LAB

LEARNING OBJECTIVES

- Use the position property to position elements on the page
- Utilize transitions and transforms to add basic animations on hover

ANIMATION

ADVANCED CSS POSITIONING

ACTIVITY — POSITIONING

EXERCIS
E

KEY OBJECTIVE

- Differentiate between various positioning techniques.

TYPE OF EXERCISE

- Groups

TIMING

4 min

1. Complete steps 1 - 4B in positioning_intro
2. Bonus: If you finish early, look up "z-index CSS". What does this property do? Write a summary in Slack.

STATIC POSITIONING

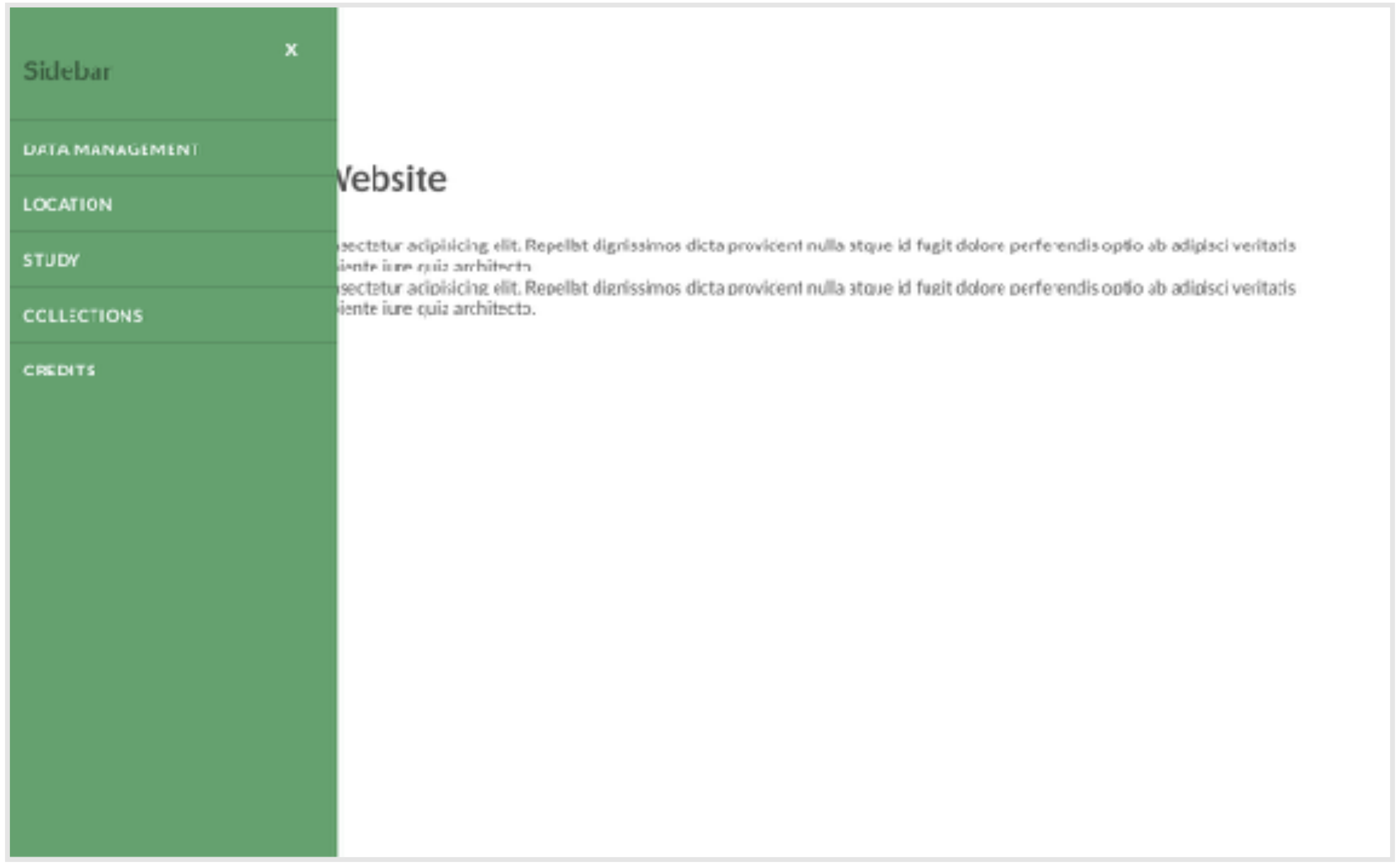
- By default, elements on a page are similar to these wooden blocks.
- They will stack one on top of the other in the same order that they are placed in an HTML file. This is referred to as the "normal flow" of a document.

STATIC POSITIONING

- We can use the position property in our CSS to take elements out of the normal flow of the document and specify where they should appear.

```css
.my-class {
  position: fixed;
}
```

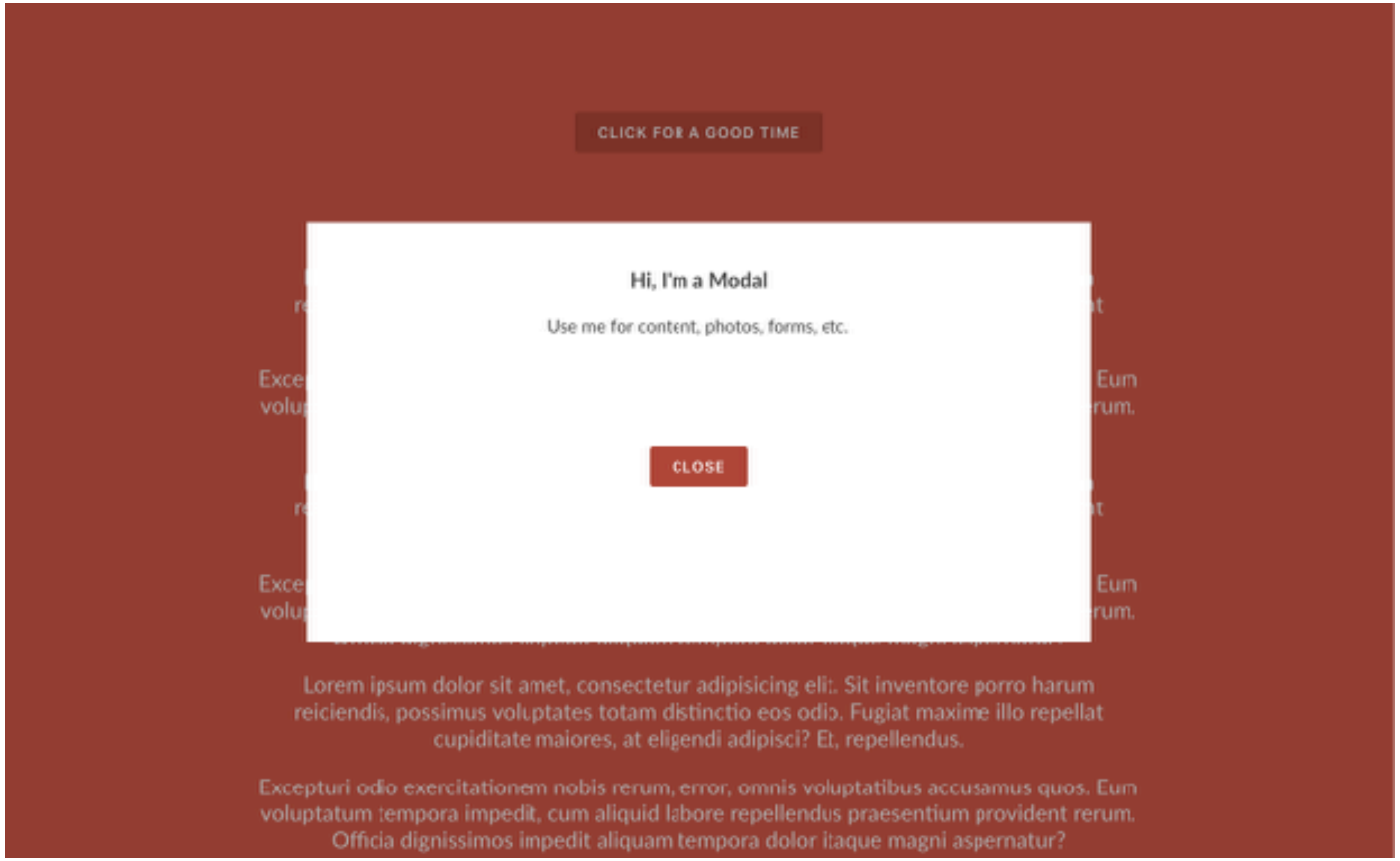
CSS POSITIONING — SIDEBAR



Vebsite

sectetur adipisicing elit. Repellat dignissimos dicta provident nulla atque id fugit dolore perferendis optio ab adipisci veritatis piente iure quia architecto.
sectetur adipisicing elit. Repellat dignissimos dicta provident nulla atque id fugit dolore perferendis optio ab adipisci veritatis piente iure quia architecto.

CSS POSITIONING — MODAL WINDOW



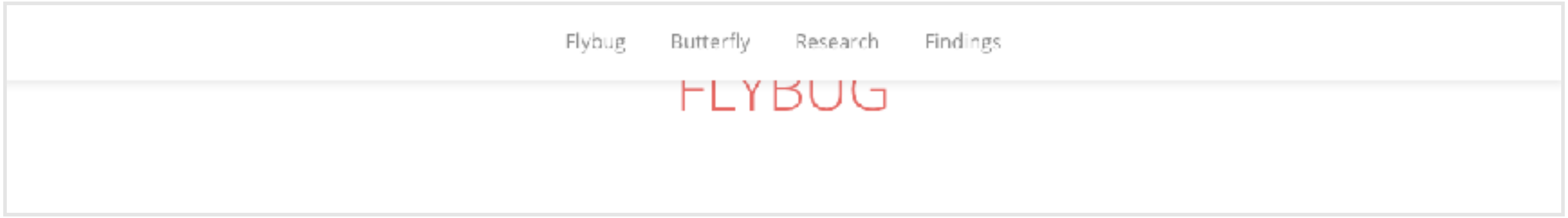
Hi, I'm a Modal

Use me for content, photos, forms, etc.

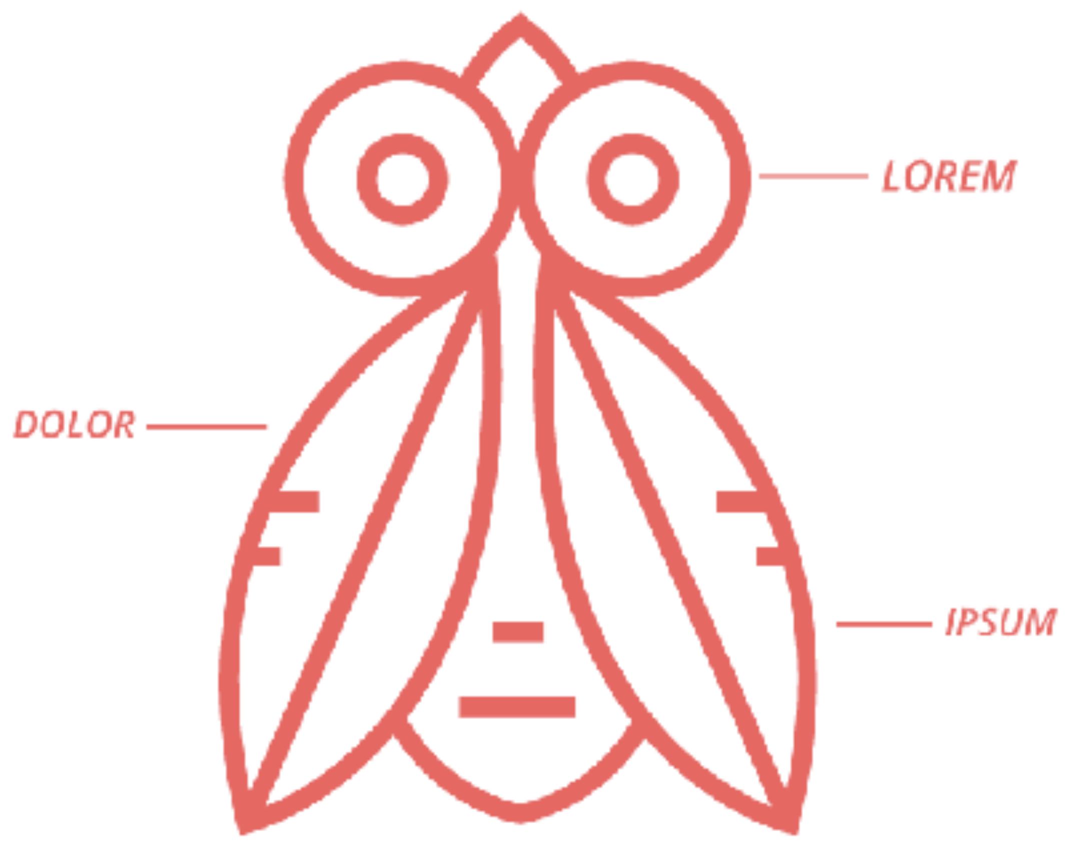
Lorem ipsum dolor sit amet, consectetur adipisicing elit. Sit inventore porro harum reiciendis, possimus voluptates totam distinctio eos odio. Fugiat maxime illo repellat cupiditate maiores, at eligendi adipisci? Et, repellendus.

Excepturi odio exercitationem nobis rerum, error, omnis voluptatibus accusamus quos. Eum voluptatum tempora impedit, cum aliquid labore repellendus praesentium provident rerum. Officia dignissimos impedit aliquam tempora dolor itaque magni aspernatur?

CSS POSITIONING — STICKY NAV



CSS POSITIONING — LABELS FOR IMAGE



ANIMATION

STATIC POSITIONING

CSS POSITIONING

STATIC

RELATIV
E

FIXED

ABSOLU
TE

STATIC POSITIONING

- The default position on each element is static.
- Elements with a position of static will appear in order and stack on top of each other, like we would expect.

```
yourSelectorHere {
  position: static;
}
```

ANIMATION

FIXED POSITIONING

CSS POSITIONING

STATIC

RELATIV
E

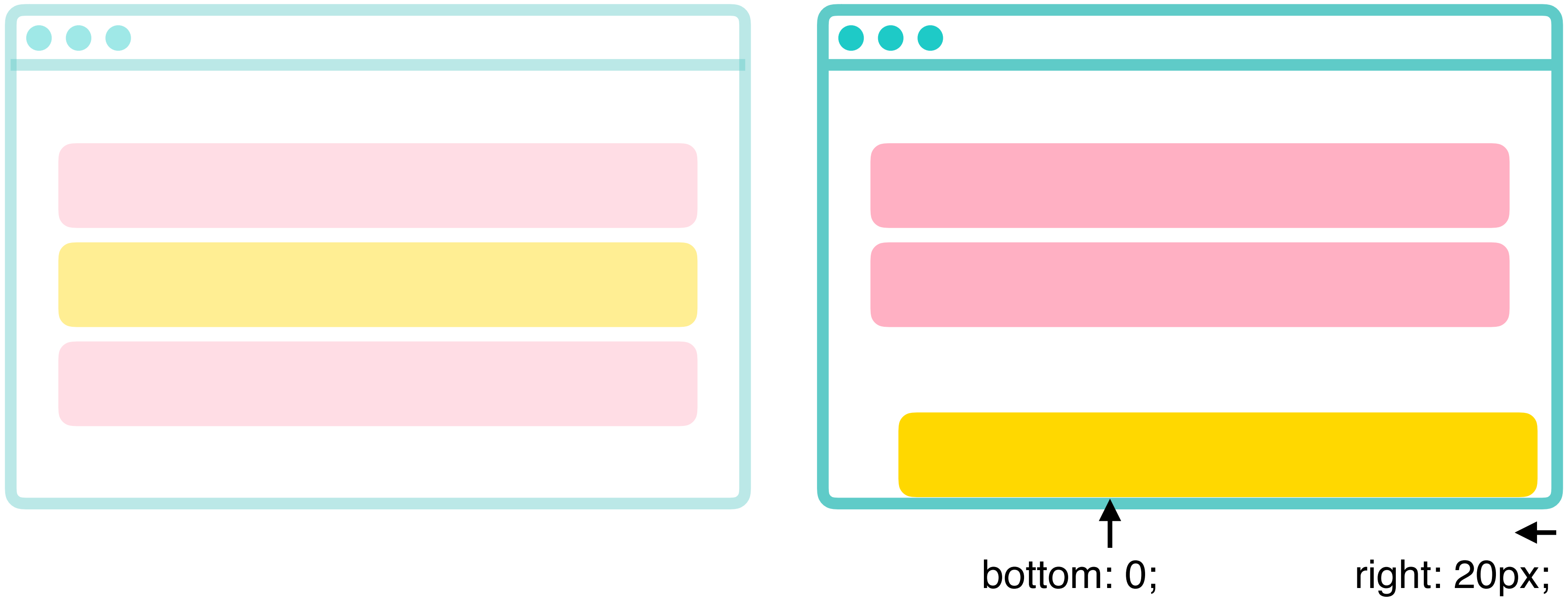
FIXED

ABSOLU
TE

FIXED POSITIONING

- Positioned in relation to the browser window
- When the user scrolls, it stays in the same place.
- Use **right, top, left** *and* **bottom** *properties to specify where the element should go in relation to the browser window.*

```
yourSelectorHere {
  position: fixed;
  bottom: 0;
  right: 20px;
}
```



ANIMATION

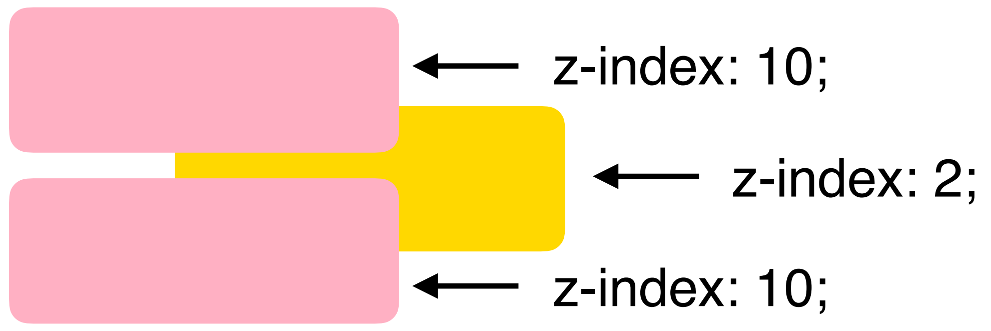
Z-INDEX

OVERLAPPING ELEMENTS — Z-INDEX

- With **relative**, **absolute**, and **fixed** positioning, elements can overlap.
- We can use **z-index** to control which elements are layered on top of each other.
- This property takes a number — the higher the number the closer that element is to the front.

```
.yellow {
  z-index: 2;
}

.pink {
  z-index: 10;
}
```



Think of this like 'bring to front' and 'send to back' in programs like Adobe Illustrator.

ACTIVITY — FIXED NAV

EXERCIS
E

KEY OBJECTIVE

- Practice using CSS positioning

LOCATION

- Starter Code > creepy_crawlers

TIMING

8 min

1. Follow step 1 in main.css

RELATIVE POSITIONING

CSS POSITIONING

STATIC

**RELATIV
E**

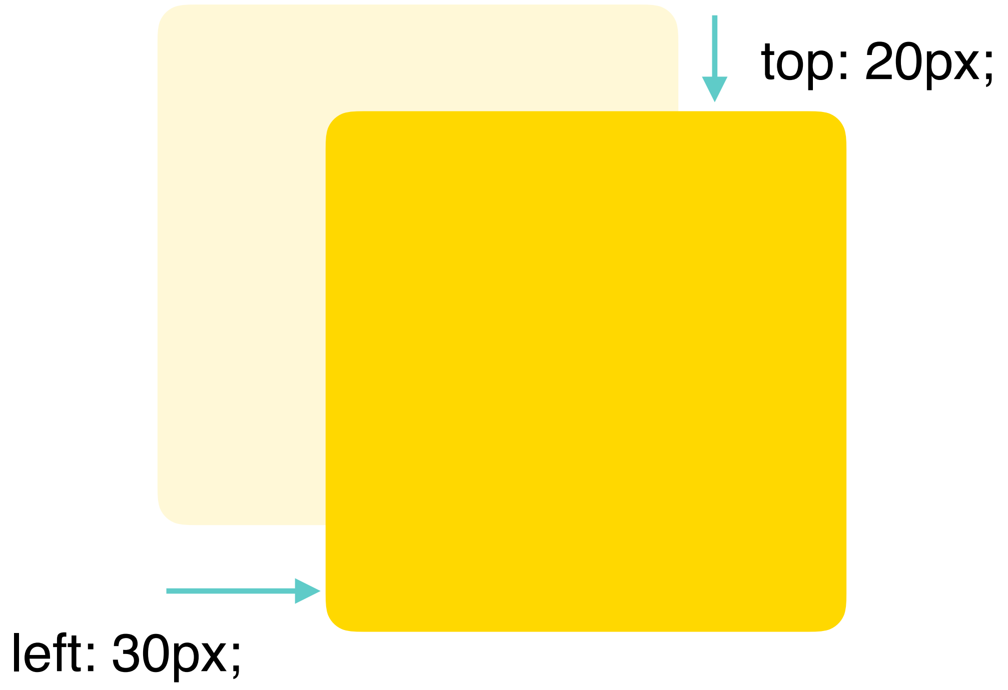
FIXED

ABSOLU
TE

RELATIVE POSITIONING

- Moves an element relative to where it would have been in normal flow.
- For example: left: 20px adds 20px to an element's left position

```
yourSelectorHere {
  position: relative;
  top: 20px;
  left: 30px;
}
```



ABSOLUTE

CSS POSITIONING

STATIC

RELATIV
E

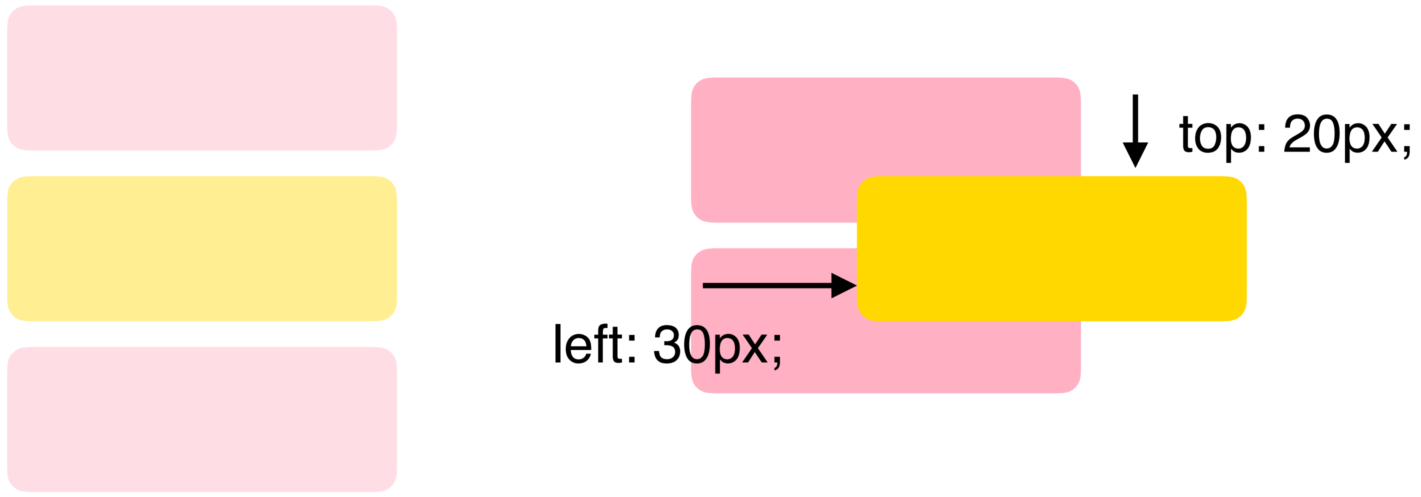
FIXED

**ABSOLU
TE**

ABSOLUTE POSITIONING

- Element is taken out of the normal flow of the document.
- No longer affects the position of other elements on the page (they act like it's not there).
- You can add the *right, top, left and bottom properties to specify where the element should appear*

```
yourSelectorHere {
  position: absolute;
  top: 20px;
  left: 30px;
}
```

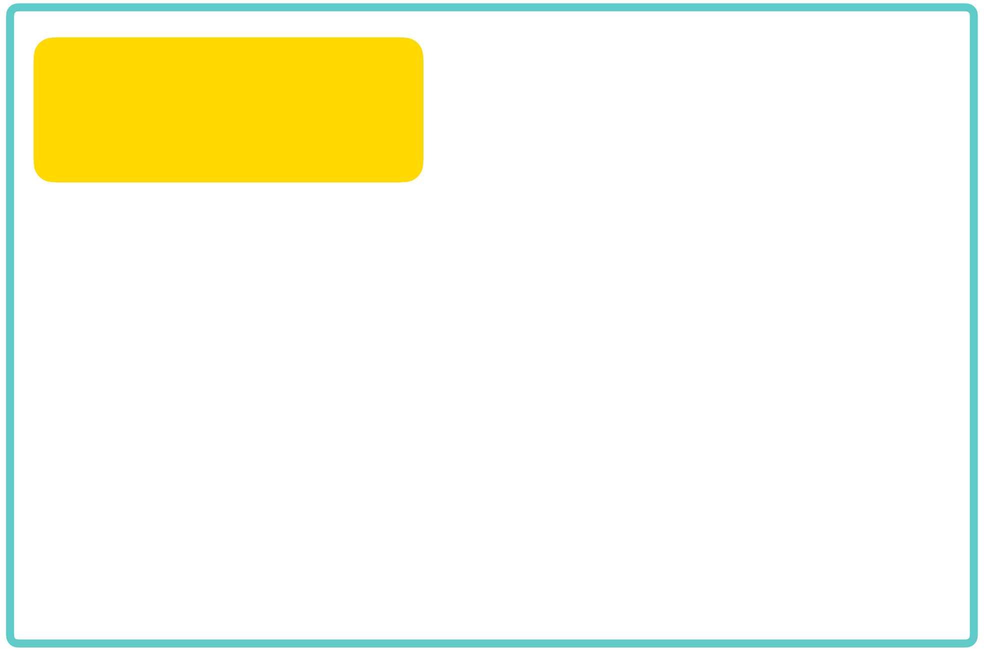


POSITIONING THINGS ABSOLUTELY

Parent we want child to be positioned relative to

```
<section>
  <div class="info"></div>
</section>
```

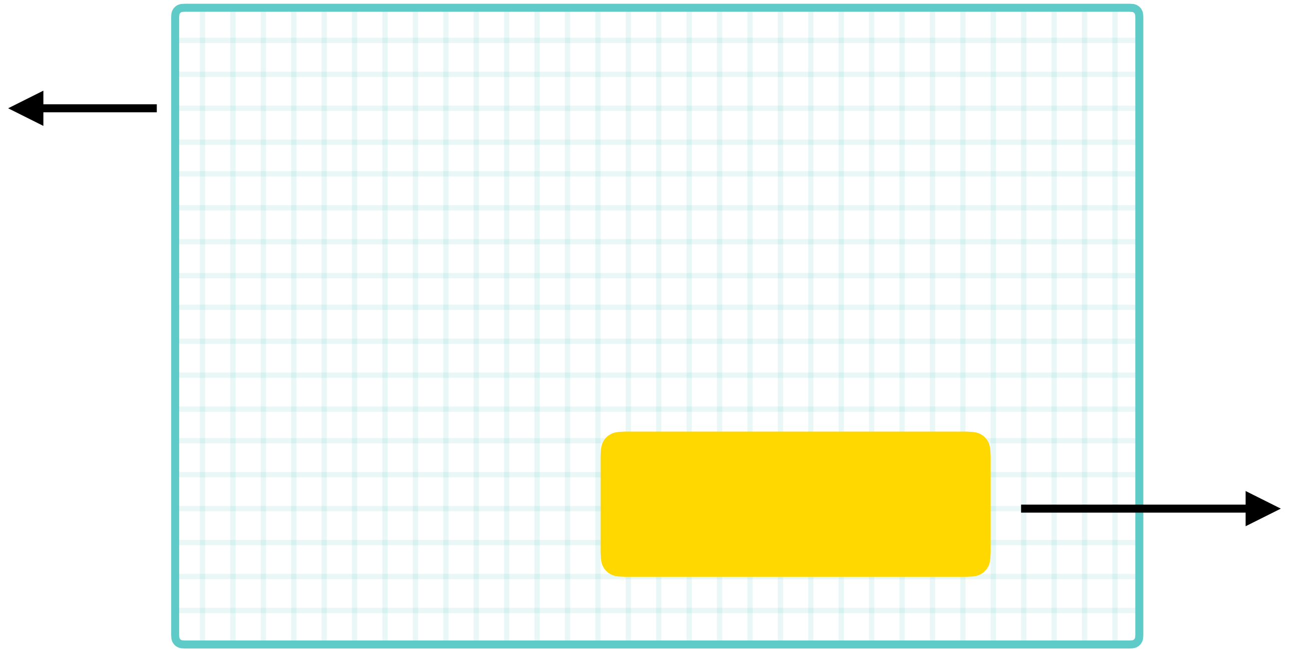
Child we want to position



POSITIONING THINGS ABSOLUTELY

Parent:

position: relative;



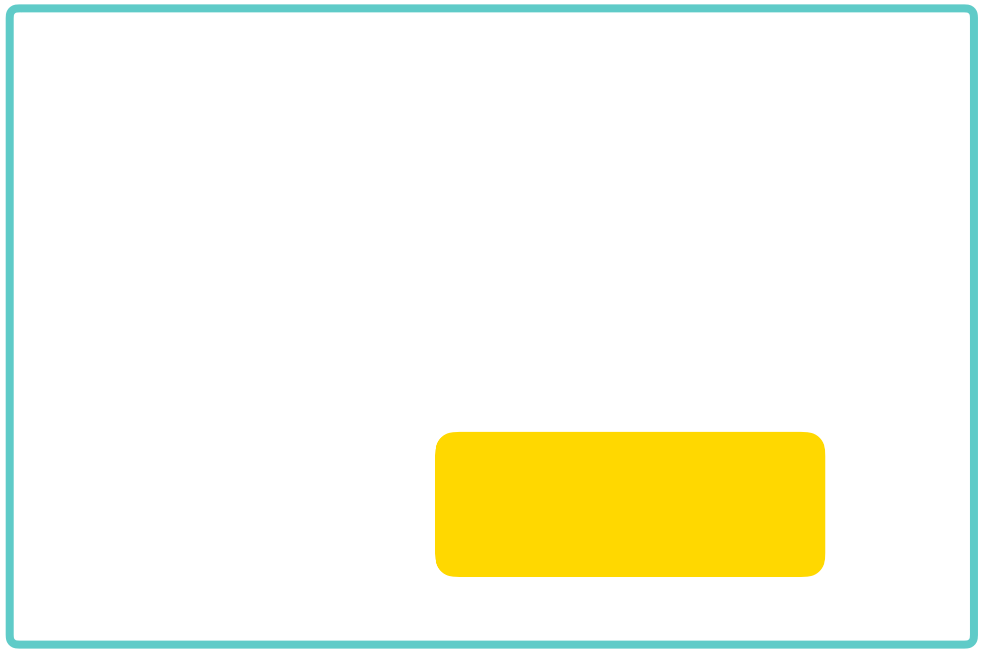
Child:

- position: absolute;
- top, bottom, left, or right

POSITIONING THINGS ABSOLUTELY

```
<section>
  <div class="info"></div>
</section>
```

```
section {
  position: relative;
}
.info {
  position: absolute;
  bottom: 20px;
  right: 50px;
}
```



ACTIVITY — ABSOLUTE POSITIONING

EXERCIS
E

KEY OBJECTIVE

- Practice using CSS positioning

LOCATION

- Starter Code > creepy_crawlers

TIMING

8 min 1. Follow step 2 in main.css

WANT TO LEARN MORE?

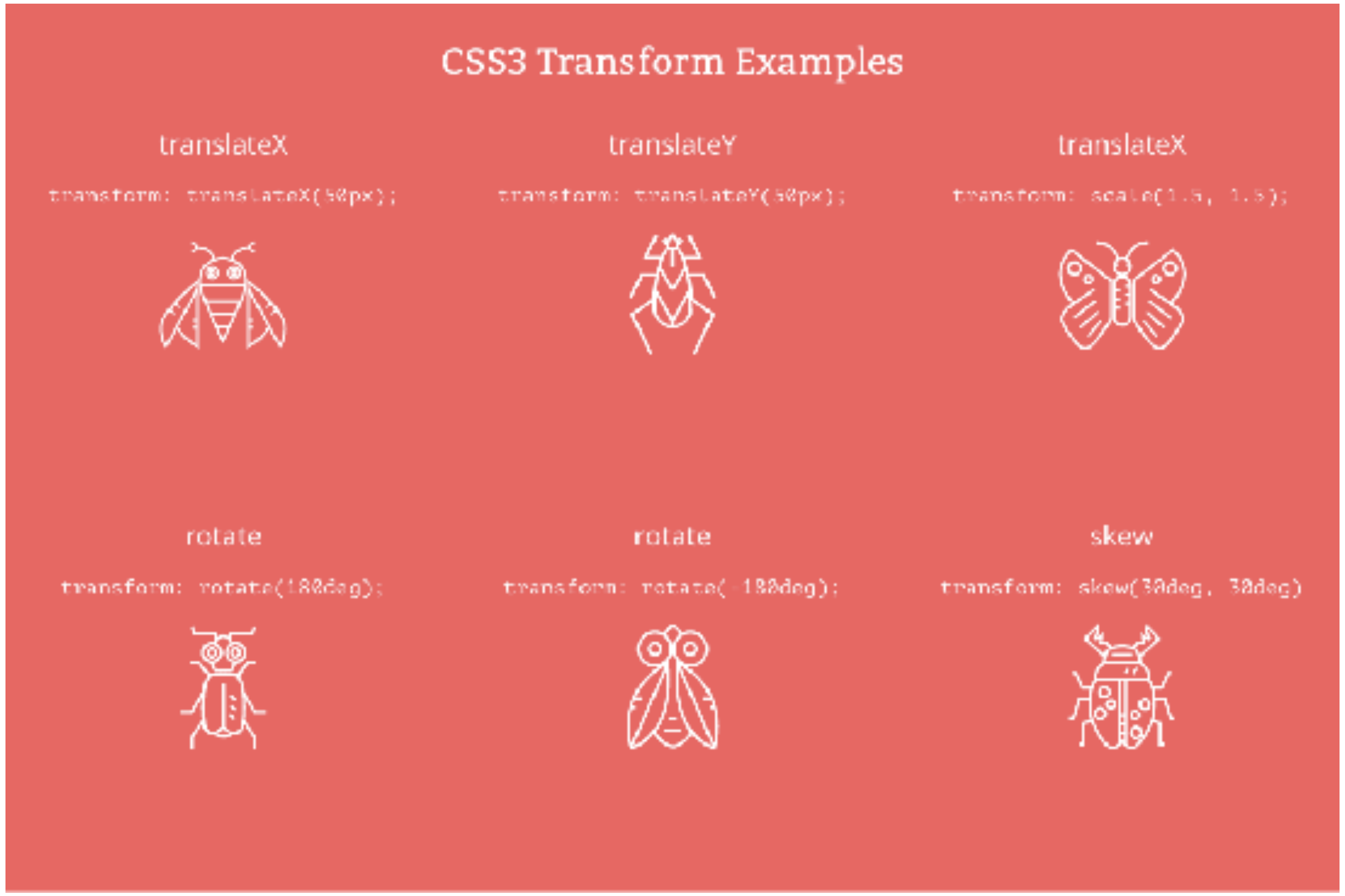
Resources for more info/examples:

- A List Apart: CSS Positioning 101

ANIMATION

TRANSITIONS

LET'S TAKE A CLOSER LOOK — TRANSFORM


CSS3 Transform Examples

translateX

```
transform: translateX(50px);
```

translateY

```
transform: translateY(50px);
```

translateX

```
transform: scale(1.5, 1.5);
```

rotate

```
transform: rotate(180deg);
```

rotate

```
transform: rotate(-180deg);
```

skew

```
transform: skew(30deg, 30deg)
```

Syntax: [W3 Schools]

TRANSITIONS

- Provide a way to control animation speed when changing properties
- Instead of having property changes take effect immediately, you can have them take place over a period of time.

```
yourSelectorHere {
    transition: what-to-transition animation-duration timing-function delay;
}
```

EXAMPLE:

```
transition: all 350ms ease-in-out;
```

TRANSITIONS - TRANSITION-PROPERTY

- Can specify a specific property to transition or "all" to transition all properties
- Default: all

```
div {
    transition: opacity 0.5s;
}
```

```
div {
    transition: all 0.5s;
}
```

TRANSITIONS - TRANSITION-DURATION

- A time value, defined in seconds or milliseconds

```
div {
    transition: height 0.5s;
}
```

```
div {
    transition: height 300ms;
}
```

TRANSITIONS

- Describes how a transition will proceed over its duration, allowing a transition to change speed during its course.
- Timing functions: **ease, linear, ease-in, ease-out, ease-in-out**

```
div {
    transition: opacity 0.5s ease;
}
```

TRANSITIONS

- Length of time before the transition starts

```
div {
    transition: background-color 0.5s ease 2s;
}
```

MORE FUN WITH TRANSITIONS — CODROPS

Fun CSS button styles: Creative buttons

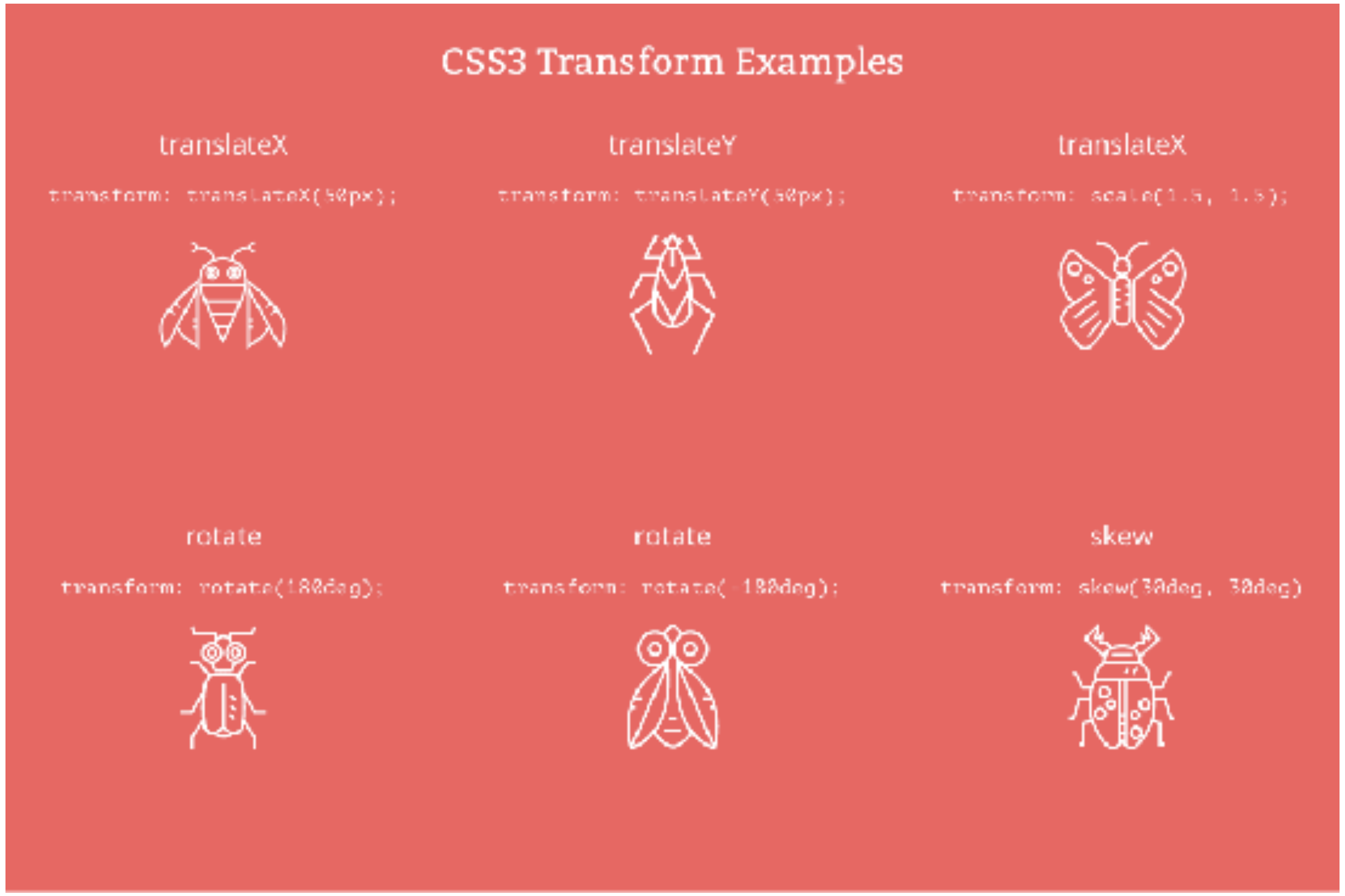
Icon hover effects: Icon Hover Effects

Modal dialogue effects (advanced): Dialogue Effects

ANIMATION

TRANSFORMATIONS

LET'S TAKE A CLOSER LOOK — TRANSFORM



CSS3 Transform Examples

translateX

```
transform: translateX(50px);
```

translateY

```
transform: translateY(50px);
```

translateX

```
transform: scale(1.5, 1.5);
```

rotate

```
transform: rotate(180deg);
```

rotate

```
transform: rotate(-180deg);
```

skew

```
transform: skew(30deg, 30deg)
```

Syntax: [W3 Schools]

ACTIVITY — TRANSFORM ON TIMER

EXERCIS
E

KEY OBJECTIVE

- Practice using CSS transitions

TYPE OF EXERCISE

- Individual/Partner Lab

TIMING

10 min

1. Follow the instructions in starter code > transform_bug > style.css
2. You'll want to use CHROME to test this!

FEWD

KEYFRAME ANIMATIONS

KEYFRAME ANIMATIONS — STEP 1

- Define your animation in your CSS file.

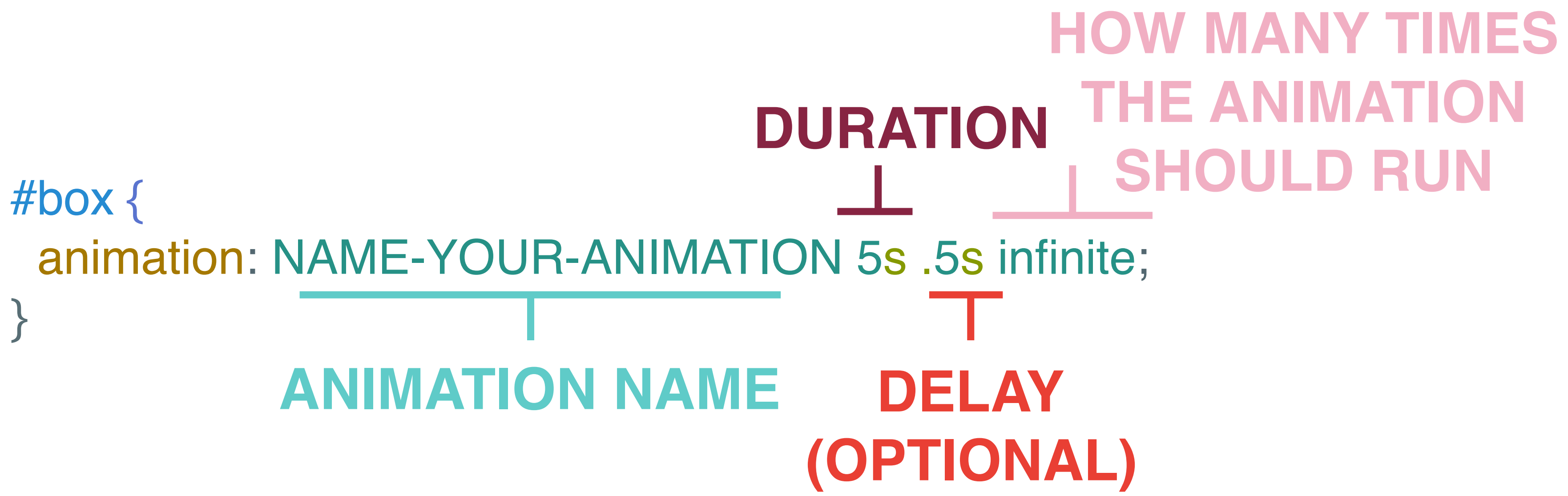
GIVE YOUR ANIMATION A NAME

```
@keyframes NAME-YOUR-ANIMATION {
  0%   {
     opacity: 0;
  }
  100% {
     opacity: 1;
  }
}
```

SPECIFY ANY PROPERTIES YOU'D LIKE TO ANIMATE

KEYFRAME ANIMATIONS — STEP 2

- Now specify what element the animation should be applied to, how long each animation cycle should last, and how many times the animation should run — 1, 2, infinite, etc.



```css
#box {
  animation: NAME-YOUR-ANIMATION 5s .5s infinite;
}
```

KEYFRAME ANIMATIONS — STEP 2

- Make an animation run when the page loads:

```
#box {
  animation: NAME-YOUR-ANIMATION 5s infinite;
}
```

- Make an animation run when the user hovers on an element:

```
#box:hover {
  animation: NAME-YOUR-ANIMATION 5s infinite;
}
```

- Store the animation in a class, and then you can have the animation run when the class gets added with JavaScript:

```
.active {
  animation: NAME-YOUR-ANIMATION 5s infinite;
}
```

ACTIVITY

EXERCIS
E

LOCATION

- starter code > animated_loaders

KEY OBJECTIVE

- Practice using keyframe animations

TIMING

10 min

1. Test out the animated loaders
2. Write CSS to recreate the animated loaders.

ACTIVITY

EXERCISE

LOCATION

- starter code > css-ghost

KEY OBJECTIVE

- Practice using keyframe animations

TIMING

10 min

1. Test out the css-ghost site
2. Write CSS to recreate this functionality.

ACTIVITY

EXERCIS
E

LOCATION

- starter code > image_overlay

KEY OBJECTIVE

- Practice working through common interactions

TIMING

20 min

1. Demo with your groups the live site and discuss which position things will have on the page — absolute, relative, or fixed?
2. Take a minute to look up "how to hide content that is flowing out of parent CSS"
3. Work through the steps in your CSS.

LAB

LEARNING OBJECTIVES

- Use the position property to position elements on the page
- Utilize transitions and transforms to add basic animations on hover

EXIT TICKETS