5.2 Grid layout

Grid container and grid items

Grid layout is a CSS layout mode that divides a webpage into a rectangular grid in which to position page elements. Grid layout is ideal for designing two-dimensional webpage layouts.

A ***grid container*** is an element that has the CSS property `display` set to `grid` to create a block-level grid container or `inline-grid` to create an inline grid container. Ex: `<div style="display: grid">`. A ***grid item*** is a child element of a grid container that is by default placed into a single grid cell.

The ***grid-template-columns*** property defines the grid container's number of columns and optionally the width of each column. Ex: `grid-template-columns: 50px 90px auto auto;` specifies 4 values that create 4 columns: the first is 50px wide, the second is 90px wide, and the third and fourth columns are automatically sized to fit the remainder of the grid width.

PARTICIPATION ACTIVITY 5.2.1: Grid layout example.

1 2 3 2x speed

```
<div id="grid-container">
   <div>1</div>
   <div>2</div>
   <div>3</div>
   <div>4</div>
   <div>5</div>
   <div>6</div>
</div>
```

```
#grid-container {
   display: grid;
   grid-template-columns: auto auto auto;
}

#grid-container > div {
   text-align: center;
   background: lightgreen;
   border: 3px solid green;
   padding: 20px;
}
```

1	2	3
4	5	6

Six elements are children of the grid container, so each element becomes a grid item. The div child selector puts a green border around each grid item.

Captions

1. The "display: grid;" declaration makes the with id grid-container a block-level grid.
2. The grid-template-columns property is assigned 3 "auto" values, so the grid container will contain 3 equally-sized columns.

3. Six elements are children of the grid container, so each element becomes a grid item. The div child selector puts a green border around each grid item.

Feedback?

PARTICIPATION ACTIVITY 5.2.2: Grid layout basics.

Refer to the animation above.

1) If `#grid-container` is modified, how many columns will the grid have?

```
#grid-container {
   display: grid;
   grid-template-
columns: auto
auto;
}
```

- (●) 2
- ( ) 3
- ( ) 4

Correct

`grid-template-columns` lists 2 "auto" values, so the grid will have 2 columns. The 6 grid items will fill 3 rows.

1	2
3	4
5	6

2) If `#grid-container` is modified, how wide is the second column?

```
#grid-container {
   width: 600px;
   display: grid;
   grid-template-
columns: 200px
auto;
}
```

- ( ) 200px
- (●) 400px
- ( ) 600px

Correct

The first column is 200px wide, so the second column "auto" value makes the second column fill the remaining grid container width. Since the grid container is 600px wide, the second column is 600px - 200px = 400px wide.

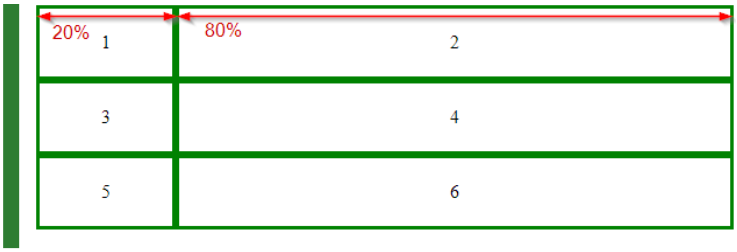
200px 1	400px 2
3	4
5	6

3) If `#grid-container` is modified, how wide is the second column?

Correct

The first column is 20% of the grid container's width = 600px × 20% = 120px. So the second column "auto" value makes the second column fill the remaining 80% of the grid container width = 600px × 80% = 480px.

```
#grid-container {
   width: 600px;
   display: grid;
   grid-template-
columns: 20%
auto;
}
```



- ○ 120px
- ◉ 480px
- ○ 600px

4) If the grid container has 10 grid items, what `grid-template-columns` value creates a grid with 2 rows and 5 columns?

```
#grid-container {
   width: 600px;
   display: grid;
   grid-template-
columns: _____;
}
```

- ○ 5
- ○ 200px auto
- ◉ 200px auto auto auto 200px

Correct

Five values are specified, so the grid has 5 columns.

1	2	3	4	5
6	7	8	9	10

Feedback?

PARTICIPATION ACTIVITY 5.2.3: Arrange the photos with a grid layout.

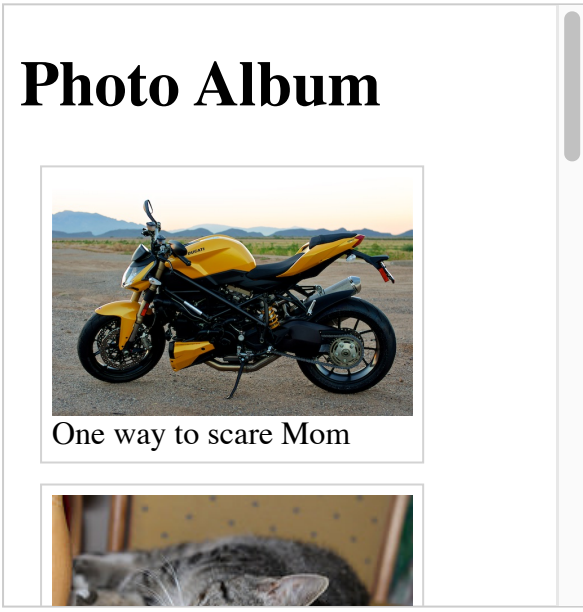
Edit the CSS so the outer div becomes a grid layout container that displays the photos and captions in two columns.

HTML CSS

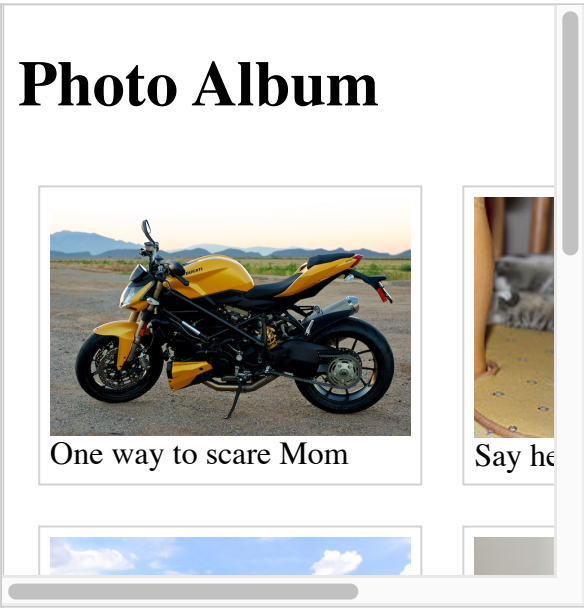
```
<body>
   <h1>Photo Album</h1>
   <div id="grid-container">
      <figure>
         <img src="https://resources.zybooks.com/WebProgrammir
         <figcaption>One way to scare Mom</figcaption>
      </figure>
      <figure>
         <img src="https://resources.zybooks.com/WebProgrammir
         <figcaption>Say hello to Daisy</figcaption>
      </figure>
      <figure>
         <img src="https://resources.zybooks.com/WebProgrammir
         <figcaption>Last summer we hiked to a beautiful lake
      </figure>
      <figure>
```

Render webpage Reset code

Your webpage



Expected webpage



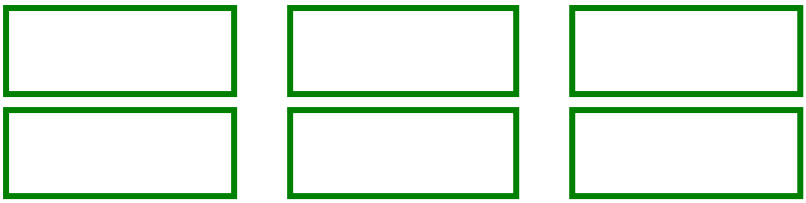
▶ View solution

Feedback?

Controlling the grid container

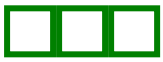
The default behavior of a grid container can be modified with various CSS properties:

- The ***gap*** property defines the gap between each grid row and column. Ex: `gap: 5px 25px;` puts a 5px gap between each row and a 25px gap between each column.



- The ***grid-template-rows*** property defines the height of each row. Ex: `grid-template-rows: 20px 40px;` makes the first row 20px tall and the second row 40px tall.

- The ***justify-content*** property horizontally aligns the grid items inside the grid container using values:
  - `start` - Aligns grid flush with the grid container's starting edge.



  - `end` - Aligns grid flush with the grid container's ending edge.

  - `center` - Aligns grid in the center of the grid container.

  - `stretch` - Stretches the grid items to fill the grid container width.

  - `space-around` - Places equal spacing between grid items with half the space on either side of the grid container.

  - `space-between` - Places equal spacing between grid items with no space on either side of the grid container.

  - `space-evenly` - Places equal spacing between grid items, including the sides of the grid container.

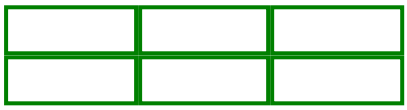
- The ***align-content*** property vertically aligns the grid items inside the grid container using values:

- `start` - Aligns grid flush with the grid container's starting edge.

- `end` - Aligns grid flush with the grid container's ending edge.

- `center` - Aligns grid in the center of the grid container.

- `stretch` - Stretches the grid items to fill the grid container height.



- `space-around` - Places equal spacing between grid items with half the space on either side of the grid container.

- `space-between` - Places equal spacing between grid items with no space on either side of the grid container.

- `space-evenly` - Places equal spacing between grid items, including the sides of the grid container.

The `justify-content` and `align-content` properties have no effect unless the grid width or height is less than the grid container's width or height.

PARTICIPATION ACTIVITY 5.2.4: Modify the grid container.

Refer to the HTML and CSS below:

```
<style>
#grid-container {
   width: 300px;
   height: 70px;
   background: mediumseagreen;
   display: grid;
   grid-template-columns: auto auto auto;
}

#grid-container > div {
   background: lightgreen;
}
</style>

<body>
   <div id="grid-container">
      <div>A</div>
      <div>B</div>
      <div>C</div>
      <div>D</div>
      <div>E</div>
      <div>F</div>
   </div>
</body>
```

1) Add a CSS declaration to `#grid-container` so the first row is 20 pixels tall and the second row is 30 pixels tall.

grid-template-rows : `20px 30px;`

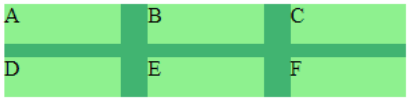
Check Show answer

Correct

grid-template-rows

The `grid-template-rows` property sets the height of the first 2 rows to 20px and 30px. The height of the grid container is larger than the combined height of the grid items, so the grid container's background color displays at the grid bottom.

2) Put a 10 pixel gap between each row and 20 pixel gap between each column.



`gap:` 10px 20px `;`

Check Show answer

Correct

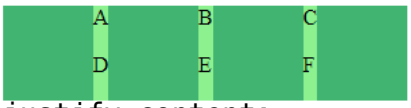
10px 20px

The first number is the row gap (10px), and the second number the column gap (20px). The CSS properties `row-gap` and `column-gap` can also be used to set the row and column gaps. Ex: `row-gap: 10px;`

3) Place an equal amount of horizontal space between each grid item, including the grid edges.

Correct

space-evenly



justify-content:
space-evenly ;

`space-evenly` places equal spacing around each grid item. An alternative value `space-between` does the same as `space-evenly` except no space is left on the grid edges.

Check Show answer

4) Center each grid item vertically in the grid.

align-content : center;

Correct

align-content

The `align-content` property can also use values `start` and `end` to align the grid items flush to the starting edge or ending edge of the grid.

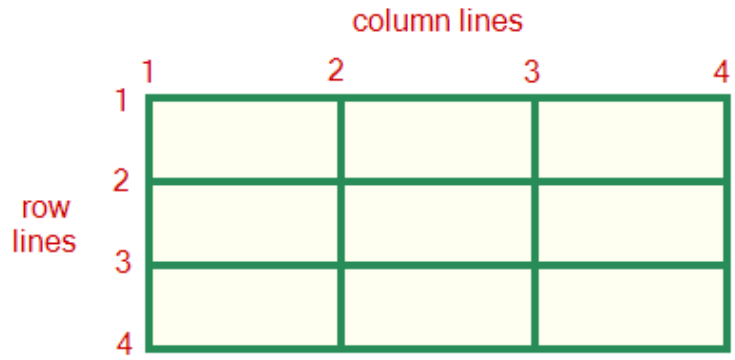
Check Show answer

Feedback?

Controlling grid item placement

A grid item by default appears in a single row and column based on the ordering of the grid item within the grid container. However, grid items may be positioned at specific grid locations using the column line and row line numbers as illustrated in the figure below.

Figure 5.2.1: Row and column lines.



Feedback?

A grid item may be placed in a specific row or column or span multiple rows and/or columns using various following CSS properties:

- The ***grid-row*** property lists the grid item's starting and ending row line numbers. Ex: `grid-row: 1 / 3;` makes the grid item start at row line 1 and end at row line 3, so

the grid item spans 2 rows.

- The ***grid-column*** property lists the grid item's starting and ending column line numbers. Ex: `grid-column: 1 / 4;` makes the grid item start at column line 1 and end at column line 4, so the grid item spans 3 columns.
- The ***grid-area*** property lists the grid item's starting and ending row and column numbers. Ex: `grid-area: 1 / 2 / 3 / 4;` makes the grid item start at row line 1 and column line 2 and end at row line 3 and column line 4, so the grid item spans 2 rows and 2 columns.

PARTICIPATION ACTIVITY 5.2.5: Rearrange the grid.

Add the following CSS modifications to alter the grid:

1. Add the following CSS to move the A item to the 1st column and 3rd row:

```
#grid-item-a {
   grid-row: 3 / 4;
}
```

   Render the webpage and observe the A item is previously where the G item was located.

2. Add the following CSS to make the B item occupy the first 3 rows of the 3rd column:

```
#grid-item-b {
   grid-column: 3 / 4;
   grid-row: 1 / 4;
}
```

   Render the webpage and observe that B is on the right edge of the grid, and a 4th row is created with an empty grid cell below B.

3. Add the following CSS to make the C item occupy the first 3 rows and 2 columns:

```
#grid-item-c {
   grid-area: 1 / 1 / 4 / 3;
}
```

   Render the webpage and observe the webpage matches the expected webpage.

HTML CSS

```
11    background: powderblue;
12    padding: 10px;
13 }
14 #grid-item-a {
15    grid-row: 3 / 4;
16 }
17
18 #grid-item-b {
19    grid-column: 3 / 4;
20    grid-row: 1 / 4;
21 }
22
23 #grid-item-c {
24    grid-area: 1 / 1 / 4 / 3;
25 }
26
```

Render webpage Reset code

Your webpage

C B

F G H

Expected webpage

C B

F G H

▼ View solution

Explain

```
--- START FILE: HTML ---

<body>
   <div id="grid-container">
      <div id="grid-item-a">A</div>
      <div id="grid-item-b">B</div>
      <div id="grid-item-c">C</div>
      <div>D</div>
      <div>E</div>
      <div>F</div>
```

```
        <div>G</div>
        <div>H</div>
        <div>I</div>
    </div>
</body>


--- END FILE: HTML ---

--- START FILE: CSS ---


#grid-container {
    width: 400px;
    font-size: 2em;
    background: cornflowerblue;
    display: grid;
    grid-template-columns: auto auto auto;
    gap: 5px 5px;
}

#grid-container > div {
    background: powderblue;
    padding: 10px;
}

#grid-item-a {
    grid-row: 3 / 4;
}

#grid-item-b {
    grid-column: 3 / 4;
    grid-row: 1 / 4;
}

#grid-item-c {
    grid-area: 1 / 1 / 4 / 3;
}


--- END FILE: CSS ---
```

Feedback?

PARTICIPATION ACTIVITY 5.2.6: Modify the grid layout.

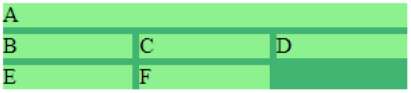
Refer to the HTML and CSS below:

```
<style>
#grid-container {
   width: 300px;
   background: mediumseagreen;
   display: grid;
   grid-template-columns: auto auto auto;
   gap: 5px 5px;
}

#grid-container > div {
   background: lightgreen;
}
</style>

<body>
   <div id="grid-container">
      <div id="grid-item-a">A</div>
      <div id="grid-item-b">B</div>
      <div id="grid-item-c">C</div>
      <div>D</div>
      <div>E</div>
      <div>F</div>
   </div>
</body>
```

1) Make the A item occupy the entire top row.




```
#grid-item-a {
   grid-column:
```
1/4 ;
```
}
```

Check **Show answer**

Correct

1 / 4

The first number is the starting column line, and the second number is the ending column line.

2) Make the C item occupy the top 2 rows in the 2nd column.


```
#grid-item-c {
   grid-column: 2 / 3;
   grid-row:
```
1/3 ;
```
}
```

Check **Show answer**

Correct

1 / 3

The first number is the starting row line, and the second number is the ending row line.

3) Make the B item occupy 2 rows and 2 columns in the

Correct

bottom-right grid corner.


```
#grid-item-b {
   grid-area:
2 / 2 / 4 /4 ;
}
```

Check Show answer

2 / 2 / 4 / 4

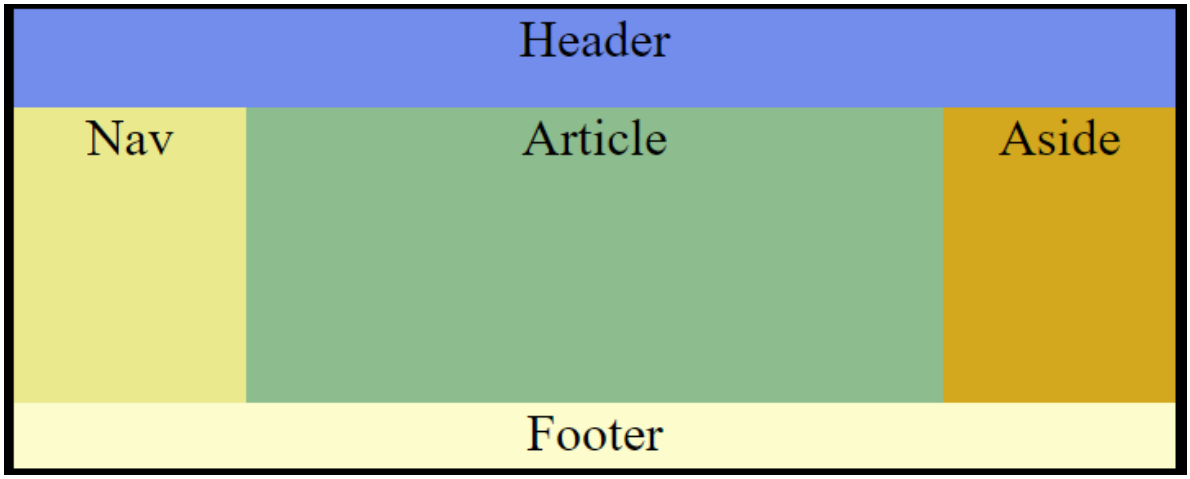
The starting row and column lines are both 2, and ending row and column lines are both 4. `grid-area` is a shorthand property that can replace a `grid-column` and `grid-row` declaration.

Feedback?

Naming grid items

Grid items may be assigned names with the `grid-area` property. The grid container's ***grid-template-areas*** property specifies the grid layout using the named grid items.

Figure 5.2.2: Layout goal for the Participation Activity below.



Feedback?

PARTICIPATION ACTIVITY 5.2.7: Grid layout using named grid items.

Add the following CSS modifications to create a grid layout for the webpage that matches the figure above:

1. Add the following CSS to the `body` selector to change the `<body>` into a grid container with 3 columns and 3 rows:

```
body {
   ...
   display: grid;
   grid-template-columns: 150px auto 150px;
   grid-template-rows: 60px 180px 40px;
}
```

Render the webpage and observe that the five grid items occupy 2 rows.

2. Use the `grid-area` property to name each of the 5 grid items:

```
header {
   background: cornflowerblue;
   grid-area: head;
}

nav {
   background: khaki;
   grid-area: nav;
}

article {
   background: darkseagreen;
   grid-area: article;
}

aside {
   background: goldenrod;
   grid-area: aside;
}

footer {
   background: lemonchiffon;
   grid-area: foot;
}
```

3. Use the `grid-template-areas` property on the grid container to layout the grid items. Each row is specified in a single string with the grid item names.

```
body {
   ...
   grid-template-areas:
      "head head head"
      "nav article aside"
      "foot foot foot";
}
```

Render the webpage, which should look like the figure above. The Header occupies all 3 columns on the top row; the Nav, Article, and Aside occupy the 3 columns in the second row; and the Footer occupies all 3 columns on the bottom row.

HTML CSS

```
<body>
   <header>Header</header>
   <nav>Nav</nav>
   <article>Article</article>
   <aside>Aside</aside>
   <footer>Footer</footer>
</body>
```

Render webpage Reset code

Your webpage

Header
Nav
Article
Aside
Footer

▼ View solution

Explain

```
--- START FILE: HTML ---

<body>
   <header>Header</header>
   <nav>Nav</nav>
   <article>Article</article>
   <aside>Aside</aside>
   <footer>Footer</footer>
</body>
```

```
--- END FILE: HTML ---

--- START FILE: CSS ---

body {
   background: black;
   font-size: 2em;
   text-align: center;
   display: grid;
   grid-template-columns: 150px auto 150px;
   grid-template-rows: 60px 180px 40px;
   grid-template-areas:
      "head head head"
      "nav article aside"
      "foot foot foot";
}

header {
   background: cornflowerblue;
   grid-area: head;
}

nav {
   background: khaki;
   grid-area: nav;
}

article {
   background: darkseagreen;
   grid-area: article;
}

aside {
   background: goldenrod;
   grid-area: aside;
}

footer {
   background: lemonchiffon;
   grid-area: foot;
}


--- END FILE: CSS ---
```

Feedback?

PARTICIPATION ACTIVITY 5.2.8: Named grid items.

Refer to the Participation Activity above.

1) What `grid-template-areas` value makes the Article span 2 columns?

- `"head head head" "nav article aside" "foot article foot"`
- `"head article head" "nav article article aside" "foot foot foot"`
- `"head head head" "nav article article" "foot foot foot"`

Correct

`article` appears twice in the second string, so Article spans 2 columns on the second row.

2) A single period can be used in a `grid-template-areas` value to represent an empty grid cell. Which `grid-template-areas` value places an empty grid cell under Nav?

- `". head head" "nav article aside" "foot foot foot"`
- `"head head head" "nav . aside" "foot foot foot"`
- `"head head head" "nav article aside" ". foot foot"`

Correct

The period is below `nav`, so the empty grid cell appears below Nav.

3) What is the correct way to name the new grid item `<section>`?

```
section {
   grid-area:
____;
}
```

- `"mysection"`
- `mysection`
- `'mysection'`

Correct

The grid item name must use alphanumeric characters, dashes, or underscores and not be quoted.

Feedback?

CHALLENGE ACTIVITY 5.2.1: Grid layout.

530096.4000608.qx3zqy7

Jump to level 1

1

Use the grid-template-areas property to make the grid render as shown in the expected webpage. **SHOW EXPECTED**

2 3 4

CSS HTML

```
1  #container {
2     display: grid;
3     text-align: center;
4     grid-template-columns: 50px 50px 50px;
5     grid-template-areas:"a a d" "b c d" "b e e";
6     /* Your solution goes here */
7
8  }
9
10 #item-a {
11    background: cornflowerblue;
12    grid-area: a;
13 }
14
15 #item-b {
16    background: khaki;
```

1 2 3 **4**

Check Next **Done**. Click any level to practice more. Completion is preserved.

✔ Testing grid-template-areas property on container

Yours `"a a d" "b c d" "b e e"`

Your webpage

A D
B C
E

View your last submission

```
grid-template-areas:"a a d" "b c d" "b e e"
/* Your solution goes here */
```

Feedback?

Exploring further:

- CSS Grid Layout Module from W3Schools
- CSS Grid Layout Browser Support from caniuse.com

How was this section? Provide section feedback