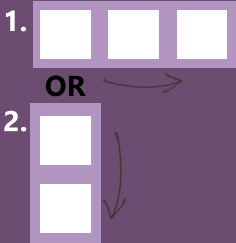


Flexbox

(display: flex;)



HTML

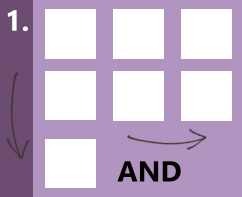
```
<div id="wrapper">
  <div class="box"> </div>
  <div class="box"> </div>
  <div class="box"> </div>
</div>
```

CSS

```
#wrapper{
  display: flex;
  flex-direction: row;
}
```

CSS Grids

(display: grid;)



HTML

```
<div id="wrapper">
  <div class="box"> </div>
  <div class="box"> </div>
  <div class="box"> </div>
  <div class="box"> </div>
  <div class="box"> </div>
  <div class="box"> </div>
  <div class="box"> </div>
  <div class="box"> </div>
</div>
```

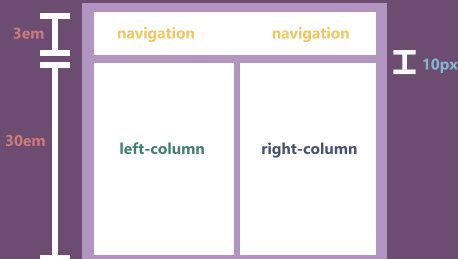
CSS

```
#wrapper{
  display: grid;
  grid-template-columns: 1fr 1fr 1fr;
}
```

GRID TEMPLATE AREAS

HTML

```
<div id="wrapper">
  <div class="navigation"> </div>
  <div class="left-column"> </div>
  <div class="right-column"> </div>
</div>
```



CSS

```
.navigation{
  grid-area: navigation;
}

.left-column{
  grid-area: left-column;
}

.right-column{
  grid-area: right-column;
}

#wrapper{
  display: grid;
  grid-template-columns: 1fr 1fr;
  grid-template-rows: 3em 30em;
  grid-template-areas:
    "navigation navigation"
    "left-column right-column";
  gap: 10px;
}
```

Grid-template-areas - Defines the layout of a grid area by referencing items within with a grid (grid-area).

Before using the "grid-template-areas" property on the parent wrapper, add the "grid-area" property to the child elements and give them a name.

Once all the child elements have a grid-area name, you can add the "grid-template-areas" property on the parent element and reference the grid area names to position these items.

Gap - Sets the space between columns and rows just like margin.

Grid-template-columns - Defines the size and amount of columns within a grid container. The amount of columns is determined by the number of values added to the grid template columns property, along with space in between the values.

POSITIONING ITEMS

*both flex and grid use these properties

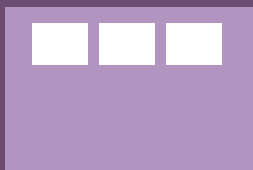
justify-content:

flex: justify-content: flex-start;
grids: justify-content: start;



align-content:

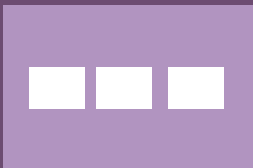
flex: align-content: flex-start;
grids: align-content: start;



flex: justify-content: center;
grids: justify-content: center;



flex: align-content: center;
grids: align-content: center;



flex: justify-content: flex-end;
grids: justify-content: end;



flex: align-content: flex-end;
grids: align-content: end;

