LAB

LEARNING OBJECTIVES

- Differentiate between inline, block, and inline-block elements
- 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



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

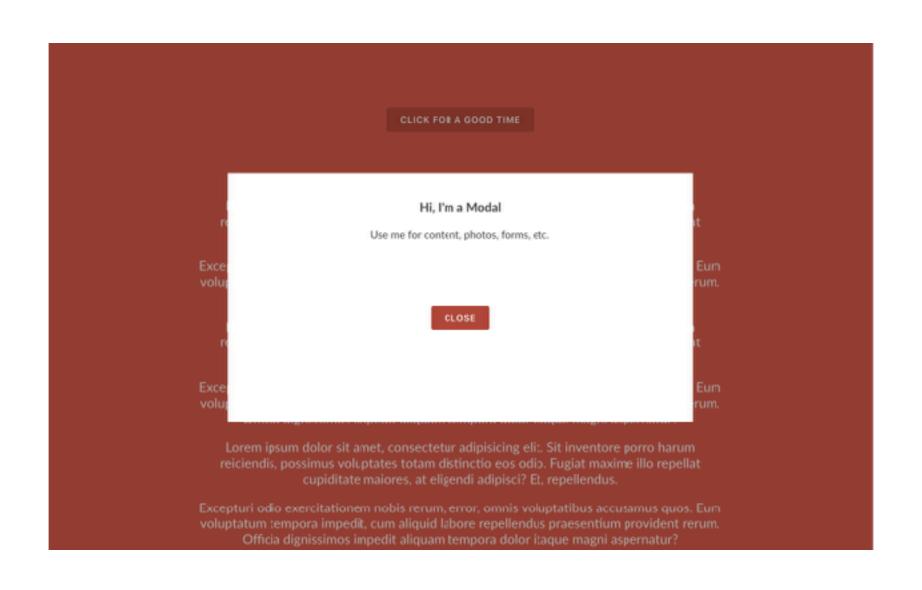
• We can use the position property in our CSS to take elements out of the normal flow of the document and specify where they should appear.

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

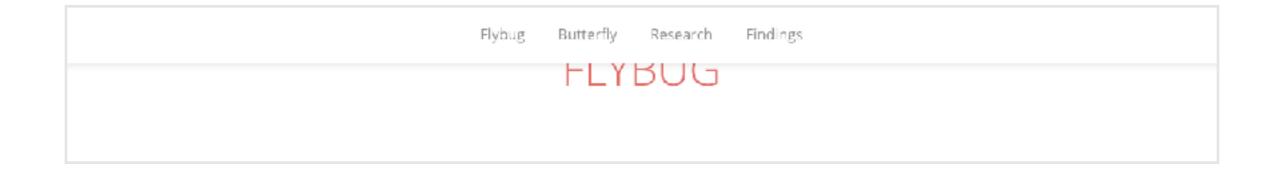
CSS POSITIONING — SIDEBAR



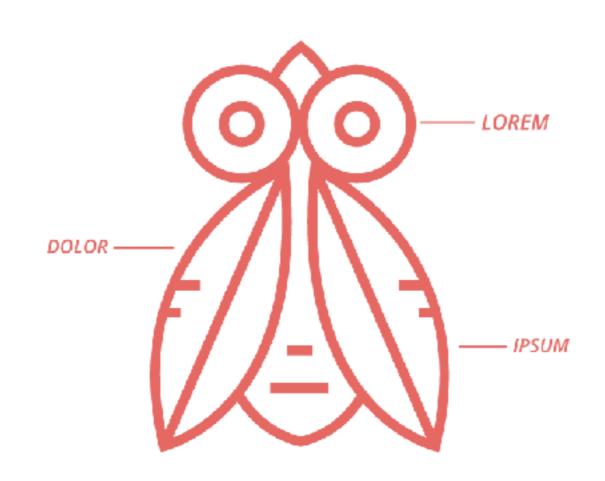
CSS POSITIONING — MODAL WINDOW



CSS POSITIONING — STICKY NAV



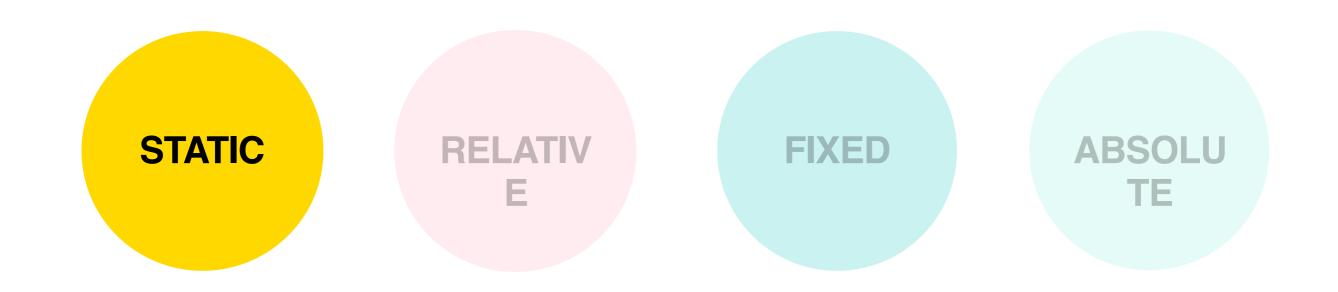
CSS POSITIONING — LABELS FOR IMAGE



ANIMATION

STATIC POSITIONING

CSS POSITIONING



STATIC POSITIONING

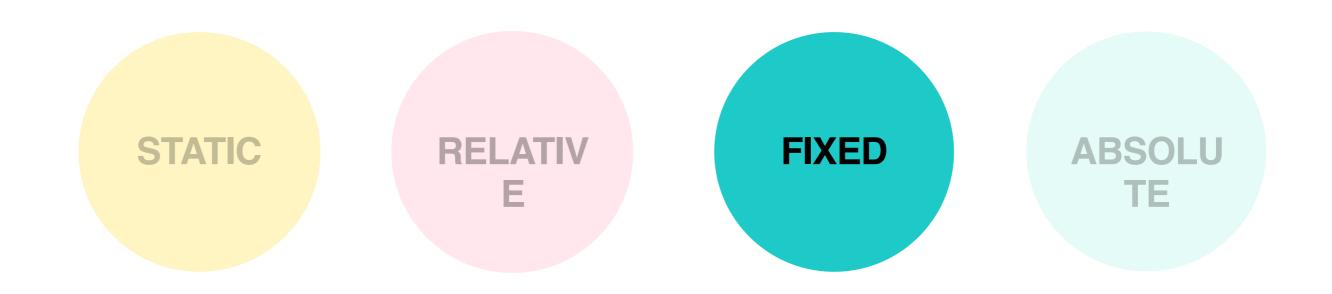
- → The default position on each element is static.
- Elements with a position of static will appear in order and stack on top of each other, like we would expect.

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

ANIMATION

FIXED POSITIONING

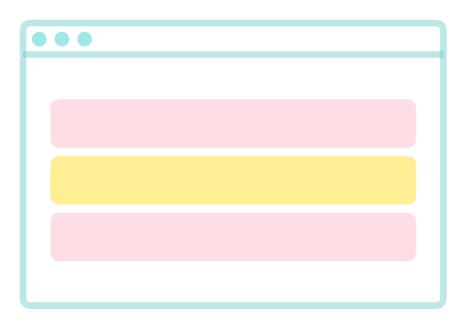
CSS POSITIONING

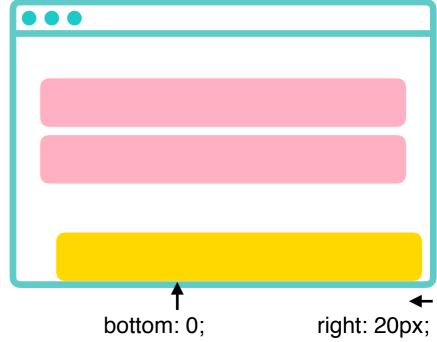


FIXED POSITIONING

- 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.

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

ACTIVITY — FIXED NAV



KEY OBJECTIVE

Practice using CSS positioning

LOCATION

Starter Code > creepy_crawlers

TIMING

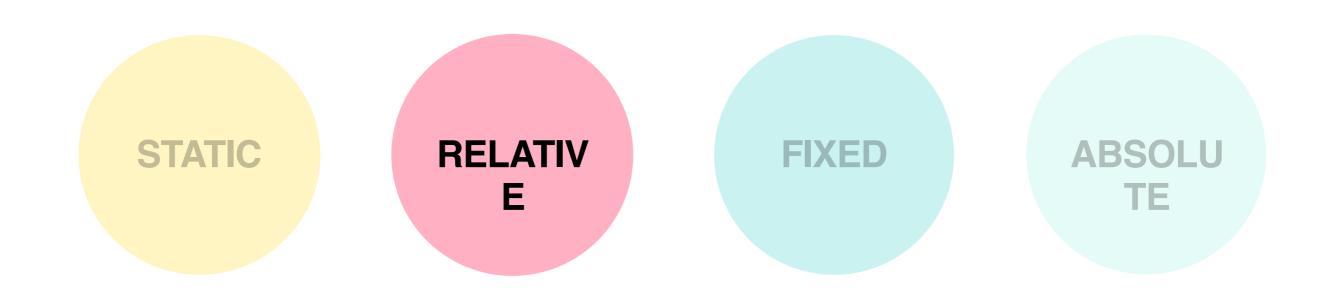
8 min

1. Follow step 1 in main.css

ANIMATION

RELATIVE POSITIONING

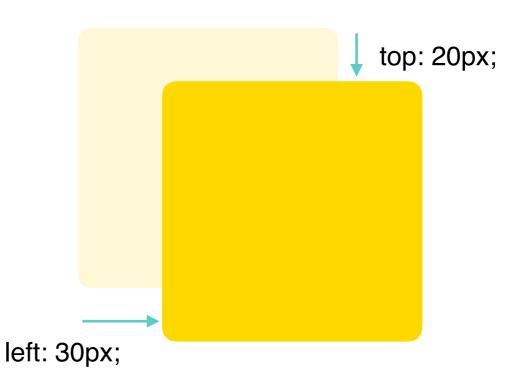
CSS POSITIONING



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

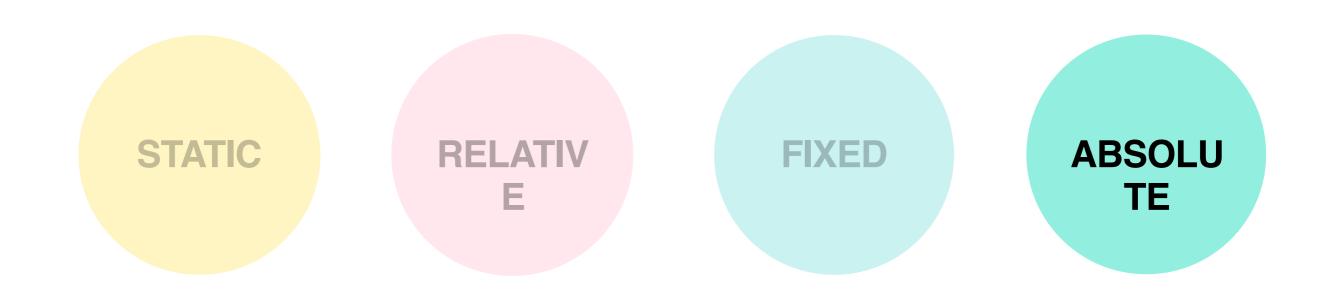
```
yourSelectorHere {
  position: relative;
  top: 20px;
  left: 30px;
}
```



ANIMATION

ABSOLUTE

CSS POSITIONING

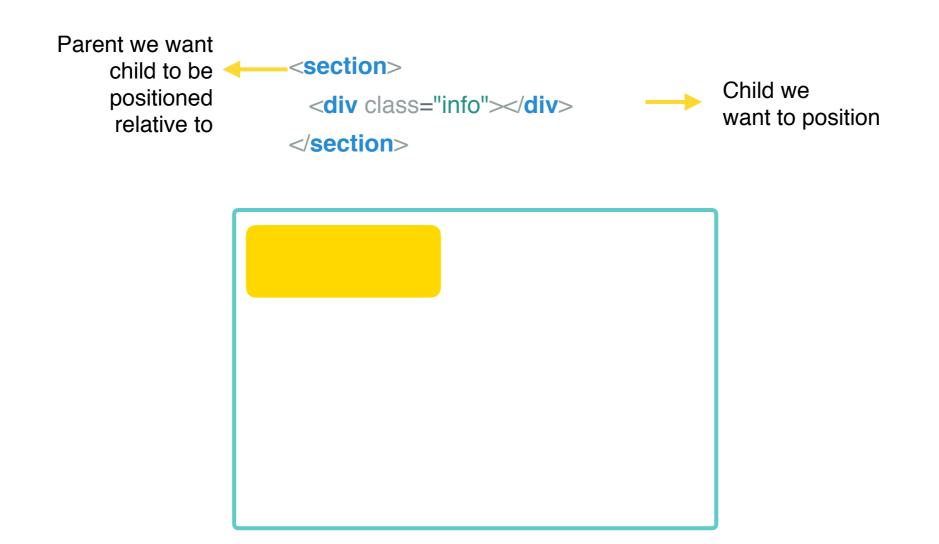


ABSOLUTE POSITIONING

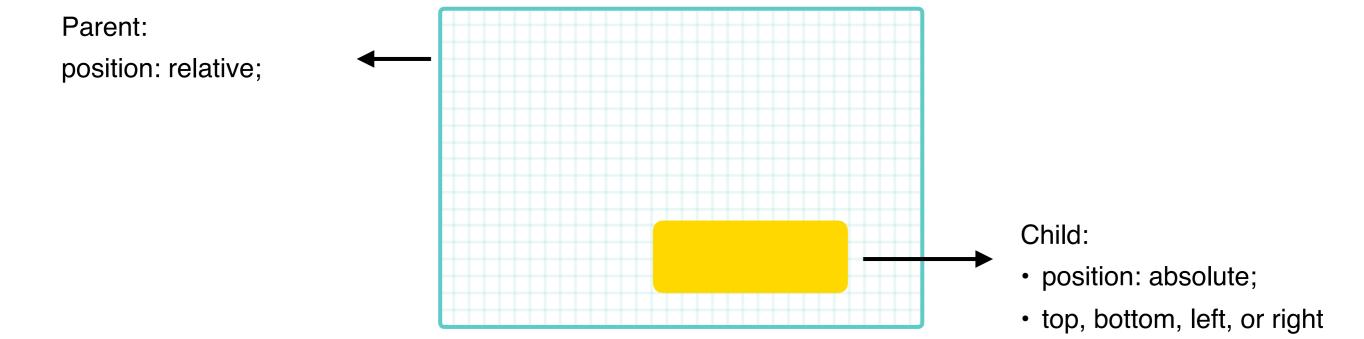
- Element is taken out of the normal flow of the document.
- No longer affects the position of other elements on the page (they act like it's not there).
- You can add the right, top, left and bottom properties to specify where the element should appear

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

POSITIONING THINGS ABSOLUTELY

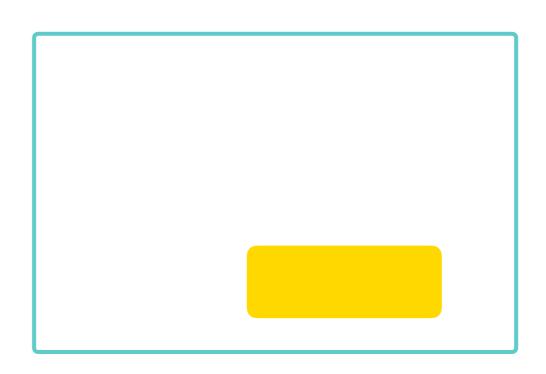


POSITIONING THINGS ABSOLUTELY



POSITIONING THINGS ABSOLUTELY

```
<section>
 <div class="info"></div>
</section>
section {
 position: relative;
.info {
 position: absolute;
 bottom: 20px;
 right: 50px;
```



ACTIVITY — ABSOLUTE POSITIONING



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?

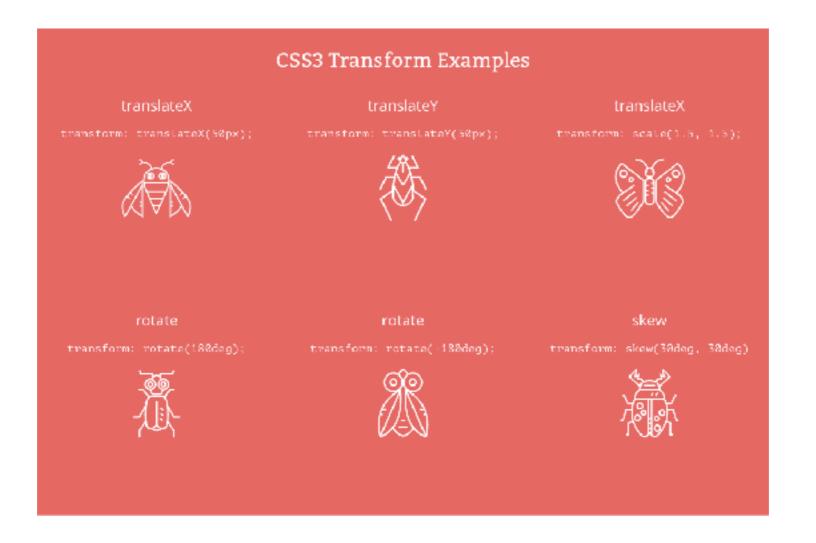
Resources for more info/examples:

→ A List Apart: CSS Positioning 101

ANIMATION

TRANSITIONS

LET'S TAKE A CLOSER LOOK — TRANSFORM



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

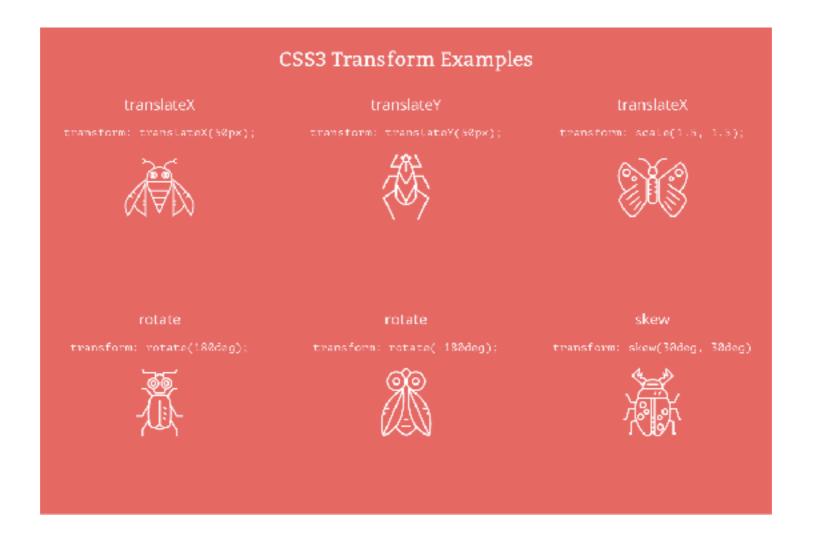
Icon hover effects: Icon Hover Effects

Modal dialogue effects (advanced): <u>Dialogue Effects</u>

ANIMATION

TRANSFORMATIONS

LET'S TAKE A CLOSER LOOK — TRANSFORM



Syntax: W3 Schools

ACTIVITY — TRANSFORM ON TIMER



KEY OBJECTIVE

Practice using CSS transitions

TYPE OF EXERCISE

Individual/Partner Lab

TIMING

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

KEYFRAME ANIMATIONS

KEYFRAME ANIMATIONS — STEP 1

Define your animation in your CSS file.

GIVE YOUR ANIMATION A NAME

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.

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

ANIMATION NAME DELAY
(OPTIONAL)
```

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



LOCATION

starter code > animated_loaders

KEY OBJECTIVE

Practice using keyframe animations

TIMING

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

ACTIVITY



LOCATION

starter code > css-ghost

KEY OBJECTIVE

Practice using keyframe animations

TIMING

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

ACTIVITY



LOCATION

starter code > image_overlay

KEY OBJECTIVE

Practice working through common interactions

TIMING

- 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

- Differentiate between inline, block, and inline-block elements
- Use the position property to position elements on the page
- Utilize transitions and transforms to add basic animations on hover

HTML BASICS

EXIT TICKETS