Animation and effects with CSS

I've been become more comfortable playing with CSS features that are beyond just making sure that typography works well in all form factors.

Rotate On Hover

One of the things that I like is the subtle and fun rotation that Smashing Magazine does with the author's photo when you first visit an article and how it moves to a fully vertical position when you hover over the image. I've always thought that that's backwards and that it should move to the diagonal position when you hover over the image... so let's try to do it that way.

The default state for the image sets the dimensions and a border (I like borders on images).

```css
img {
  border: 5px solid rebeccapurple;
  height: 150px;
  width: 150px;
}
```

The change happens in the hover state. Here we do two things:

- We tell the browser that we want the rotation to happen from the bottom left coorner
- We use rotate the image -15 degrees (15 degrees counter clockwise)

```css
img:hover {
  transform-origin: bottom left;
  transform: rotate(-15deg);
}
```

The basics work but there's one more thing that I want to do. Transform allows us to chain effects so why not make the image slightly bigger as we rotate it? The

modified code adds a `scale` transform to the existing rotation.

```css
img:hover {
  transform-origin: bottom left;
  transform: scale(1.2) rotate(-15deg);
}
```

You can see the finished product in [this pen](#)

We can also choose to rotate the image, with our without the scaling, and do nothing with it on hover:

```css
img {
  border: 5px solid rebeccapurple;
  height: 150px;
  width: 150px;
  transform-origin: bottom left;
  transform: rotate(-15deg);
}
```

Enlarging text on hover

One thing that I wish we could do is make the links slightly larger on hover so the text of the link is .

Because we don't want reader to get confused between the englarged link text and the surrounding content we made the background black and the text white.

```css
a:hover {
  display: inline-block;
  background: black;
  color: white;
  transform-origin: bottom-left;
  transform: scale(1.25);
}
```

Enlarge a search box

Some of my favorite sites expand the search box when users hover over the search box to a pre-set width.

Based on Searchbox Purecss Hover transition I built the following demo to test such an idea.

The markup

The example uses font awesome for the search icon. It would probably work better if we were to use a SVG icon… but for the demo it'll be fine.

```html
<div class="search-container">
  <div class="search-box">
    <input
      class="s-box"
      type="text"
      name="search"
      placeholder="Enter search parameters">
    <a class="s-btn" href="">
      <i class="fas fa-search"></i>
    </a>
  </div>
</div>
```

In the (S)CSS side, we first defined the container as a flex container, set up as a row and aligned to the end of the document (right side in left-to-right languages).

```css
.search-container {
  display: flex;
  flex-flow: row;
  justify-content: flex-end;
}
```

The next step is to setup the dimensions and look of the `search-box` container.

Using the ampersand to create nested selectors for the button and hover states. These are the styles that we'll animate to when we hover over the element.

```css
.search-box {
  height: 40px;
  border: 2px solid rebeccapurple;
  border-radius:40px;
  padding: 10px;
  &:hover{
    .s-box{
      width: 240px;
      padding: 0 6px;
    }
    .s-btn{
      background:#fff;
    }
  }
}
```

The button (class s-btn) contains the icon we're using to represent the search. It is also where we would attach the click event for processing the form.

The button is also a flex container so we can center the icon both vertically and horizontally inside its parent.

```css
.s-btn {
  color: rebeccapurple;
  float: right;
  width: 40px;
  height: 40px;
  transition: 0.4s;
  display: flex;
  justify-content: center;
  align-items: center;

  i {
    font-size: 20px;
  }
}
```

The final element to style is the input element where we'll type what we want to search for.

```css
.s-box{
  border: none;
  background:none;
  outline:none;
  float: left;
  padding: 0;
  color:rebeccapurple;
  font-size:1rem;
  transition:.5s;
  line-height:40px;
  width: 0px;
  }
}
```

The product takes more work than I thought but the result still looks nice as I expected.