

Creating layouts

- ✓ Video: CSS web layout 4 min
- ✓ Reading: Understanding flexbox 20 min
- ✓ Reading: CSS units of measurement 10 min
- ✓ Video: Basic flexbox 5 min
- ✓ Video: Flex charts 7 min
- ✓ Video: CSS grids 5 min
- ✓ Reading: Grids and flexbox cheat sheet 15 min
- ✓ Video: Grid showcase 8 min
- ✓ Reading: Grid template area 20 min
- ✓ Lab: Create a grid layout 1h

Create a grid layout

[Open Lab](#)

Instructions

Introduction

In this lab, you will be creating a grid layout commonly called the Holy Grail layout.

Goal

- To create a grid layout using the grid template area

Objectives

- Add styling to the CSS page with the help of the grid template area
- Configure rules for different properties inside the grid template area

Instructions

Initial code for the HTML is already provided. The rules for different areas within the grid template area are already provided in the CSS code. You have to add the rules for the container class as per the instructions provided.

The two sets of rules to be added for the container class will have one set for regular **container** class and another using media query for a different size. Follow the instructions below.

First, write rules by adding properties for the **container** class as below.

Step 1: Add a display property that will create a grid.

```
1 display: grid;
```

Step 2: It should have a maximum width of 900 pixels.

```
1 max-width: 900px;
```

Step 3: The minimum height for it will be the length of 50 viewport height.

```
1 min-height: 50vh;
```

Step 4: Add a property for grid template columns that will span 100 % of the width.

```
1 grid-template-columns: 100%;
```

Step 5: Add a property for grid template values for five rows, of which the middle one will have a value of 1 fractional area and the rest will be set to auto.

```
1 grid-template-rows: auto auto 1fr auto auto;
```

Step 6: Finally, you will create a grid template area that will contain five values: **header**, **left**, **main**, **right** and **footer**.

```
1 grid-template-areas: "header" "left" "main" "right" "footer";
```

Similar to the rules you have defined above, you will again add a different set of rules inside the media query when the minimum width of the viewport is 440 pixels.

Step 7: For the container class the three grid template columns will have respective values of 150 pixels, 1 fractional area and 150 pixels again.

```
1 grid-template-columns: 150px 1fr 150px;
```

Step 8: For the three grid template rows, the middle value should be 1 fractional, while the two others will be set to auto.

```
1 grid-template-rows: auto 1fr auto;
```

Step 9: This time, you will be creating a 3 x 3 grid template area that will have only header in the first row. It will have **left**, **main** and **right** in the second row and finally have only **footer** in the last row.

```
1 grid-template-areas: "header header header"
2                     "left main right"
3                     "footer footer footer";
```

Finally, you should preview your code and see the final output. Try resizing your browser viewport to see how the layout changes.

