

42 (styles.css) changing the border-radius, for three  
border-radius: 30px 60px 25px 12px; → this is incorrect!  
border-radius: 30px 25px 60px 12px; → this is correct :)  
↓ ↓ ↓ ↘ bottom-left  
top-left top-right bottom-right

43 (styles.css) using transform property on the .one selector to rotate it  
counter clockwise by 0.6 degrees.  
transform: rotate(-0.6 deg);  
↘ counter

44 (styles.css) rotating the .two element clockwise by 0.4°  
transform: rotate(0.4 deg);

Wednesday, 24<sup>th</sup> of Jan 2024.

CSS Flexbox - Photo Gallery

1 adding DOCTYPE, html, head, and body elements.

2 <meta name="viewport" content="width=device-width, initial-scale=1">  
<meta charset="utf-8">

3 adding title element and linking element to styles.css

<title></title>

<link rel="stylesheet" href="styles.css">

4 adding header + assign a class, and adding <h1></h1>

<header class="header">

<h1></h1>

</header>

5 adding <div class="gallery"></div> and adding <img> x 9 inside it. ↗ nine.

6 adding  x 9

7 (styles.css) styling the .gallery → width: 50; border: 5px solid red;

img → width: 100%; padding: 5px; border: 5px solid blue;

8 (styles.css) \* {  
↗ global selector.

box-sizing: content-box; → I do not completely understand.

}

9 (styles.css) changing the content-box into → border-box.

10 (styles.css) removing the .gallery and img selectors & all the rules within.

⑪ (styles.css) adding new selector

```
.gallery img {  
  width: 100%;  
  max-width: 350px;  
  height: 300px;  
}
```

⑫ (styles.css) styling the header

```
.header {  
  text-align: center;  
  text-transform: uppercase;  
  padding: 32px;  
  background-color: #0a0a23;  
  color: #fff;  
  border-bottom: 4px solid #fdb347;  
}
```

⑬ (styles.css) removing the margin from the body element, set the font, and add background-color.

```
body {  
  margin: 0;  
  font-family: sans-serif;  
  background-color: #f5f6f7;  
}
```

⑭ (styles.css) adding flexbox.

Flexbox → one-dimensional CSS layout that can control the way items are spaced out & aligned with a container.

To use it, give an element a display property of flex. This will make the element a **flex container**. Any direct children of a flex container are called **flex items**.

```
.gallery {  
  display: flex;  
}
```

flex-direction: row;

⑮ flex-wrap: wrap;

⑯ justify-content: center;

⑰ (styles.css) #flex-direction:

- row (default): horizontal axis left to right [→]
- row-reverse: horizontal axis right to left [←]
- column: vertical axis top to bottom [↓]
- column-reverse: vertical axis bottom to top [↑]

⑱ (styles.css) align-items: center;

⑲ (styles.css) padding: 20px 10px;  
max-width: 1400px;  
margin: 0 auto;

} for .gallery

⑳ (styles.css) [for .gallery img] → object-fit: cover;

㉑ (styles.css) adding gap CSS shorthand property for .gallery flex container. gap: 16px;



②② (styles.css) adding border-radius: 10px; for .gallery img.

②③ (styles.css) adding ::after pseudo-element. This creates an element that is the last child of the selected element. [I don't understand this]

example →

```
.container ::after {  
  content: " ";  
  width: 860px;  
}
```

```
.gallery ::after {  
  content: " ";  
  width: 350px;  
}
```

\* typography for nutrition label → 67 steps

\* quiz → 68 steps ⇒ TRIBUTE PAGE.

\* CSS pseudo selectors → 66 steps.

\* CSS cat painting → 82 steps.

\* Piano → 33 steps ⇒ TECHNICAL DOCUMENTATION PAGE

\* CSS variables city skyline → 118 steps

\* CSS magazine → 80 steps ⇒ PRODUCT LANDING PAGE

\* CSS ferris wheel → 29 steps

\* CSS penguin → 104 steps

total steps: 647 steps 😊