

Lesson 5 Extra Challenge 2

PORTHOLE IMAGES WITH CSS TRANSITIONS

In this lab we'll experiment with hover effects on circles. Since we have the border radius property, we can create circular shapes and they have been appearing more often as design elements in websites.

THE HTML

1. Create a new project called CSSPortholeImages
2. Add html document called index.html with the usual html, head, and body tags.
3. Add the following between the opening and closing <body> tags. The mark-up contains an unordered list with four list items

```
<div class="container">
  <section class="main">
    <ul class="ch-grid">
      <li>
        <div class="ch-item ch-img-1">
          <div class="ch-info">
            <h3>Games</h3>
            <p>
              by Bill Hicks <a href="http://www.iristimes.com">Quickly! go there now</a>
            </p>
          </div>
        </div>
      </li>
      <li>
        <div class="ch-item ch-img-2">
          <div class="ch-info">
            <h3>Illustration</h3>
            <p>
              by Steven Wright <a href="http://www.independent.ie ">Captain Semantics</a>
            </p>
          </div>
        </div>
      </li>
      <li>
        <div class="ch-item ch-img-3">
          <div class="ch-info">
            <h3>Characters</h3>
            <p>
              by Johnny Vegas <a href="http://www.examiner.ie "/">Lovable Mentalist</a>
            </p>
          </div>
        </div>
      </li>
    </ul>
  </section>
</div>
```

```

                                </div>
        </li>
    </li>
        <div class="ch-item ch-img-4">
            <div class="ch-info">
                <h3>Modelling</h3>
                <p>
                    by Benedict Cumberbatch <a href="http://www.dribbble.com">Smartypants Himself</a>
                </p>
            </div>
        </div>
    </li>
</ul>
</section>
</div>

```

Load the HTML in a browser and observe. We'll then move on to add the styling

THE CSS

1. Create a new CSS page called style.css and save it to the css folder within your main folder (same as index.html)
2. Set body background and apply a natural box layout model to all elements

```

* {
    -moz-box-sizing: border-box;
    -webkit-box-sizing: border-box;
    box-sizing: border-box;
}

body {
    font-family: sans-serif;
    background-color: #54403C;
}

```

3. Define a common style for the list and the list items:

```

.ch-grid {
    margin: 20px 0 0 0;
    padding: 0;
    text-align: center;
}

```

```
.ch-grid li {
    width: 220px;
    height: 220px;
    display: inline-block;
    margin: 20px;
}
```

Reload the html and observe changes

4. The CSS is going to reveal the description of each list-item (in this case an image) by scaling it up and we will also animate the inset box shadow of the item itself. So let's position the item and set an inset box shadow and a transition:

```
.ch-item {
    width: 100%;
    height: 100%;
    border-radius: 50%;
    position: relative;
    cursor: default;
    box-shadow: inset 0 0 0 6px rgba(255,255,255,0.8), 0 1px
2px rgba(0,0,0,0.8);
    transition: all 0.4s ease-in-out;
}
```

Refresh browser and observe changes

5. Notice in the HTML we've given two classes to the item (not the list item but its child div): one is ch-item and the other one will be used to define a specific background image. The images have been supplied in the images folder. If you want to supply your own, resize them to 220px height and width to fit the styling dimensions, or change the style dimensions in step 4 (height/width)

```
.ch-img-1 {
    background: url(../images/gaming.jpg);
}

.ch-img-2 {
    background: url(../images/character1.jpg);
}
```

```
.ch-img-3 {
    background-image: url(../images/character.jpg);
}

.ch-img-4 {
    background-image: url(../images/modelling.png);
}
```

Refresh browser and observe changes

6. The description element will be positioned absolutely and we'll give it a semi-transparent background by setting an RGBA value. It's opacity is going to be 0 and we'll scale it down to 0, too:

```
.ch-info {
    position: absolute;
    background: rgba(63,147,147, 0.8);
    width: inherit;
    height: inherit;
    border-radius: 50%;
    opacity: 0;
    transition: all 1.4s ease-in-out;
    transform: scale(0);
}
```

Some comments: the width and height are inherited from the parent div to make the description element the same size as its parent. The border-radius makes it circular. Comment out the opacity and transform property. See what happens. The opacity makes the description panel disappear and the transform makes it resize to zero size. Take out the comments when you've observed in the browser

7. The title of the item will have some fitting paddings and margins and a smooth text shadow:

```
.ch-info h3 {
    color: #fff;
    text-transform: uppercase;
    letter-spacing: 2px;
```

```

    font-size: 20px;
    margin: 0 30px;
    padding: 45px 0 0 0;
    height: 140px;
    font-family: 'Open Sans', Arial, sans-serif;
    text-shadow: 0 0 1px #fff, 0 1px 2px rgba(0,0,0,0.3);
}

```

8. The paragraph element has 0 opacity and a transition (we want to fade it in on hover but with a delay):

```

.ch-info p {
    color: #fff;
    padding: 10px 5px;
    font-style: italic;
    margin: 0 30px;
    font-size: 12px;
    border-top: 1px solid rgba(255,255,255,0.5);
    opacity: 0;
    transition: all 1s ease-in-out 0.4s;
}

```

9. The link will be in uppercase letters and we'll make the hover colour yellow:

```

.ch-info p a {
    display: block;
    text-decoration: none;
    color: #fff;
    color: rgba(255,255,255,0.7);
    font-style: normal;
    font-weight: 700;
    text-transform: uppercase;
    font-size: 9px;
    letter-spacing: 1px;
    padding-top: 4px;
    font-family: 'Open Sans', Arial, sans-serif;
}

```

```
.ch-info p a:hover {
    color: #fff222;
    color: rgba(255,242,34, 0.8);
}
```

10. And now, the interesting hover action

The item will animate its box shadow's spread radius from 16px to 1px:

```
.ch-item:hover {
    box-shadow: inset 0 0 0 1px rgba(255,255,255,0.1), 0 1px 2px
    rgba(0,0,0,0.1);
}
```

Remember the transition kicks in when the property changes. In this case the box-shadow property has changed so the transition set up in the .ch-item class kicks in as the box-shadow property has changed

11. The description will fade in and scale up to 1:

```
.ch-item:hover .ch-info {
    transform: scale(1);
    opacity: 1;
}
```

Remember the transition kicks in when the property changes. In this case the opacity has changed so the transition set up in the .ch-item class kicks in as the opacity property has changed. We can now see the element as the property has changed from 0 to 1.

12. And the paragraph of the description will just fade in (with a delay):

```
.ch-item:hover .ch-info p {
    opacity: 1;
}
```

13. Fin.