

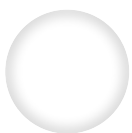
2.4.9 Use Alignment Properties to Position Grid Content

Now that we know how to set up a grid and position and size grid items, let's investigate how alignment properties can position their content.

IMPORTANT

The `float`, `display: inline-block`, `vertical-align` and `column-*` properties have no effect on a grid item.

The following grid properties are used on the grid container element to align the grid items on a certain axis.



REWIND

The same properties were used by flexbox!

The justify-items Property

Arranges grid items along the row axis. This value applies to all the grid items in the grid container.

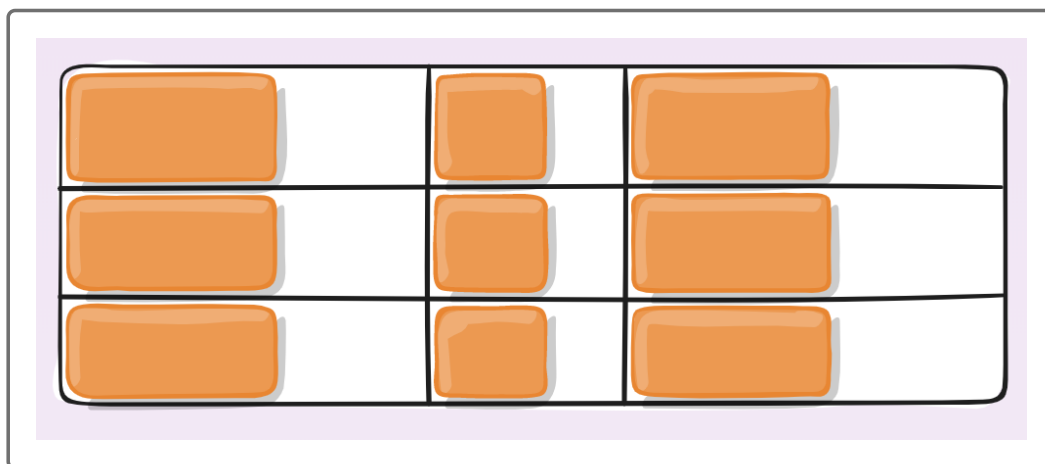
Here's how this property is written and the values it accepts:

```
.grid-container {  
  justify-items: start | end | center | stretch;  
}
```

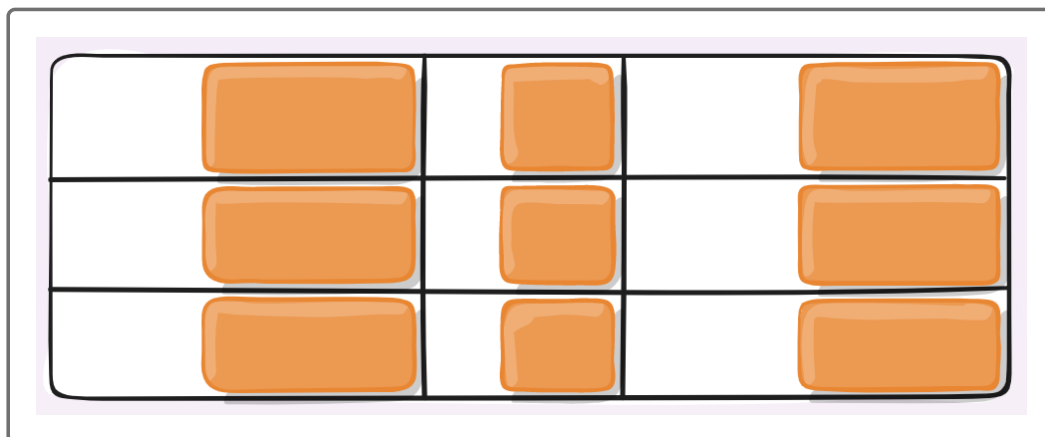
HIDE HINT

In the declaration, the "|" (or pipe) is simply separating the four possible property values that `justify-items` accepts.

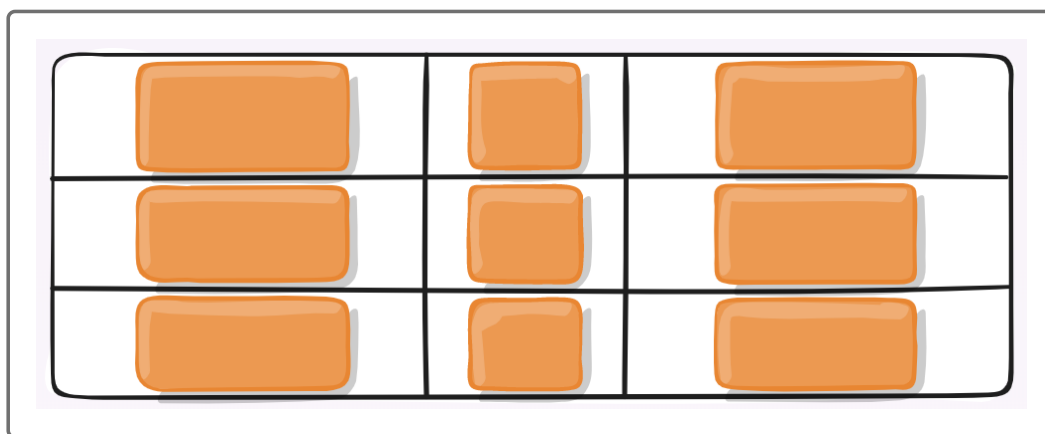
- The `start` value aligns items at the start edge of the grid cell:



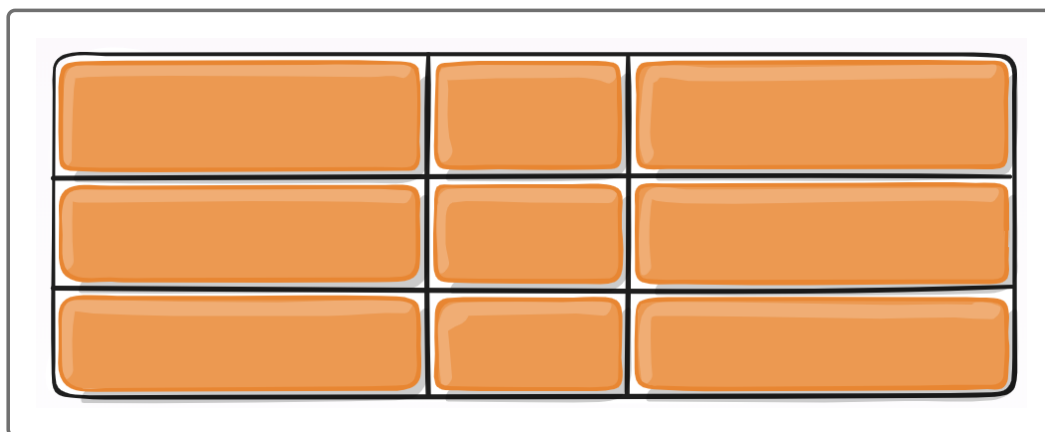
- The `end` value aligns items at the end edge of the grid cell:



- The `center` value centers items of the grid cell:



- The `stretch` value fills the width of the cell (default value):



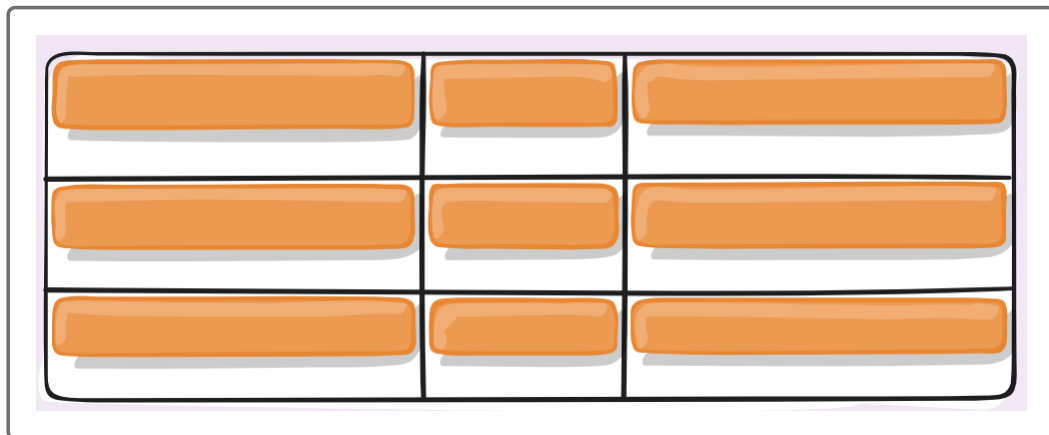
The align-items Property

Places grid items along the column axis. This value applies to all the grid items in the grid container.

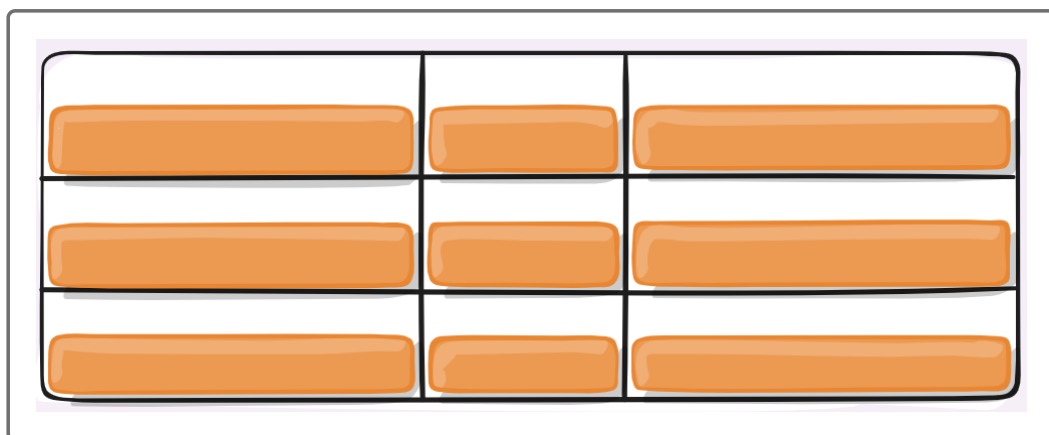
Here's how this property is written and the values it accepts:

```
.grid-container {  
  align-items: start | end | center | stretch;  
}
```

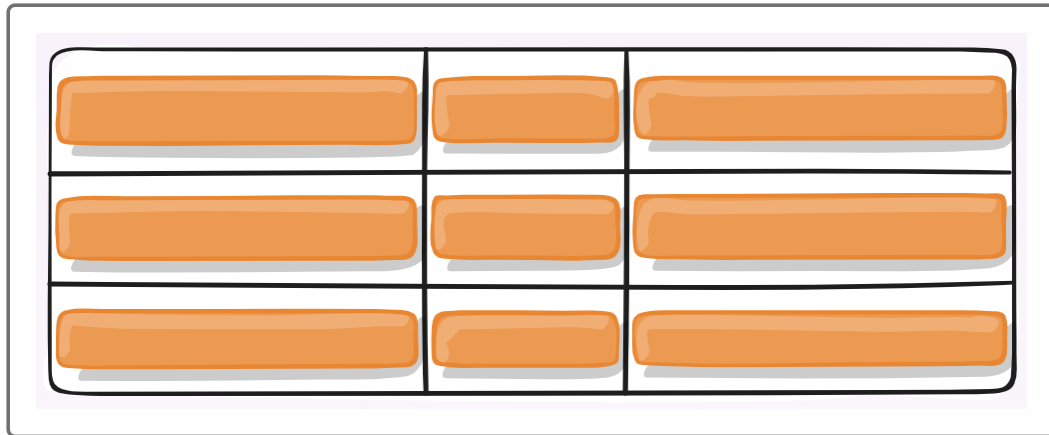
- The **start** value aligns items at the top edge of the grid cell.



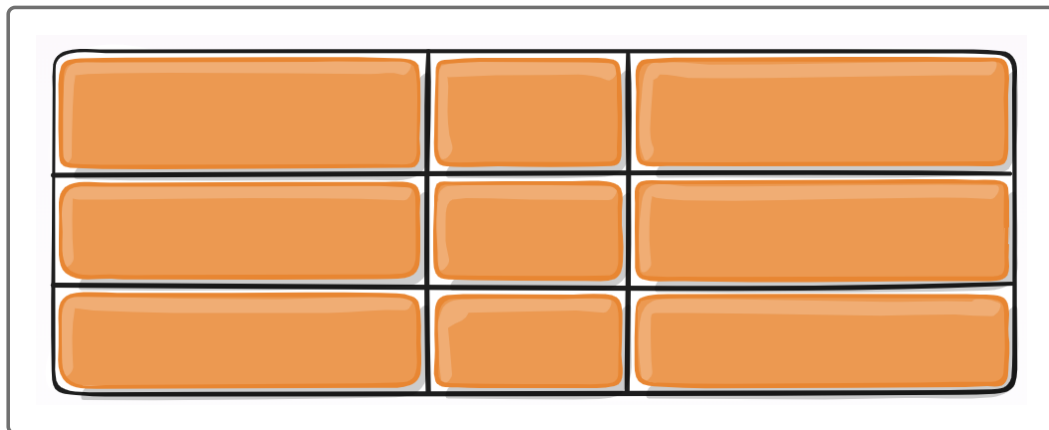
- The **end** value aligns items at the bottom edge of the grid cell.



- The `center` value centers items of the grid cell.



- The `stretch` value fills the height of the cell (default value).



The place-items Property

This is shorthand notation that combines two declarations with one property that is space-separated. If only one value is present, this value will be assigned to both properties.

Here's how this property is written and the values it accepts:

```
.grid-container {  
  place-items: <align-items> <justify-items>;  
}
```

}

DEEP DIVE ▲

DEEP DIVE

Alignment can also be customized on the grid-item level using similar property axis labels. Instead of `*-items`, we will use the `*-self` post-fix for the `align-self` and `justify-self` properties. To learn more, see the [MDN web docs on the align-self property](https://developer.mozilla.org/en-US/docs/Web/CSS/align-self). (<https://developer.mozilla.org/en-US/docs/Web/CSS/align-self>).

Let's pause for an add, commit, and push to Git. Use the following commands to do so:

```
git add .  
git commit -m "example grid"  
git push origin feature/grid
```

Since we are finished with our grid example and our mock-up doesn't contain any orange boxes with numbers, let's remove the CSS declarations in the example's rules in our style sheet so that they look like this:

```
.service-grid-container {  
  
}  
  
.service-grid-item {
```

```
}
```

You can also go ahead and delete the rules for `.box1`, `.box2`, and `.box3`, as we won't be using them moving forward.

© 2020 Trilogy Education Services, a 2U, Inc. brand. All Rights Reserved.