

CSC 337

2:00 PM

ALRIGHT, TIME TO OVERHAUL THE SITE! IT'S A BIG CHANGE, BUT AS LONG AS I'M CAREFUL AND BACK-UP AT EVERY STEP, THERE SHOULDN'T REALLY BE ANY WAY TO SCREW THIS UP.



2:03 PM

OKAY, I'VE LOST THE LAST THREE YEARS OF CONTENT AND THE ENTIRE SITE IS NOW FORMATTED SIX INCHES TO THE RIGHT FOR NO APPARENT REASON.

WONDERFUL.



10:57 PM

IT WAS A LONG UPHILL BATTLE, BUT EVERYTHING IS RESTORED AND THE NEW LAYOUT IS ALMOST DONE. JUST GOTTA GET THESE FIVE BUTTONS TO APPEAR IN A STRAIGHT LINE AND THAT'S IT.



2:00 AM

...ROBYN? ARE YOU STILL UP?

```
.navi {  
  text-align: center;  
  width: 100px;  
  padding-top: 24px;  
  display: inline-block;  
}
```



CSS Layout
Rick Mercer

Basic CSS Layout

- We used a bit of CSS layout techniques

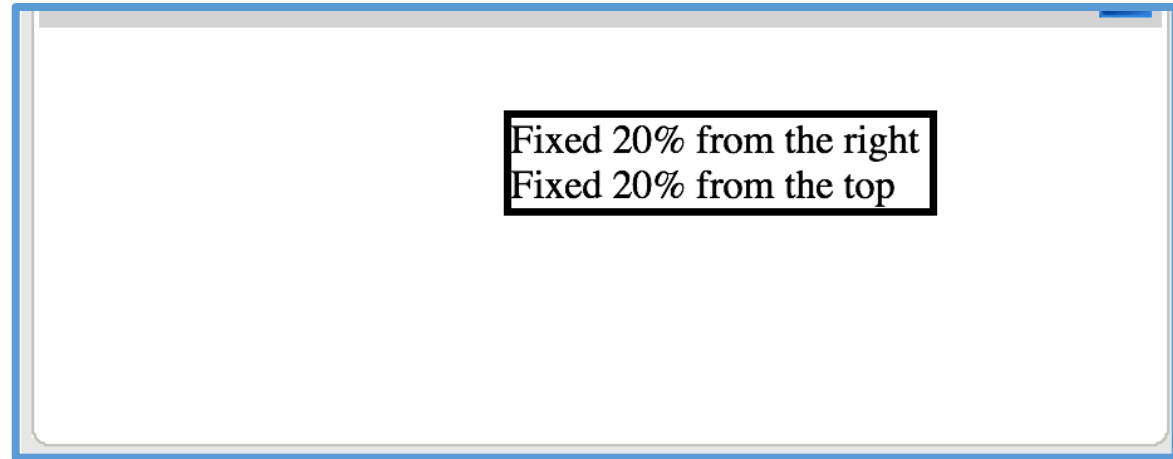
`float: left;`
`float: right;`

- We did not cover

`position: absolute;`
`position: relative;`
`position: fixed;`

The position property fixed

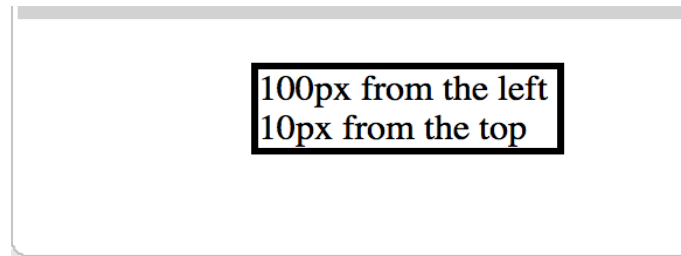
```
.aClass {  
  position: fixed;  
  right: 20%;  
  top: 20%;  
  border: solid;  
}
```



property	value	description
position	static	default position
	relative	offset from its normal static position
	absolute	a fixed position within its containing element
	fixed	a fixed position within the browser window
<u>top</u> , <u>bottom</u> , <u>left</u> , <u>right</u>	positions of box's corners	

position: absolute

```
.abs {  
  position: absolute;  
  left: 100px;  
  top: 20px;  
  border: solid;  
}
```



- Removed from normal flow, like float
- Position relative the the block element containing them
- Actual position determined by top, bottom left or right values

Alignment vs. float vs. position

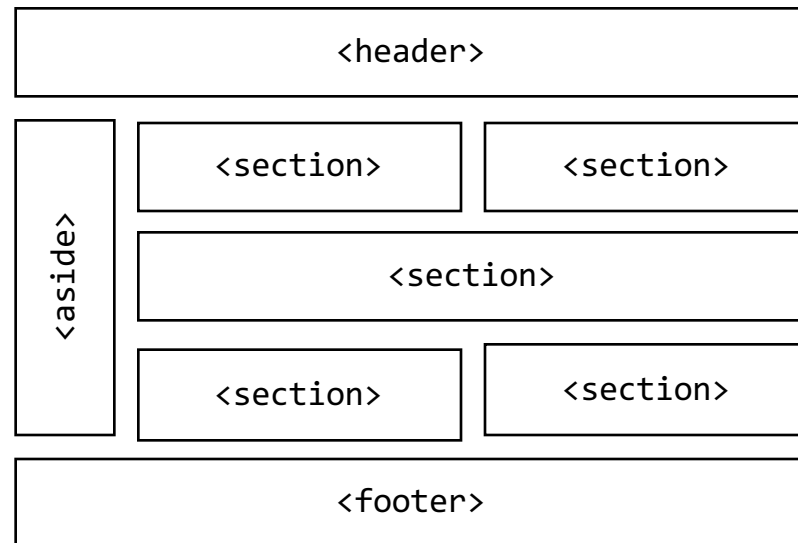
- if possible, lay out an element by aligning its content with text-align
 - it aligns the content within it (not the block itself)
- if alignment won't work, try floating the element
- if floating won't work, try positioning the element
 - absolute/fixed positioning are a last resort and should not be overused
- Better yet, use grid CSS

Grid Layout



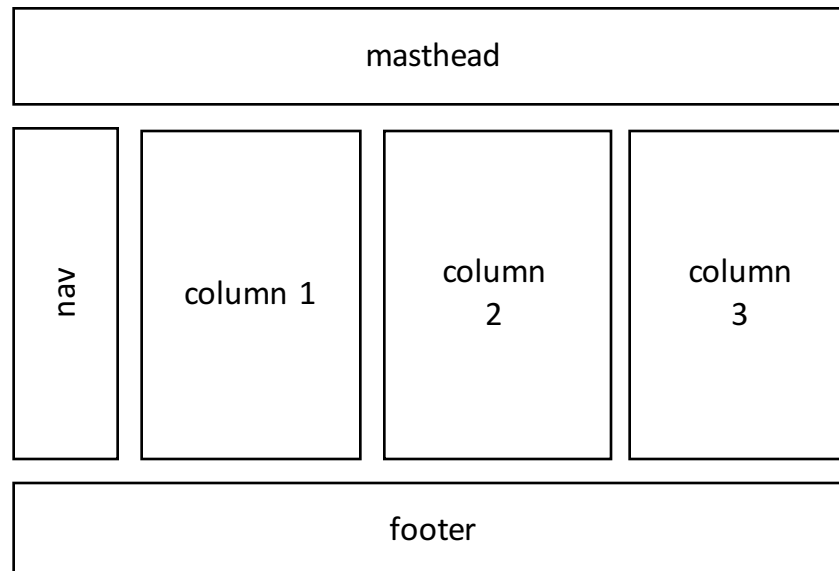
The Grid Layout Model

- The Grid Layout model uses CSS to structure content using rows and columns
- Grids are extremely flexible and provide an easier to use option for organizing content than HTML tables
- Most powerful layout system available in CSS



Grid layouts and Grid Items

- Grid layouts are very similar to tables because they feature rows and columns
- They are best suited for more complex layouts
- The content in grid layouts are also modular, which allows you to move content from one part of the page to another

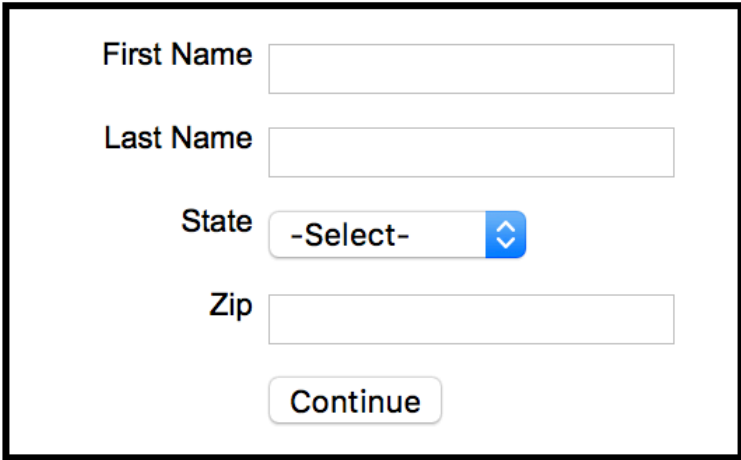


A simple form without grid

- This form used this css file

```
.section {  
  width: 240px;  
  height: 140px;  
  padding: 10px;  
  border: solid black 2pt;  
  font-family: sans-serif;  
}
```

```
.labels {  
  float: left;  
  width: 80px;  
  height: 30px;  
  font-family: Arial;  
  font-size: 8pt;  
  text-align: right;  
}
```

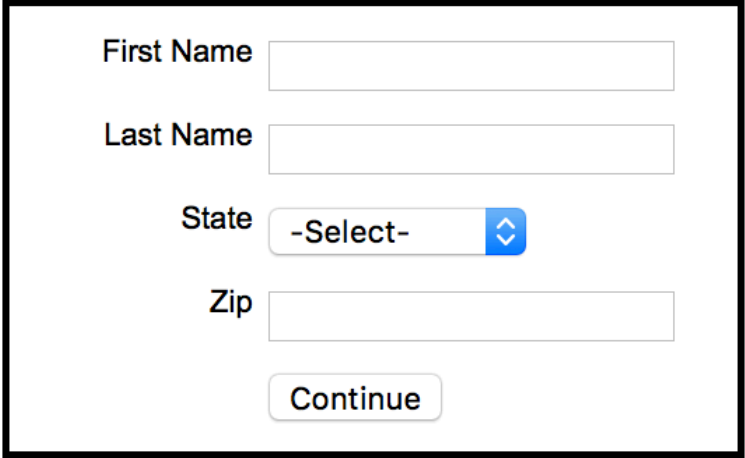


The image shows a visual representation of the CSS form. It is a rectangular box with a black border. Inside, the labels 'First Name', 'Last Name', 'State', and 'Zip' are aligned to the right of the box. Each label is followed by an input field. The 'State' field is a dropdown menu with '-Select-' and up/down arrows. At the bottom, there is a 'Continue' button.

```
.fields {  
  float: right;  
  width: 160px;  
  height: 30px;  
  font-family: Arial;  
  font-size: 10pt;  
}
```

display: grid;

```
form {  
  display: grid;  
  grid-template-columns:  
    80px 140px;  
}  
  
.section {  
  /* same */  
}  
  
.labels {  
  grid-column: 1 / 2;  
  font-family: Arial;  
  font-size: 8pt;  
  text-align: right;  
}
```



First Name

Last Name

State

Zip

```
.fields {  
  grid-column: 2 / 2;  
  height: 30px;  
  font-family: Arial;  
  font-size: 10pt;  
}
```

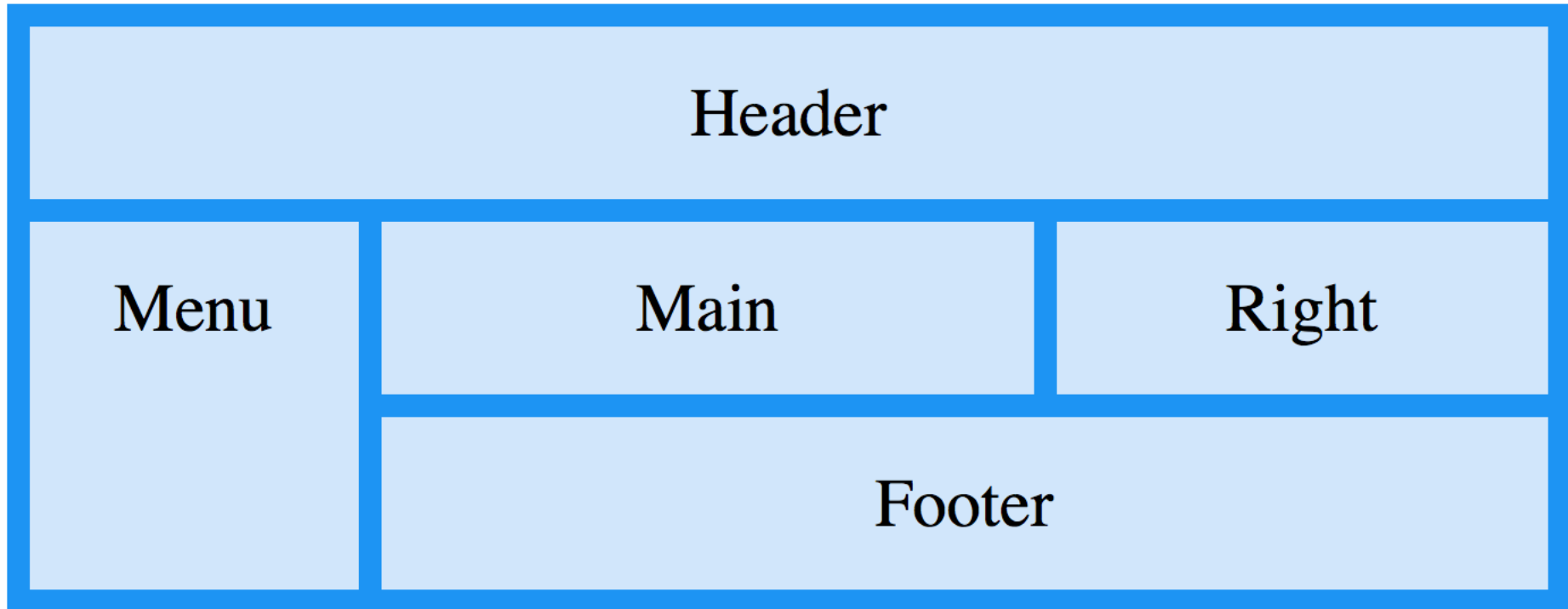
The Grid Layout Model

- CSS Grid Layout excels at dividing a page into major regions, or defining the relationship in terms of size, position, and layer, between parts of a control built from HTML primitives
- Like tables, grid layout enables an author to align elements into columns and rows
- Many more layouts are either possible or easier with CSS grid than they were with tables.
- A grid container's child elements could position themselves so they actually overlap and layer

An Abstract Example Layout w3Schools

Grid Layout

This grid layout contains six columns and three rows:



An Abstract Example Layout

```
.item1 { grid-area: header; }
.item2 { grid-area: menu; }
.item3 { grid-area: main; }
.item4 { grid-area: right; }
.item5 { grid-area: footer; }

.grid-container {
  display: grid;
  grid-template-areas:
    'header header header header header header'
    'menu main main main right right'
    'menu footer footer footer footer footer';
  grid-gap: 10px;
  background-color: #2196F3;
  padding: 10px;
}

.grid-container>div {
  background-color: rgba(255, 255, 255, 0.8);
  text-align: center;
  padding: 20px 0;
  font-size: 30px;
}
```

```
<body>

<h1>Grid Layout</h1>

<p>This grid layout contains
  six columns and three rows:</p>

<div class="grid-container">
  <div class="item1">Header</div>
  <div class="item2">Menu</div>
  <div class="item3">Main</div>
  <div class="item4">Right</div>
  <div class="item5">Footer</div>
</div>

</body>
</html>
```

See another way to layout out divs

https://www.w3schools.com/css/tryit.asp?filename=trycss_grid_item