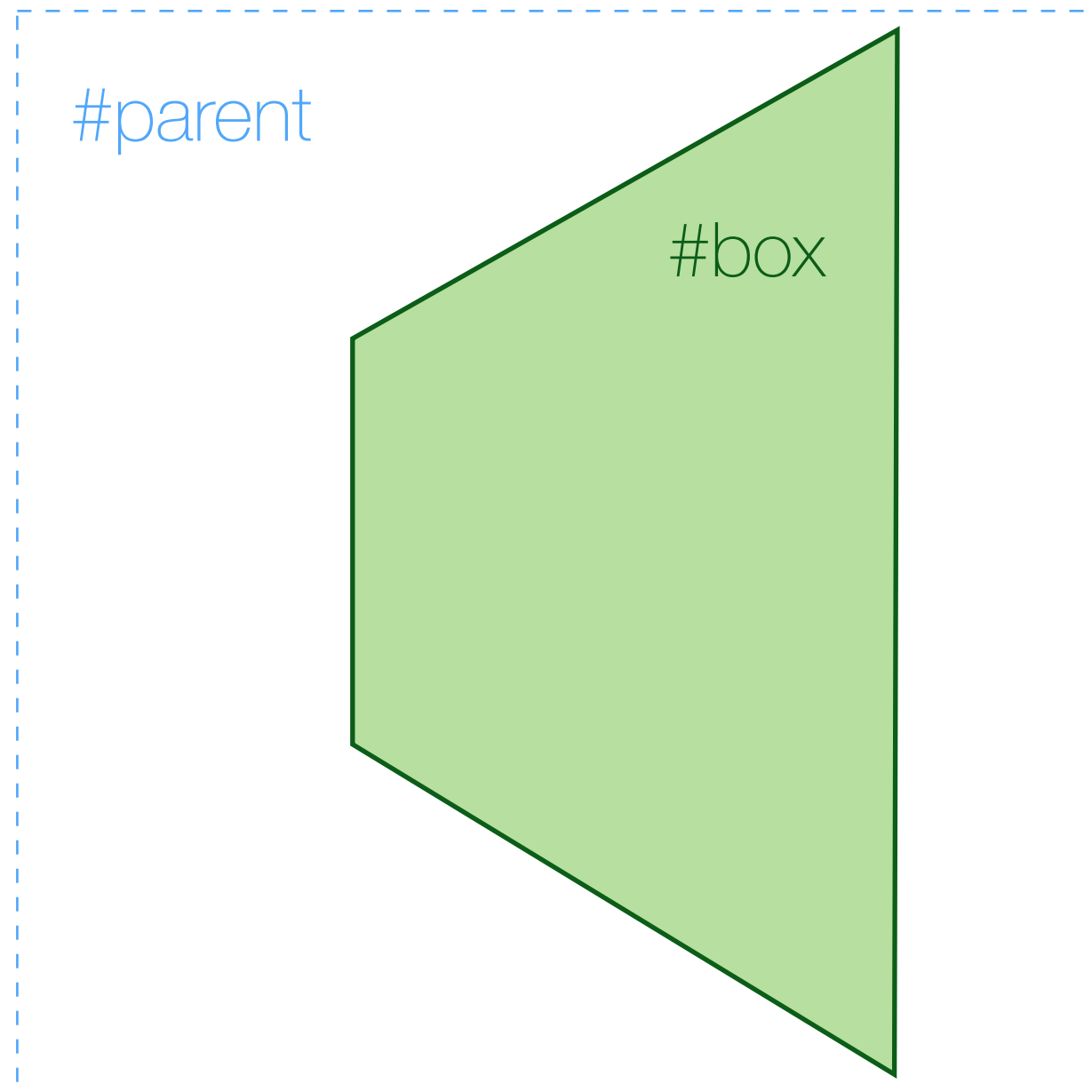
- Resolution independent
- Smooth & fast animations
- Easy hit-testing & accessibility

Code!

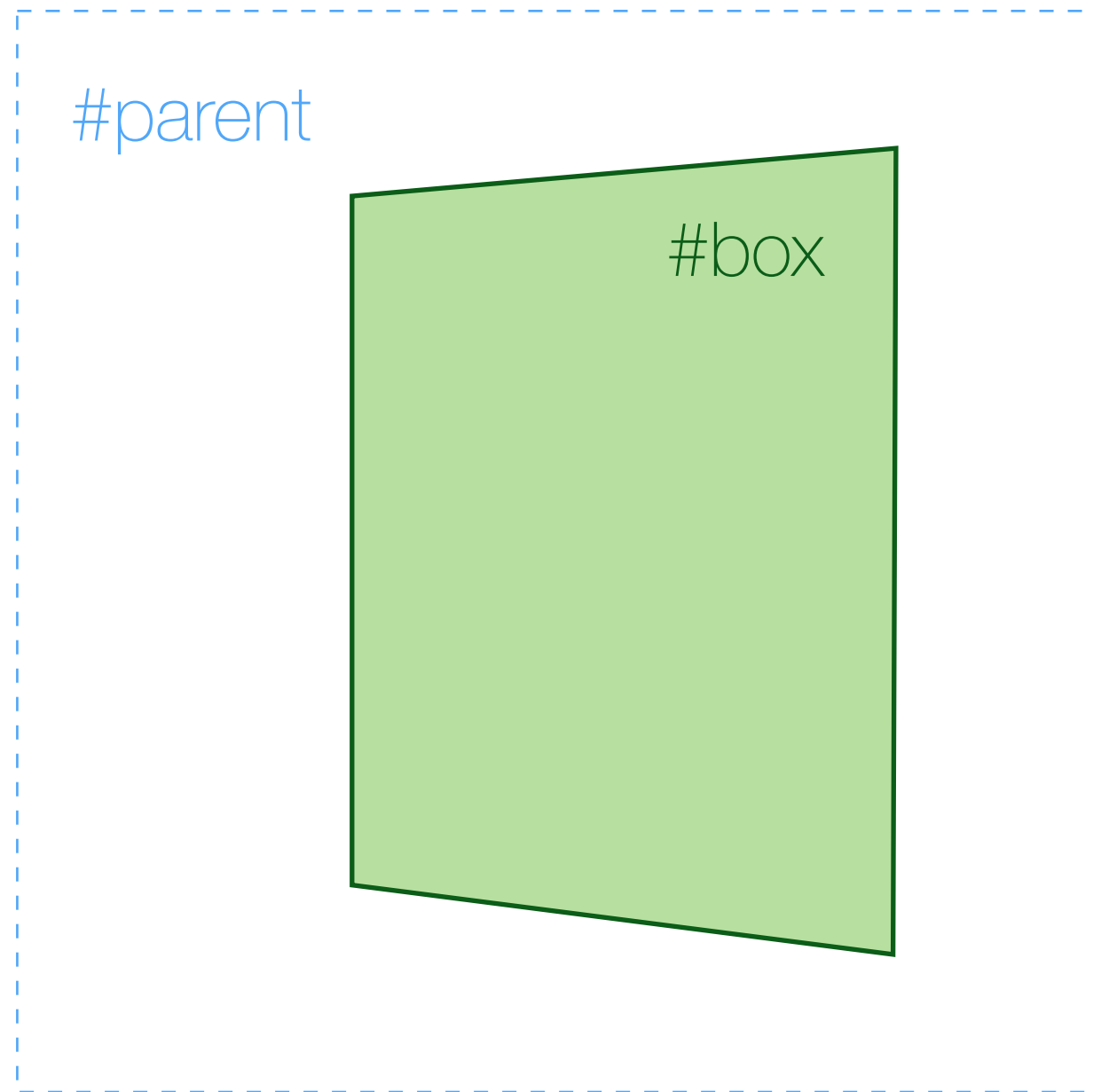
perspective



small perspective

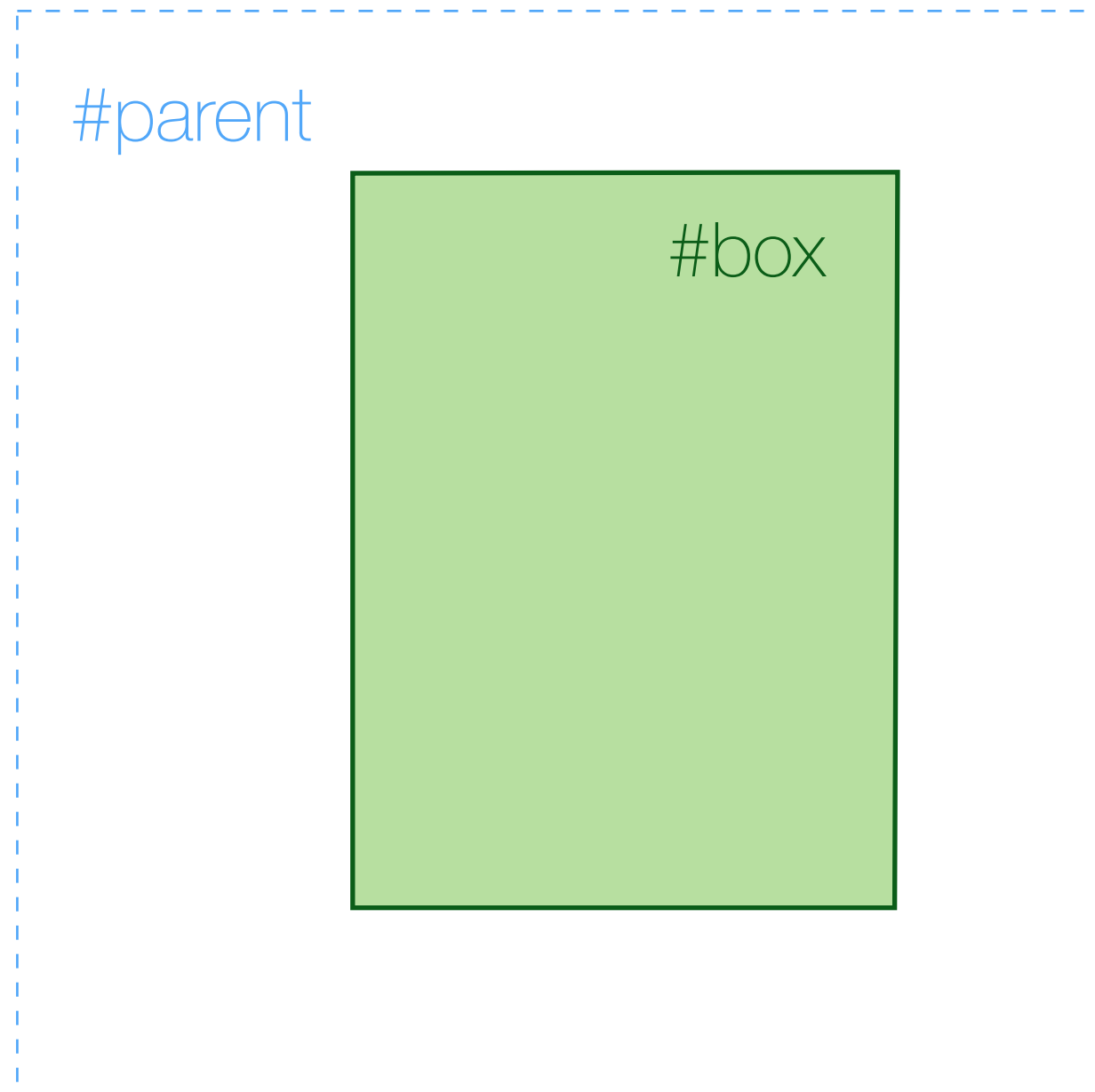


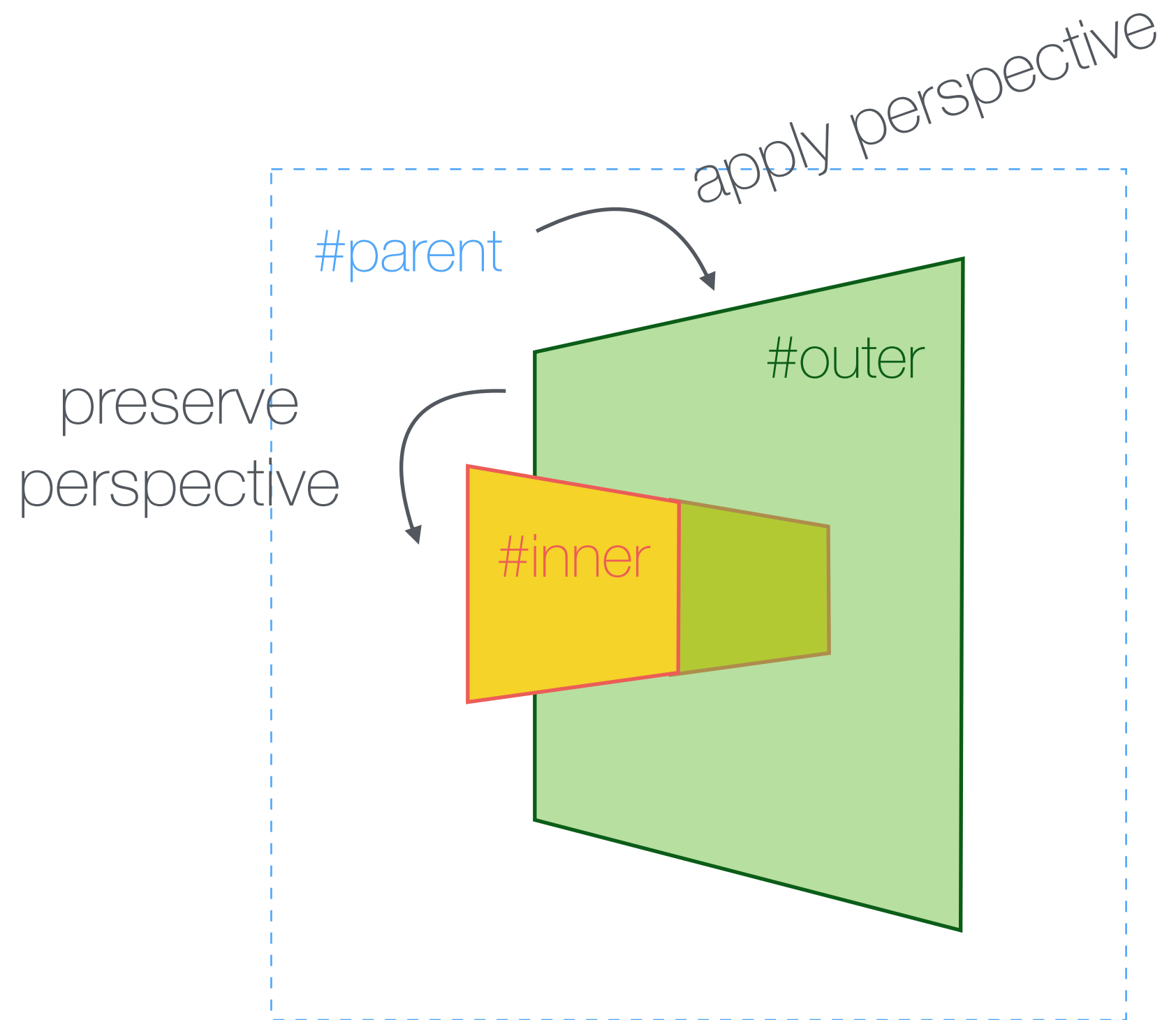
large perspective

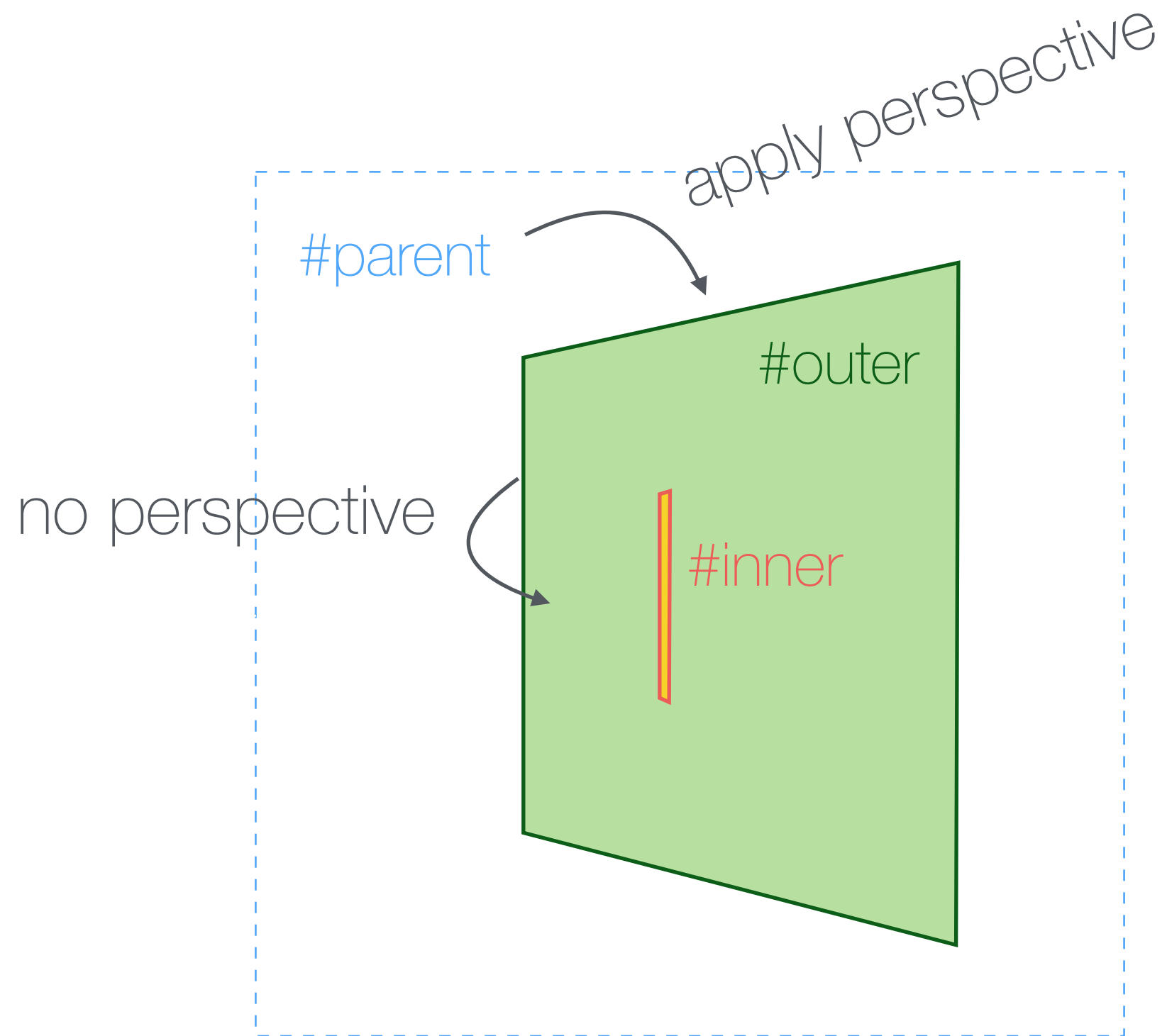


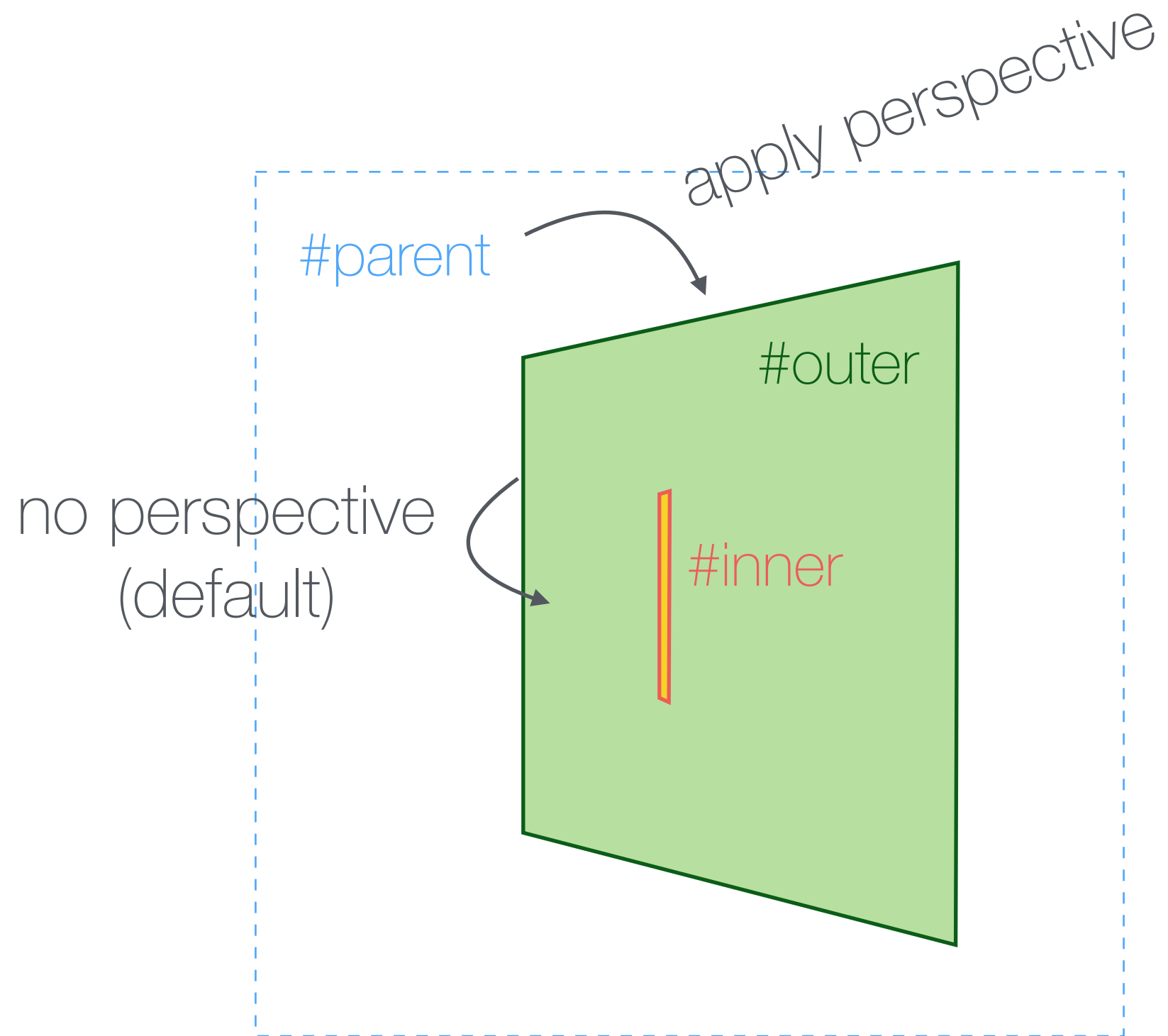
infinite perspective

default
value







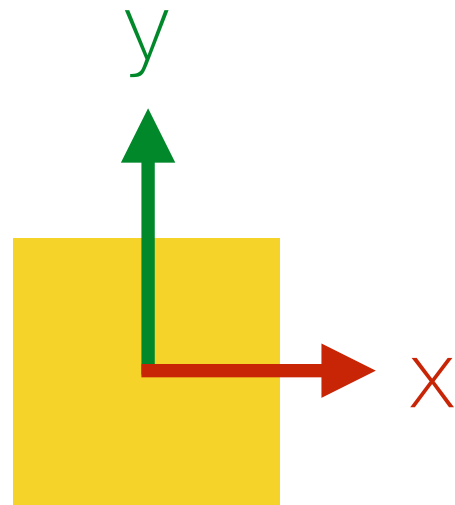


Gotchas

Gotchas

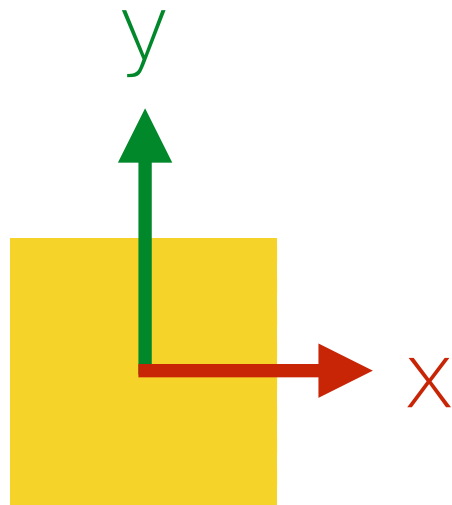
- Order of operation matters

Gotchas



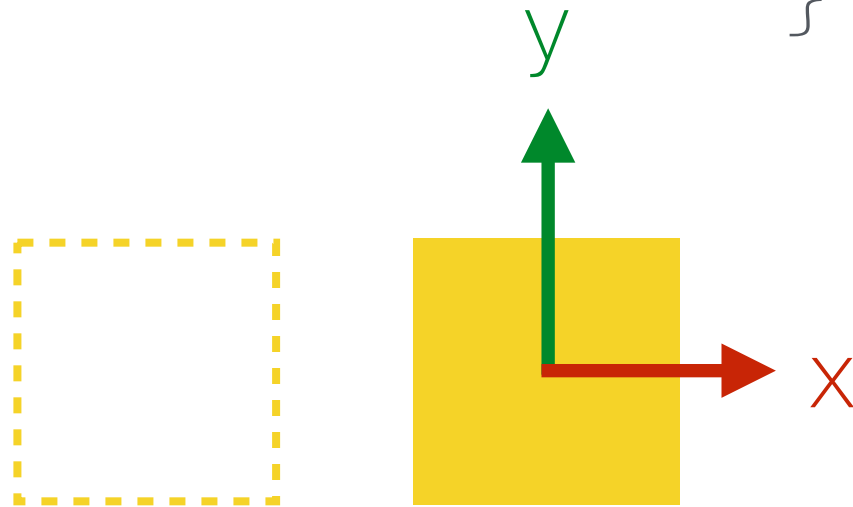
Gotchas

```
#box {  
  transform:  
    translateX(20px)  
    rotate(45deg);  
}
```



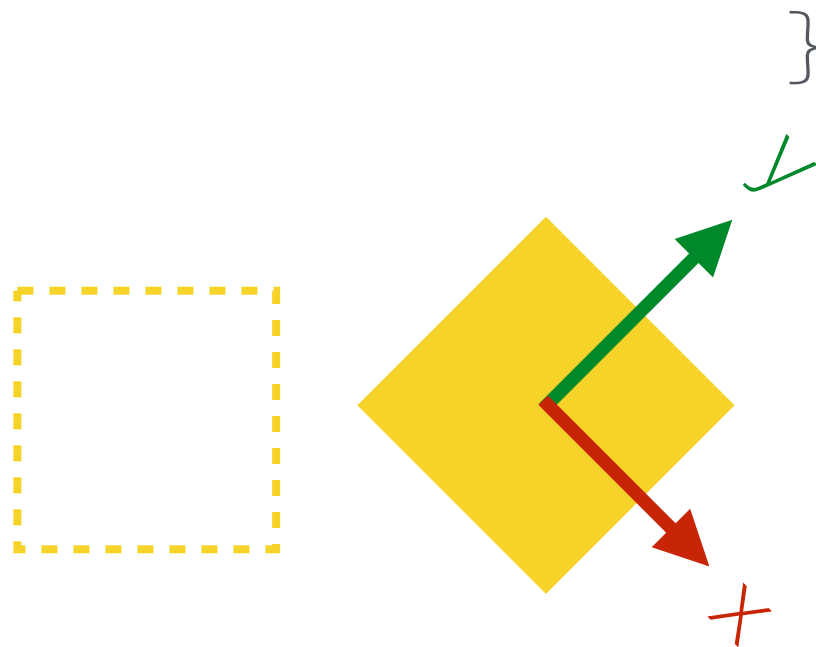
Gotchas

```
#box {  
  transform:  
    translateX(20px)  
    rotate(45deg);  
}
```



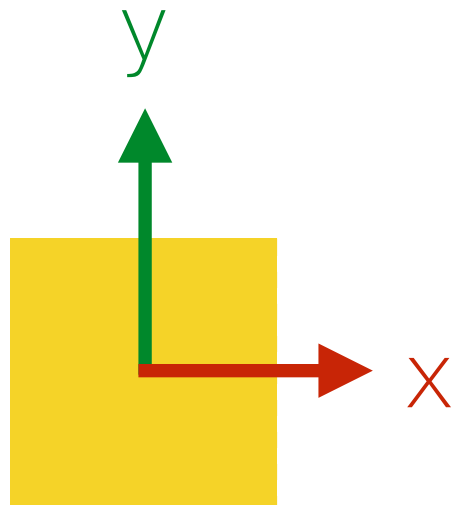
Gotchas

```
#box {  
  transform:  
    translateX(20px)  
    rotate(45deg);  
}
```



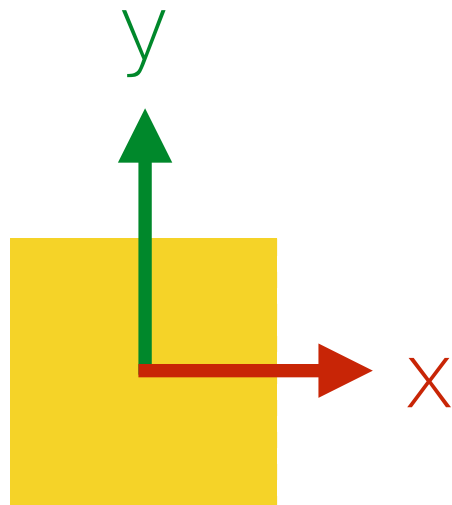
Gotchas

```
#box {  
  transform:  
    translateX(20px)  
    rotate(45deg);  
}
```



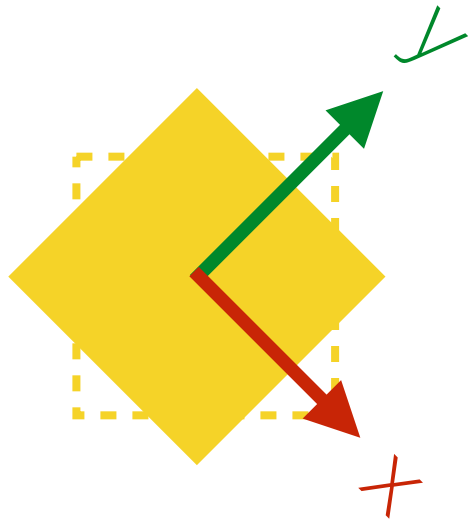
Gotchas

```
#box {  
  transform:  
    rotate(45deg)  
    translateX(20px) ;  
}
```



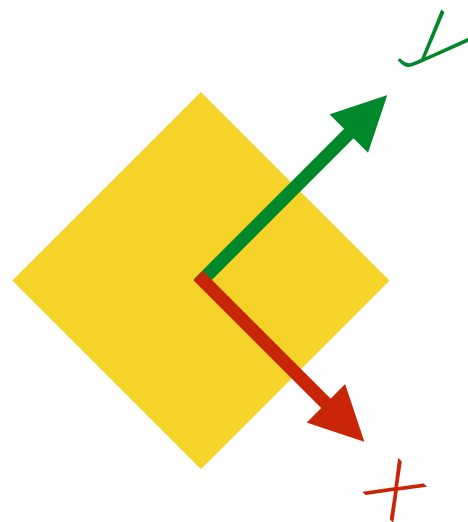
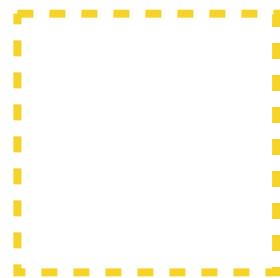
Gotchas

```
#box {  
  transform:  
    rotate(45deg)  
    translateX(20px);  
}
```



Gotchas

```
#box {  
  transform:  
    rotate(45deg)  
    translateX(20px);  
}
```



Gotchas

- Order of operation matters

Gotchas

- Order of operation matters
- No “preserve-3d” on IE

Gotchas

```
button {  
  transform: rotateX(30deg);  
  transform-style: preserve-3d;  
}  
button:after {  
  transform: rotateX(-90deg);  
}  
.button_container:hover button:after {  
  transform: rotateX(-90deg) scaleY(2);  
}  
  
.button_container:active button:after {  
  transform: rotateX(-90deg) scaleY(0);  
}
```

Gotchas

```
button {  
  transform: rotateX(30deg);  
}
```

```
button:after {  
  transform: perspective(800px) rotateX(-60deg);  
}
```

```
.button_container:hover button:after {  
  transform: perspective(800px) rotateX(-60deg)  
           scaleY(2);  
}
```

```
.button_container:active button:after {  
  transform: perspective(800px) rotateX(-60deg)  
           scaleY(0);  
}
```

Gotchas

- Order of operation matters
- No “preserve-3d” on IE

Gotchas

- Order of operation matters
- No “preserve-3d” on IE
- They’re just rectangles

What
Support
Benefits
Code!
Gotchas

github.com/jingidy/practical-css-transforms-2014

CSS Transforms Cheat Sheet

Properties

transform

translate(1px, 5px), translateX(1px), translateY(5px)
scale(1.2, 1.5), scaleX(1.2), scaleY(1.5)
rotate(5deg) **positive = clockwise**
rotate(5deg, 10px, 5px) **move x, y, then rotate**
skew(10deg, 5deg), skewX(10deg), skewY(5deg)
matrix(1, 0, 0, .5, ...) **3x3 = 9 numbers**

translate3d(1px, 5px, 2px)
translateZ(5px) **positive = closer**
scale3d(1.2, .5, 2), scaleZ(2)
rotate3d(1, .5, .2, 10deg) **first 3 numbers are multipliers**
rotateX(10deg), rotateY(5deg), rotateZ(2deg)
perspective(800px) **same as property, but not inherited**
matrix3d(1, 0, .3, ...) **4x4 = 16 numbers**

e.g. `transform: translateX(-5px) rotate(10deg);`

perspective-origin, transform-origin

left, center (**default**), right, top, bottom, %, px

e.g. `transform-origin: left top 10px;` **3rd value is Z origin**

transform-style

flat (**default**), preserve-3d

perspective

e.g. `perspective: 800px;` **smaller = closer = more distorted**

backface-visibility

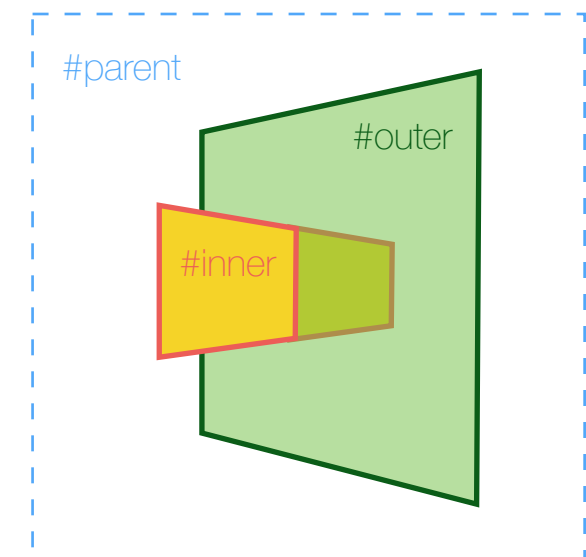
visible (**default**), hidden

Browser Support

	IE	Firefox	Chrome	Safari	Opera			
2D	>9							
3D	>10					IE 10 & 11 don't support preserve-3d		
	Mobile Safari	Opera Mini	Android	Blackberry	Opera Mobile	Chrome for Android	Firefox for Android	IE Mobile
2D								
3D			>22					

How To 3D

```
#parent {  
  perspective: 800px;  
}  
  
#outer {  
  transform: rotateY(-40deg);  
  transform-style: preserve-3d;  
}  
  
#inner {  
  transform: rotateY(90deg);  
}
```



Tips

- order of operation matters
- IE doesn't support preserve-3d

Prefixes

-webkit Safari, Chrome
-moz Firefox
-o Opera
-ms IE

e.g. `-webkit-transform: scale(1.1)`