

Core Lab Interaction

Homework 4

Recode and animate a Josef Albers using CSS animations

1. Pick a source image, Google image search “Josef Albers Interaction of Color”
2. Use relative sizes (% , vh, vmin instead of px) as often as possible

Deploy as 2 versions: full screen, and loading icon

Often, apps will have a loading icon, centered in the middle of the page, quietly animating away to let the user know that the computer isn't frozen

1. Before you begin, wrap a container div around everything and set the CSS. This shouldn't change anything and you can finish your work without it getting in the way:

```
<body>
  <div class="container-1">

    ""
  </div>
</body>
```

```
.container-1 {
  width:100%;
  height:100%;
}
```

2. After you finish, try alternating this container between these 2 classes.

```
.container-2{
  width:20vh; /* 20% screen size */
  height:20vh;
  border-radius: 50%; /* circle border */
  overflow: hidden; /* crop 1/3 */
  position:relative; /* crop 2/3 */
  z-index:1; /* crop 3/3 */
}
```

Notice that your piece will probably re-size all wrong :) It's okay! *(From my experience, it will be too difficult to properly replicate a Josef Albers piece entirely using %)*

3. Duplicate your homework file into two HTML files, one for the full screen, one for the small icon. Adjust the size values in the icon file until it's properly sized again.

Helpful Things

Remember there are 3 parts to an animation: the **keyframes**, the **animation class**, and the **div**

```
@keyframes animationName {  
  100% { transform: rotateZ(360deg); }  
}  
.rotate-class {  
  animation: animationName 10s infinite linear;  
}
```

```
<div class="rotate-class">  
</div>
```

- remember to give your body and html width:100% and height:100%, and any divs that need to have volume also set their width and height
- remember our new flex-box tool for centering: <https://css-tricks.com/centering-css-complete-guide/> (put it in the parent container, the div outside of the div)

```
/* vertical and horizontal */  
  
.flex-center{  
  display: flex;  
  justify-content: center;  
  align-items: center;  
}
```

- Josef Albers pieces often use opacity and divs covering other divs. There is a CSS command for opacity, and use absolute positioning to put a div on top of a div:

```
/* between 0.0 and 1.0 */  
  
.translucent{  
  opacity: 0.5  
}
```

```
.box1{  
  position: absolute;  
  top: 25%;  
  left: 75%;  
}
```

- a complete list of transforms: https://www.w3schools.com/cssref/css3_pr_transform.asp
- by default, the point of rotation (transform-origin) is 50%, 50% (the center): https://www.w3schools.com/cssref/css3_pr_transform-origin.asp
- examples: <https://css-tricks.com/snippets/css/keyframe-animation-syntax/>
- the 4th argument, the “linear” or “step(60)” in the clock example can be replaced with other timing functions (try replacing “linear” with “ease-in-out”): https://www.w3schools.com/cssref/css3_pr_transition-timing-function.asp
- we went over a few of the animation arguments, check out a more complete list here: <https://developer.mozilla.org/en-US/docs/Web/CSS/animation>