

CSS Exercises

Exercise #1

In this task, you have an image that is overflowing the box.

- We want the image to scale down to fit inside the box without any extra white space, but we do not mind if some part of the image is cropped.

```
/* CSS Styles */
.box {
  width: 350px;
  height: 450px;
  border: 5px solid black;
}

img {
}

<!-- HTML Code-->
<div class="box">
  
</div>
```

Hint: use `object-fit` and note that you have to set the `width` and `height` to 100% as well because `object-fit` property doesn't affect the size of the `` tag — it only affects how the image is shown inside the `` tag.



Exercise #2

Create a similar looking table and style it to be as the preview image.

- You may need to use ``table-layout`` and ``border-collapse`` properties.

Band	Year formed	No. of Albums	Most famous song
Buzzcocks	1976	9	Ever fallen in love (with someone you shouldn't've)
The Clash	1976	6	London Calling
The Stranglers	1974	17	No More Heroes
Total albums			77

A summary of the UK's most famous punk bands

Exercise #3

Use the given HTML structure with the following instructions.

- 1) There's no padding in the container div element.
- 2) Use the float property to make the nested div elements float to the left, you should also clear/fix the layout in the container div CSS styling place somehow after doing so.
- 3) The horizontal and vertical gaps between the nested div elements is 15px each.
- 4) The width of the container div element should not be calculated manually and should be dynamic (use 100% unit).
- 5) Ensure that the layout is not broken when you resize the page.

```
<div class="container">
  <div class="full-width">Full Width</div>
  <div class="one-third">1/3</div>
  <div class="one-third">1/3</div>
  <div class="one-third">1/3</div>
  <div class="one-half">1/2</div>
  <div class="one-half">1/2</div>
  <div class="one-fourth">1/4</div>
  <div class="one-fourth">1/4</div>
  <div class="one-fourth">1/4</div>
  <div class="one-fourth">1/4</div>
</div>
```



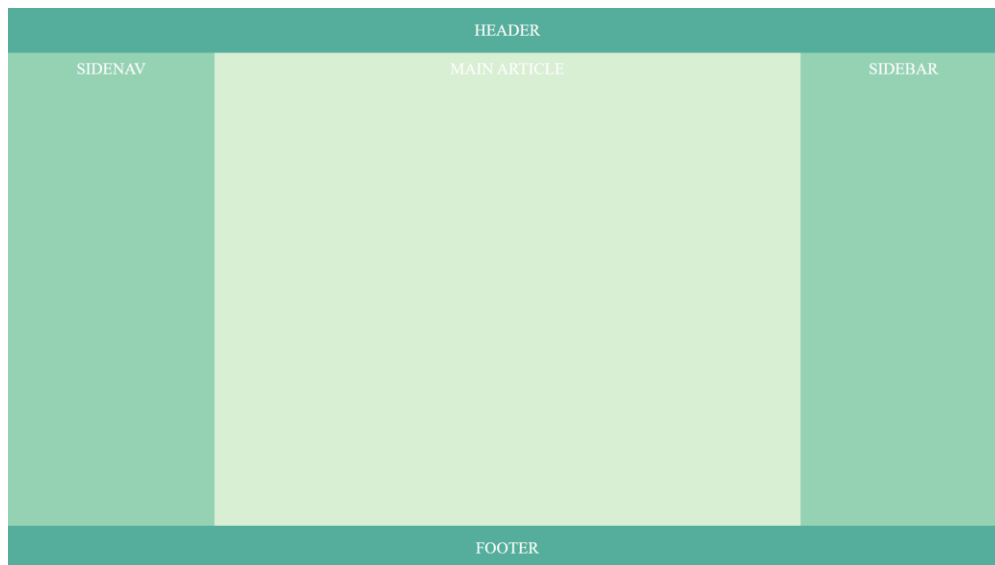
Exercise #4

Use the given HTML structure with the following instructions.

- 1) Using flex-box properties, create a [holy-grail design](#) similar to the preview image below.
- 2) Use media queries to adjust the direction of the middle part to be column when the screen size reaches 500px or less.

```
<div class="flex-header">HEADER</div>
<div class="flex-main">
  <div class="flex-nav">SIDENAV</div>
  <div class="flex-article">MAIN ARTICLE</div>
  <div class="flex-aside">SIDEBAR</div>
</div>
<div class="flex-footer">FOOTER</div>
```

Used Colors: #55AD9B, #95D2B3, #D8EFD3



Exercise #5

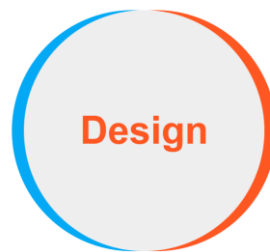
Use the given HTML structure with the following instructions.

- 1) The width and height of the div element is 100px.
- 2) Center the div element horizontally and vertically on the page using flex properties.
- 3) Center the contents (i.e., the word "Design") both horizontally and vertically using flex properties as well.

```
<div>Design</div>
```

Hint: use the [box-shadow](#) property.

Used Colors: #03a9f4, #ff5722



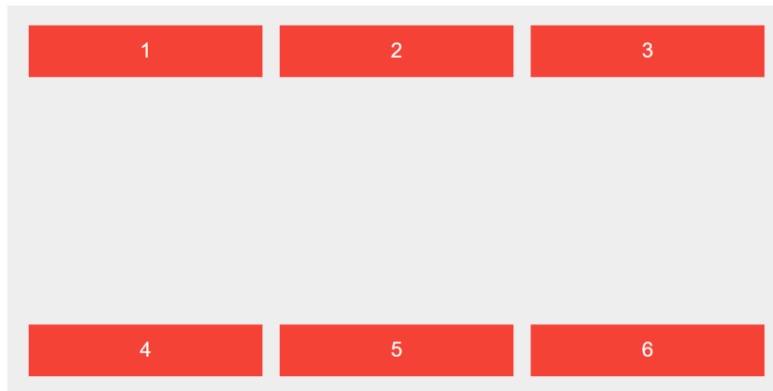
Exercise #6

Use the given HTML structure with the following instructions.

- 1) The width of the container div element is 600px and the height is 300px.
- 2) Center the container div element horizontally and vertically in the page using flex properties.
- 3) The minimum width and minimum height of each nested div element inside the container div is 180px and 40px, respectively.
- 4) Use flex properties to arrange the nested div elements like the attached image.
- 5) The gap between the nested div elements should be 10px.

```
<div class="container">  
  <div>1</div>  
  <div>2</div>  
  <div>3</div>  
  <div>4</div>  
  <div>5</div>  
  <div>6</div>  
</div>
```

Used Colors: #eeeeee, #f44336



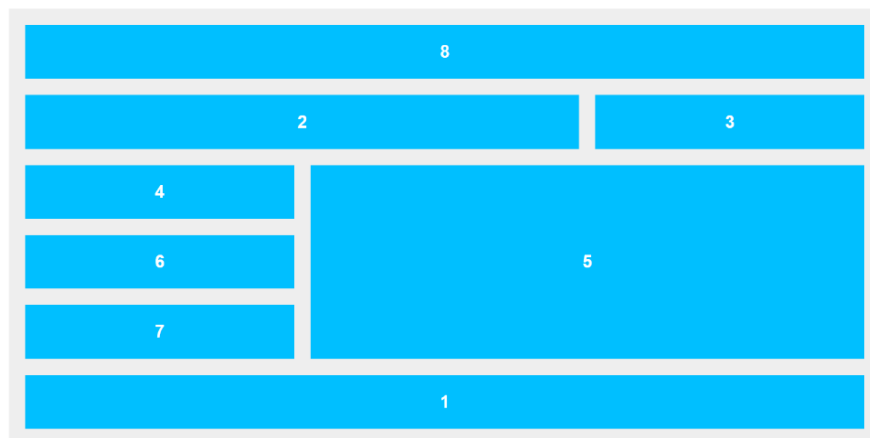
Exercise #7

Use the given HTML structure with the following instructions.

- 1) Center the div element with the class of 'grid' horizontally and vertically on the page using grid properties.
- 2) Use the grid properties to arrange the nested div elements to create a design close to the attached preview image.

```
<div class="grid">  
  <div class="one">1</div>  
  <div class="two">2</div>  
  <div class="three">3</div>  
  <div class="four">4</div>  
  <div class="five">5</div>  
  <div class="six">6</div>  
  <div class="seven">7</div>  
  <div class="eight">8</div>  
</div>
```

Used Colors: #eeeeee, #00bfff



Exercise #8

Use the given HTML structure with the following instructions.

- 1) Using both grid and flex properties, create a [holy-grail design](#) similar to the preview image below.
- 2) Define the grid canvas to have 3 grid rows with values of 50px, automatic, and 50px, respectively.
- 3) The grid canvas should also have 10 grid columns each filling the whole width *greedily*.
- 4) You should use [grid areas property](#) to define the 10 columns.
- 5) You may use flexbox for minor tasks (e.g., centering text or items).
- 6) Use CSS variables to define the three used colors.

```
<div class="logo">Logo</div>
<ul class="navbar">
  <li>Home</li>
  <li>About</li>
  <li>Services</li>
  <li>Contact</li>
</ul>
<div class="main">MAIN CONTENT</div>
<div class="sidebar">SIDEBAR</div>
<div class="footer">Copyright &copy; 2024</div>
```

Tip: you may refer to `'auto'`, `'1fr'`, and `'var()'`.

Used Colors: #55AD9B, #95D2B3, #D8EFD3

