

Test your skills: Grid

The aim of this skill test is to assess whether you understand how a [grid and grid items](#) behave. You will be working through several small tasks that use different elements of the material you have just covered.

Note: You can try out solutions in the interactive editors below. However, it may be helpful to download the code and use an online tool such as [CodePen](#) , [jsFiddle](#) , or [Glitch](#) to work on the tasks.

If you get stuck, then ask us for help — see the [Assessment or further help](#) section at the bottom of this page.

Task 1

In this task, you should create a grid into which the four child elements will auto-place. The grid should have three columns sharing the available space equally and a 20-pixel gap between the column and row tracks. After that, try adding more child containers inside the parent container with the class of `grid` and see how they behave by default.

Your final result should look like the image below:



Try updating the live code below to recreate the finished example:

One

Two

Three

Four

```
.grid {  
  display:grid;  
  grid-template-columns: 1fr 1fr 1fr;  
  gap:20px  
}
```

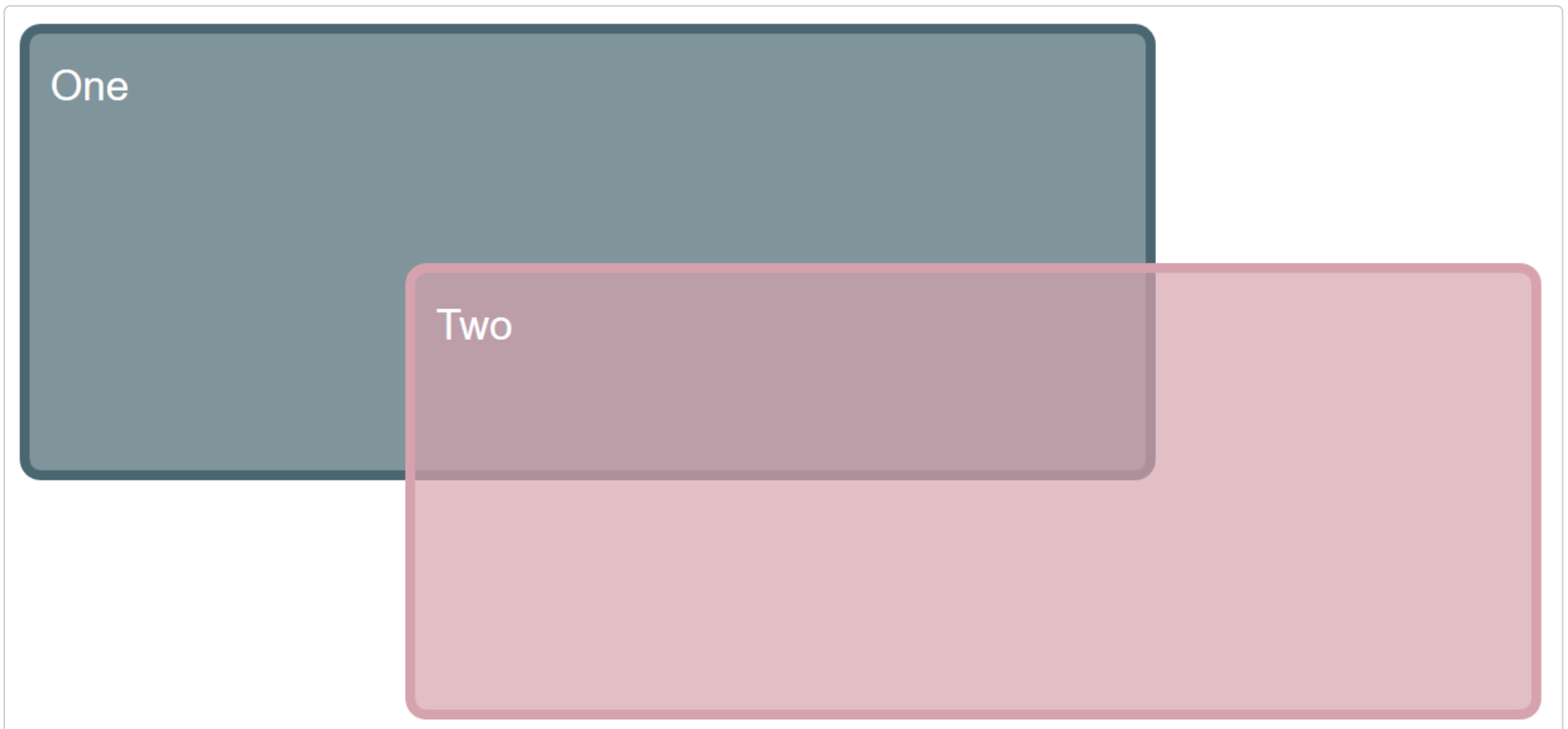
```
<div class="grid">  
  <div>One</div>  
  <div>Two</div>  
  <div>Three</div>  
  <div>Four</div>  
</div>
```

[Reset](#)

[Download the starting point for this task](#) to work in your own editor or in an online editor.

Task 2

In this task, we already have a grid defined. By editing the CSS rules for the two child elements, cause them to span over several grid tracks each. The second item should overlay the first as in the image below:



Try updating the live code below to recreate the finished example:



```
gap: 10px;
}

.item1 {
grid-column: 1/4;
grid-row: 1/3;
order: 1
}

.item2 {
grid-column: 2/5;
grid-row: 2/4
}
```

```
<div class="grid">
  <div class="item1">One</div>
  <div class="item2">Two</div>
</div>
```



Additional question:

- Can you now cause the first item to display on top without changing the order of items in the source?

[Download the starting point for this task](#) to work in your own editor or in an online editor.

Task 3

In this task, there are four direct children in this grid. The starting point has them displayed using auto-placement. Use the `grid-area` and `grid-template-areas` properties to lay the items out as shown in the image below:



Try updating the live code below to recreate the finished example:



```
.grid {  
  display: grid;  
  grid-template-columns: 1fr 2fr;  
  gap: 10px;  
  grid-template-areas:  
    "one one"  
    "two three"  
    ". four";  
}
```

```
,  
.one{  
  grid-area: one;  
}  
  
<div class="grid">  
  <div class="one">One</div>  
  <div class="two">Two</div>  
  <div class="three">Three</div>  
  <div class="four">Four</div>  
</div>
```


Reset

[Download the starting point for this task](#) to work in your own editor or in an online editor.

Task 4

In this task, you will need to use both grid layout and flexbox to recreate the example as seen in the image below. The gap between the column and row tracks should be 10px. You do not need to make any changes to the HTML in order to achieve this.



balloon

red

sky

blue

Hot air balloon



balloons

houses

train

harborside



balloons

inflating

green

blue



balloon

sun

sky

summer

bright

Try updating the live code below to recreate the finished example:



balloon red sky
blue Hot air balloon



balloons houses
train harborside



balloons inflating
green blue



balloon sun sky
summer bright

```
.container {  
  display:grid;  
  grid-template-columns:repeat(3, 1fr);  
  gap:10px  
}
```

```
.tags {  
  display:flex;  
  flex-flow: row wrap;  
  justify-content: center;  
}
```

```
      <li>balloons</li>  
      <li>houses</li>  
      <li>train</li>  
      <li>harborside</li>  
    </ul>  
  </div>  
  <div class="card">  
      
    <ul class="tags">  
      <li>balloons</li>  
      <li>close-up</li>  
    </ul>  
  </div>  
</div>
```

//



//

Reset

[Download the starting point for this task](#) to work in your own editor or in an online editor.

Assessment or further help

You can practice these examples in the Interactive Editors above.

If you would like your work assessed or are stuck and want to ask for help:

1. Put your work into an online shareable editor such as [CodePen](#) , [jsFiddle](#) , or [Glitch](#) . You can write the code yourself or use the starting point files linked to in the above sections.
2. Write a post asking for assessment and/or help at the [MDN Discourse forum Learning category](#) . Your post should include:
 - A descriptive title such as "Assessment wanted for grid skill test 1".
 - Details of what you have already tried and what you would like us to do; for example, tell us if you're stuck and need help or want an assessment.
 - A link to the example you want assessed or need help with, in an online shareable editor (as mentioned in step 1 above). This is a good practice to get into — it's very hard to help someone with a coding problem if you can't see their code.
 - A link to the actual task or assessment page, so we can find the question you want help with.

Last modified: Jul 11, 2022, [by MDN contributors](#)

