CSS Layout Part 2

Web Development

Lab 07: Learn CSS Layout Part 2

01. position

Vel et enim consulatu. Te civibus copiosae salutandi vel. Adhuc sonet libris ad eam, mundi affert mea ex. Dicunt feugiat patrioque et mel, id qui nusquam maluisset, ei vim justo ceteros vituperata. Mei saepe mediocrem ut. Repudiare definitiones ea ius, sint commodo est ea, nam no nemore diceret.

Te iriure moderatius vis, nam prodesset honestatis te. Atqui facilisi at est. Ex duo vocent incorrupte eloquentiam. Agam deterruisset vel at, has no illum ipsum alterum. Virtute vivendo officiis his et, ius viris tollit homero ad. In sit euismod salutatus, cu eos malorum luptatum consulatu, et nec debet antiopam.

Saperet maiestatis instructior te per, cu vel tota cotidieque. Vix illum regione deterruisset cu, ne cum diam suavitate complectitur, nec ex erant principes. Augue omittam no sea, putant forensibus usu te. Te iusto dicam verear mei. Dolorum posidonium no vel.

This element is relatively-positioned. If this element was position: static; its absolutely-positioned child would escape and would be positioned relative to the document body.

This element is absolutely-positioned. It's positioned relative to its parent.

margin: auto;

Lorem ipsum dolor sit amet, eos ut diam interesset, cu modo necessitatibus pri. Ne sit elit dicit, eum dico autem convenire an. Sed ei clita nullam, elit legimus voluptatibus ei his. Duo facilisi cotidieque at, invidunt platonem incorrupte ut has.

Vel et enim consulatu. Te civibus copiosae salutandi vel. Adhuc sonet libris ad eam, mundi affert mea ex. Dicunt feugiat patrioque et mel, id qui nusquam maluisset, ei vim justo ceteros vituperata. Mei saepe mediocrem ut. Repudiare definitiones ea ius, sint commodo est ea, nam no nemore diceret.

the box model

I'm smaller...

And I'm bigger!
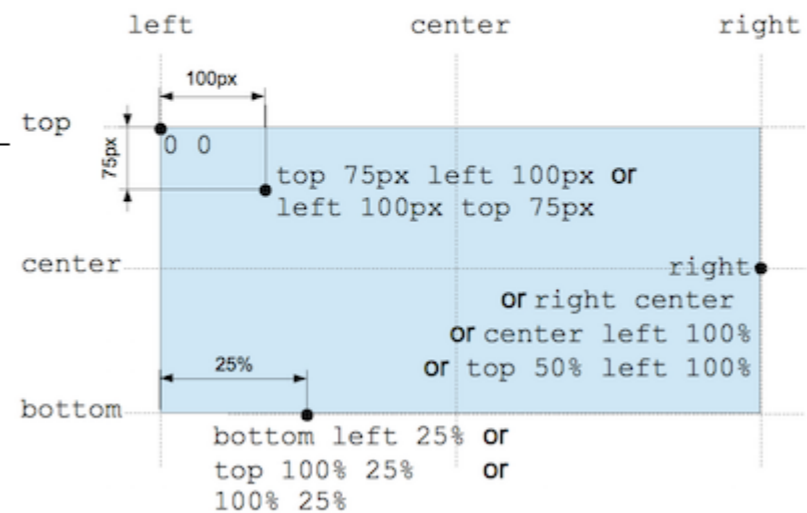
box sizing

We're the same size now!

Hooray!

CSS Layout Part 2

- The clearfix hack

- Float layout example

- Percent width

- Mobile Site Design

- Media queries Inline-block

- Viewport Meta Tag

- Inline-block layout

- Column

- Flexbox

- Further Reading

```
<div>
  <p>
    <img src="css-positioning.png" alt="An Image">
    Lorem ipsum dolor sit amet, consectetur adipiscing elit. Phasellus imperdiet, Uh oh... this image is taller
    than the element containing it, and it's floated, so it's overflowing outside of its container!
  </p>
</div>
```

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Phasellus imperdiet, Uh oh... this image is taller than the element containing it, and it's floated, so it's overflowing outside of its container!

left   center   right

top

100px
75px

0 0
top 75px left 100px or
left 100px top 75px

center   right

or right center
or center left 100%
or top 50% left 100%

25%

bottom

bottom left 25% or
top 100% 25%   or
100% 25%

```
img {
  float: right;
}
```

Bad things can sometimes happen when using floats:
In this case an image overflows the container, we used float right for the image within the <p>, however because it is taller than the text it overflows.

overflow Property

- The overflow property specifies what happens if content overflows an element's box.

- This property specifies whether to clip content or to add scrollbars when an element's content is too big to fit in a specified area.
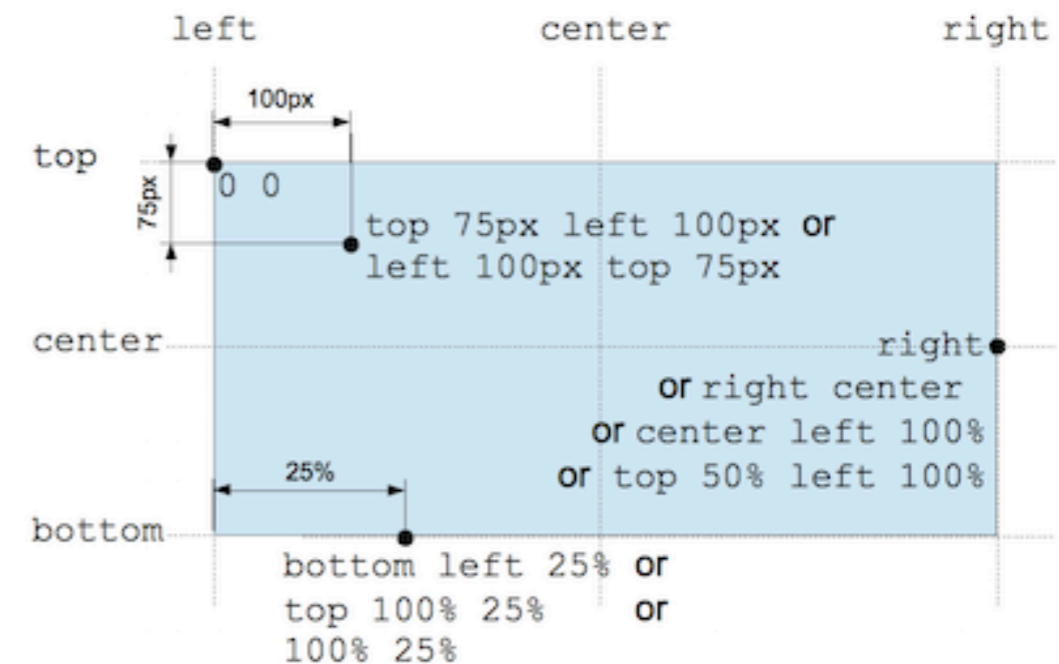
Value	Description
visible	The overflow is not clipped. It renders outside the element's box. This is default
hidden	The overflow is clipped, and the rest of the content will be invisible
scroll	The overflow is clipped, but a scroll-bar is added to see the rest of the content
auto	If overflow is clipped, a scroll-bar should be added to see the rest of the content
initial	Sets this property to its default value. Read about *initial*
inherit	Inherits this property from its parent element. Read about *inherit*

```html
<div class="clearfix">
  <p>
    <img src="css-positioning.png" alt="An Image">
    Lorem ipsum dolor sit amet, consectetur adipiscing elit. Phasellus imperdiet, Uh oh... this image is taller
    than the element containing it, and it's floated, so it's overflowing outside of its container!
  </p>
</div>
```

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Phasellus imperdiet, Uh oh... this image is taller than the element containing it, and it's floated, so it's overflowing outside of its container!


```css
img {
  float: right;
}

.clearfix{
    overflow: auto;
}
```

Float layout example

- It is common to do entire layouts using float. Here is the same layout we did with position last week, but using float instead.

```html
<h1> 12. float layout example </h1>
<div class="clearfix">
  <nav>
    <ul>
      <li> <a href="#">Home</a>       </li>
      <li> <a href="#">Taco Menu</a>  </li>
      <li> <a href="#">Draft List</a> </li>
      <li> <a href="#">Hours</a>      </li>
      <li> <a href="#">Directions</a> </li>
      <li><a href="#">Contact</a>     </li>
    </ul>
  </nav>
  <section>
    <p>
      This example works just like the last one. Notice we put a clearfix on the container. It's not needed in this
      example, but it would be if the nav was longer than the non-floated content.
    </p>
  </section>
  <section>
    <p>
      Lorem ipsum dolor sit amet, consectetur adipiscing elit. Phasellus imperdiet, nulla et dictum interdum, nisi
      lorem egestas odio, vitae scelerisque enim ligula venenatis dolor. Maecenas nisl est, ultrices nec congue
      eget, auctor vitae massa. Fusce luctus vestibulum augue ut aliquet. Mauris ante ligula, facilisis sed ornare
      eu, lobortis in odio. Praesent convallis urna a lacus interdum ut hendrerit risus congue. Nunc sagittis
      dictum nisi, sed ullamcorper ipsum dignissim ac. In at libero sed nunc venenatis imperdiet sed ornare turpis.
      Donec vitae dui eget tellus gravida venenatis. Integer fringilla congue eros non fermentum. Sed dapibus
      pulvinar nibh tempor porta. Cras ac leo purus. Mauris quis diam velit.
    </p>
  </section>
</div>
```

12. float layout example

- Home
- Taco Menu
- Draft List
- Hours
- Directions
- Contact

This example works just like the last one. Notice we put a clearfix on the container. It's not needed in this ex be if the nav was longer than the non-floated content.

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Phasellus imperdiet, nulla et dictum interdum, nisi l vitae scelerisque enim ligula venenatis dolor. Maecenas nisl est, ultrices nec congue eget, auctor vitae massa vestibulum augue ut aliquet. Mauris ante ligula, facilisis sed ornare eu, lobortis in odio. Praesent convallis u ut hendrerit risus congue. Nunc sagittis dictum nisi, sed ullamcorper ipsum dignissim ac. In at libero sed nu imperdiet sed ornare turpis. Donec vitae dui eget tellus gravida venenatis. Integer fringilla congue eros non dapibus pulvinar nibh tempor porta. Cras ac leo purus. Mauris quis diam velit.

```css
nav {
  float: left;
  width: 200px;
  border: solid 1px;
}

section {
  margin-left: 200px;
  border: solid 1px;
}

.clearfix {
  overflow: auto;
}
```

percent width

- Percent relative to the containing block.

```css
article img {
    float: right;
    width: 50%;
}

nav {
    float: left;
    width: 25%;
    border: solid 1px;
}

section {
    margin-left: 25%;
    border: solid 1px;
}

.clearfix {
    overflow: auto;
}
```
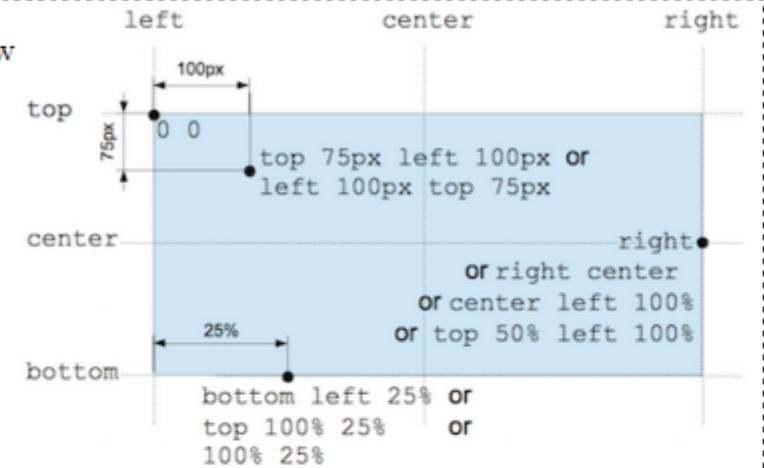
```html
<article class="clearfix dashed">
  <img src="css-positioning.png" alt="an image">
  <p>
    You could even use <code>min-width</code> and <code>max-width</code> to
  </p>
</article>

<nav>
  <ul>
    <li> <a href="#">Home</a>       </li>
    <li> <a href="#">Taco Menu</a>  </li>
    <li> <a href="#">Draft List</a> </li>
    <li> <a href="#">Hours</a>      </li>
    <li> <a href="#">Directions</a> </li>
    <li><a href="#">Contact</a>     </li>
  </ul>
</nav>
<section>
  <p>
    This example works just like the last one. Notice we put a clearfix on the container. It's not needed in this example, but it would be if the nav was longer than the non-floated content.
  </p>
</section>
<section>
  <p>
    Lorem ipsum dolor sit amet, consectetur adipiscing elit. Phasellus imperdiet, nulla et dictum interdum, nisi lorem egestas odio, vitae scelerisque enim ligula venenatis dolor. Maecenas nisl est, ultrices nec congue eget, auctor vitae massa. Fusce luctus vestibulum augue ut aliquet. Mauris ante ligula, facilisis sed ornare eu, lobortis in odio. Praesent convallis urna a lacus interdum ut hendrerit risus congue. Nunc sagittis dictum nisi, sed ullamcorper ipsum dignissim ac. In at libero sed nunc venenatis imperdiet sed ornare turpis. Donec vitae dui eget tellus gravida venenatis. Integer fringilla congue eros non fermentum. Sed dapibus pulvinar nibh tempor porta. Cras ac leo purus. Mauris quis diam velit.
  </p>
</section>
```
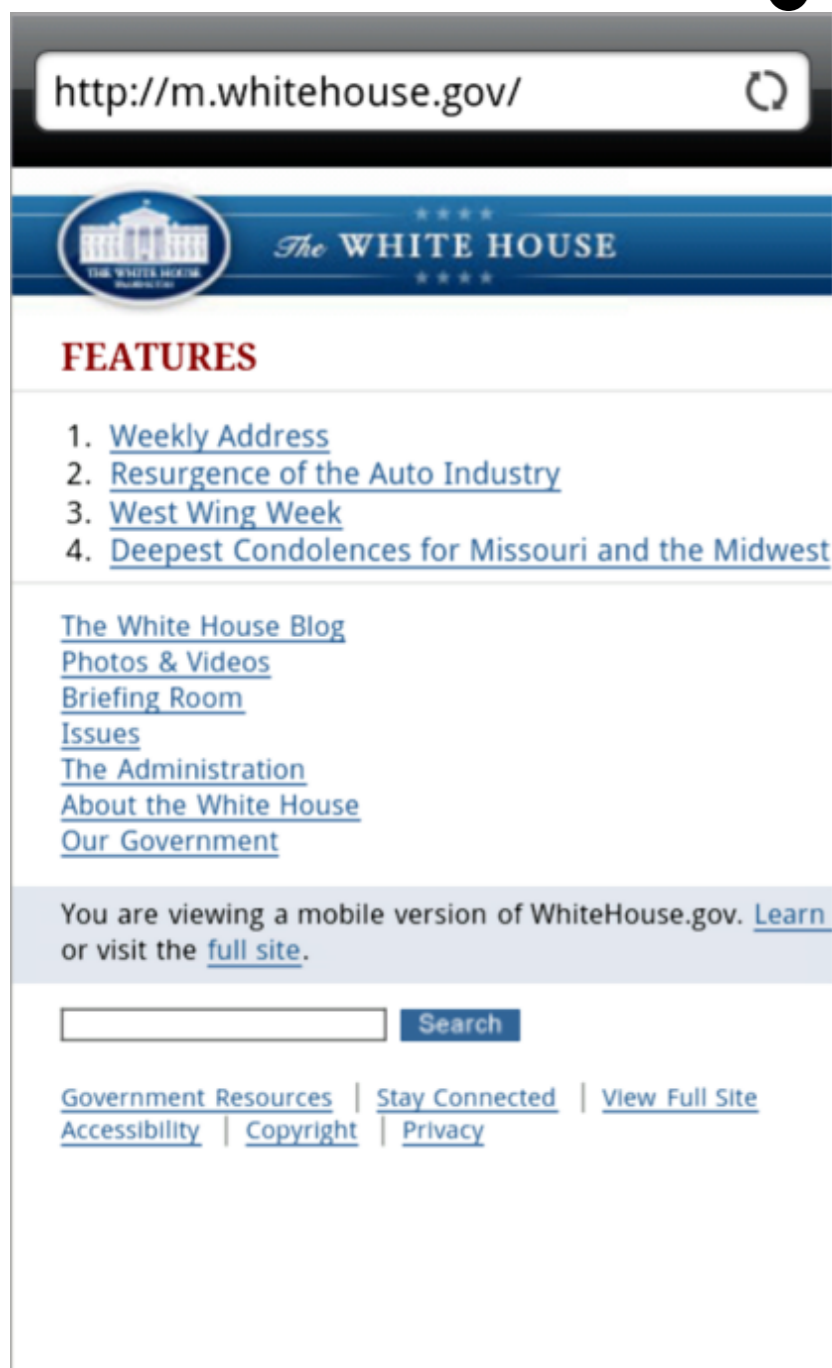
13. percent width

You could even use min-width and max-width to limit how big or small the image can get!

left   center   right

top

100px

75px

0 0

top 75px left 100px or
left 100px top 75px

center

right

or right center
or center left 100%
or top 50% left 100%

25%

bottom

bottom left 25% or
top 100% 25%   or
100% 25%

- Home
- Taco Menu
- Draft List
- Hours
- Directions
- Contact

This example works just like the last one. Notice we put a clearfix on the container. It's not needed in this example, but it would be if the nav was longer than the non-floated content.

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Phasellus imperdiet, nulla et dictum interdum, nisi lorem egestas odio, vitae scelerisque enim ligula venenatis dolor. Maecenas nisl est, ultrices nec congue eget, auctor vitae massa. Fusce luctus vestibulum augue ut aliquet. Mauris ante ligula, facilisis sed ornare eu, lobortis in odio. Praesent convallis urna a lacus interdum ut hendrerit risus congue. Nunc sagittis dictum nisi, sed ullamcorper ipsum dignissim ac. In at libero sed nunc venenatis imperdiet sed ornare turpis. Donec vitae dui eget tellus gravida venenatis. Integer fringilla congue eros non fermentum. Sed dapibus pulvinar nibh tempor porta. Cras ac leo purus. Mauris quis diam velit.
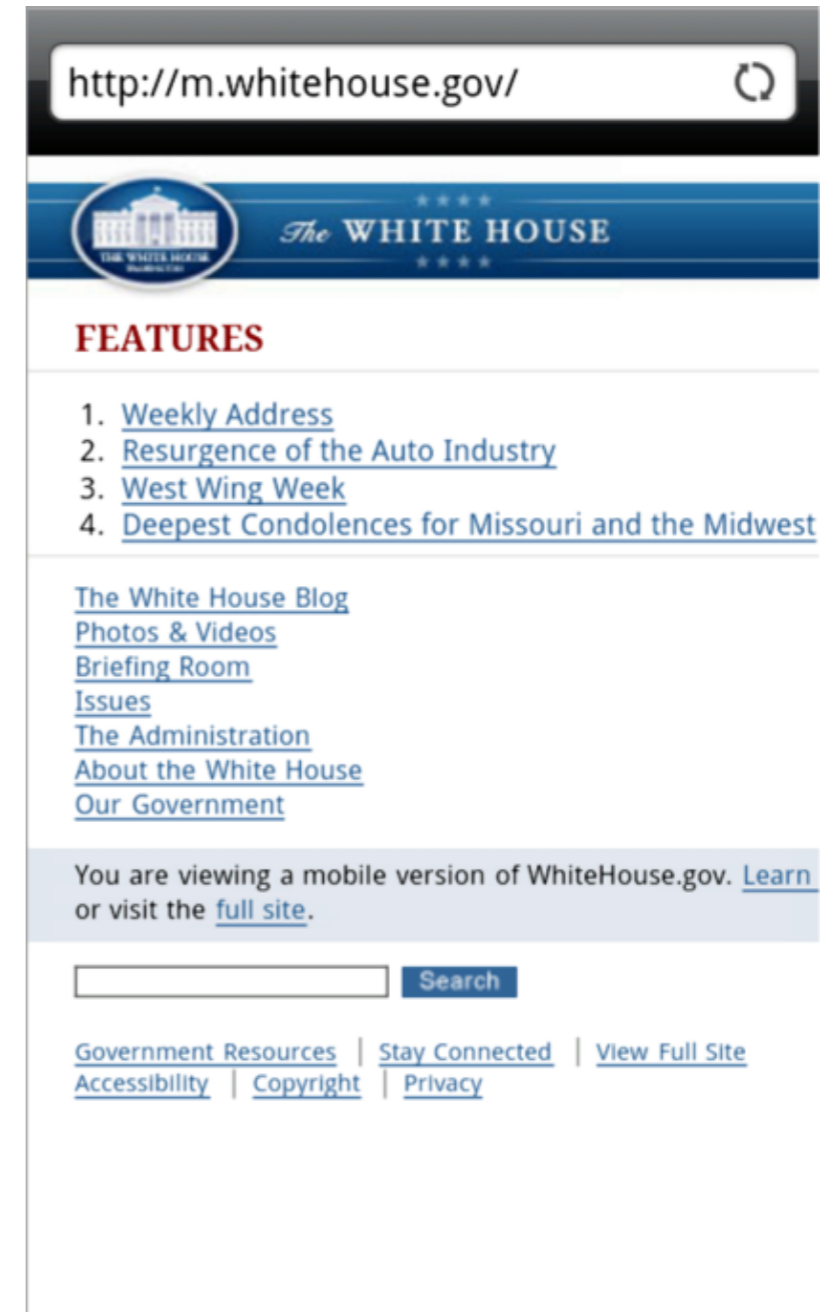
Mobile Web Design Best Practices

- **Three Approaches to Mobile Web:**

  ◦ Create a separate website hosted within your current domain targeted for mobile users e.g. m.itworld.com

  ◦ Use CSS to configure your current website for display on both mobile and desktop devices.

  ◦ Develop a mobile app which resides on the device

Mobile Web Limitations

- Small Screen Size

- Low bandwidth

- Poor connectivity

- No hover states

- Slow error prone typing

- Less context

- Inaccurate clicks

- Slow hardware

- Awkward controls

- Limited processor and memory

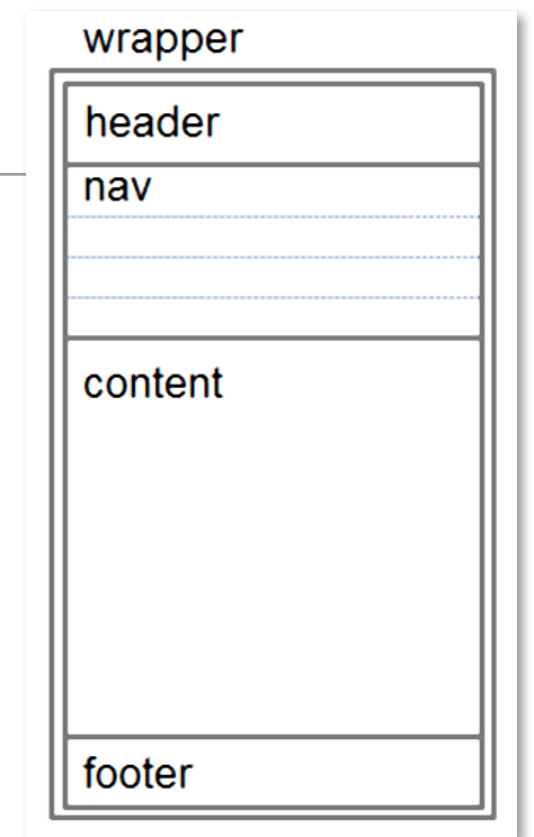- http://www.whitehouse.gov

- http://m.whitehouse.gov

Approaches to mobile web

- Make web sites that work well on a variety of screens (presentation)

- Adjust your content for mobile users (content)

- Give your users a smooth experience even on a slow connection (performance)

Design Techniques for Mobile Web

- Single column design

- Avoid floats, tables, frames

- Descriptive page title

- Descriptive heading tags

- Optimize images

- Descriptive alt text for images

- Eliminate unneeded images

- Navigation in lists

- Em or percentage font size units

- Common font typefaces

- Good contrast between text and background colors

- Provide "Skip to Content" hyperlink

- Provide "Back to Top" hyperlink

Design Techniques for Mobile Web

- Set most padding and margin to 0.

- Background images may have to be adjusted so they have no-repeat and are positioned in the right top of the screen.

- Logos may also need modifying.

Media queries

- Responsive design is the strategy of making a site that "responds" to the browser and device that it is being shown on so it is awesome no matter what.

- CSS3 media queries are the most powerful tool for doing this. Allow you to gather data about the site visitor and use it to conditionally apply CSS styles.

- A simple technique would be to consider the min-width media feature, which allows us to apply specific CSS styles if the browser window drops below a particular width that we can specify. Target these widths:

  - 320px, 480px, 600px, 768px, 900px, 1200px

- These are sometimes called 'breakpoints'

Media Queries

- Determines the capability of the mobile device, such as screen resolution

- Directs the browser to styles configured specifically for those capabilities

- Example:

```
<link href="lighthousemobile.css" rel="stylesheet"
    media="only screen and (max-device-width: 480px)">
```

media queries

```
@media screen and (min-width:600px) {
  nav {
    float: left;
    width: 25%;
    border: solid 1px;
  }
  section {
    margin-left: 25%;
    border: solid 1px;
  }
}

@media screen and (max-width:599px) {
  nav li {
    display: inline;
  }
}
```

```
<h1> 14. media queries </h1>
<div class="clearfix">
<nav>
  <ul>
    <li> <a href="#">Home</a>         </li>
    <li> <a href="#">Taco Menu</a>  </li>
    <li> <a href="#">Draft List</a> </li>
    <li> <a href="#">Hours</a>        </li>
    <li> <a href="#">Directions</a> </li>
    <li><a href="#">Contact</a>      </li>
  </ul>
</nav>
<section>
  <p>
    This example works just like the last one. Notice we put a clearfix on the
    container. It's not needed in this example, but it would be if the nav was
    longer than the non-floated content.
  </p>
</section>
<section>
  <p>
    Lorem ipsum dolor sit amet, consectetur adipiscing elit. Phasellus
    imperdiet, nulla et dictu
    scelerisque enim ligula ve
    congue eget, auctor vitae
    Mauris ante ligula, facil:
    convallis urna a lacus int
    dictum nisi, sed ullamcorp
    venenatis imperdiet sed or
    venenatis. Integer fringi:
    pulvinar nibh tempor port
  </p>
</section>
</div>
```

14. media queries

Home Taco Menu Draft List Hours Directions Contact

This example works just like the last one. Notice we put a clearfix on the container. It's not needed in this example, but it would be if the nav was longer than the non-floated content.

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Phasellus imperdiet, nulla et dictum interdum, nisi lorem egestas odio, vitae scelerisque enim ligula venenatis dolor. Maecenas nisl est, ultrices nec congue eget, auctor vitae massa. Fusce luctus vestibulum augue ut aliquet. Mauris ante ligula, facilisis sed ornare eu, lobortis in odio. Praesent convallis urna a lacus interdum ut hendrerit risus congue. Nunc sagittis dictum nisi, sed ullamcorper ipsum dignissim ac. In at libero sed nunc venenatis imperdiet sed ornare turpis. Donec vitae dui eget tellus gravida venenatis. Integer fringilla congue eros non fermentum. Sed dapibus pulvinar nibh tempor porta. Cras ac leo purus. Mauris quis diam velit.
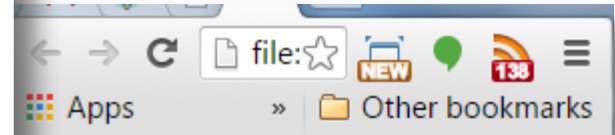
14. media queries

- Home
- Taco Menu
- Draft List
- Hours
- Directions
- Contact

This example works just like the last one. Notice we put a clearfix on the container. It's not needed in this example, but it would be if the nav was longer than the non-floated content.

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Phasellus imperdiet, nulla et dictum interdum, nisi lorem egestas odio, vitae scelerisque enim ligula venenatis dolor. Maecenas nisl est, ultrices nec congue eget, auctor vitae massa. Fusce luctus vestibulum augue ut aliquet. Mauris ante ligula, facilisis sed ornare eu, lobortis in odio. Praesent convallis urna a lacus interdum ut hendrerit risus congue. Nunc sagittis dictum nisi, sed ullamcorper ipsum dignissim ac. In at libero sed nunc venenatis imperdiet sed ornare turpis. Donec vitae dui eget tellus gravida venenatis. Integer fringilla congue eros non fermentum. Sed dapibus pulvinar nibh tempor porta. Cras ac leo purus. Mauris quis diam velit.

Viewport Meta Tag

- Default action for most mobile devices is to zoom out and scale the web page

- Viewport Meta Tag

- Configures width and initial scale of browser viewport

<meta name="viewport" content= "width=device-width, initial-scale=1.0">

http://www.quirksmode.org/mobile/viewports.html

inline-block

```
.box {
    display: inline-block;
    width: 200px;
    height: 100px;
    margin: 1em;
    border: solid 1px;
}

.after-box {
    border: solid 1px;
}
```

Grid of boxes that fills the browser width and wraps nicely. You can set a width and height.

```
<div class="box">
    <p>
        I'm floating!
    </p>
</div>
<div class="box">
    <p>
        I'm floating!
    </p>
</div>
    <div class="box">
    <p>
        I'm floating!
    </p>
</div>
    <div class="box">
    <p>
        I'm floating!
    </p>
</div>
    <div class="box">
    <p>
        I'm floating!
    </p>
</div>
</div>
    <div class="box">
    <p>
        I'm floating!
    </p>
</div>
</div>
    <div class="box">
    <p>
        I'm floating!
    </p>
</div>
</div>
    <div class="box">
    <p>
        I'm floating!
    </p>
</div>
</div>
    <div class="box">
    <p>
        I'm floating!
    </p>
</div>
<section class="after-box">
    <p>
        I'm using clear so I don't float next to the above boxes.
    </p>
</section>
</body>
</html>
```

15. inline block

I'm floating!	I'm floating!	I'm floating!
I'm floating!	I'm floating!	I'm floating!
I'm floating!	I'm floating!	I'm floating!

I'm using clear so I don't float next to the above boxes.

inline-block layout

- You can use inline-block for layouts. Remember to set the width and vertical-align to top.

```html
<div class="dashed">
  <nav class="dashed">
    <ul>
      <li> <a href="#">Home</a>         </li>
      <li> <a href="#">Taco Menu</a>    </li>
      <li> <a href="#">Draft List</a>   </li>
      <li> <a href="#">Hours</a>        </li>
      <li> <a href="#">Directions</a>  </li>
      <li><a href="#">Contact</a>       </li>
    </ul>
  </nav>
  <div class="column dashed">
  <section>
    <p>
    This example works just like the last one. Notice we put a clearfix on the
    container. It's not needed in this example, but it would be if the nav was
    longer than the non-floated content.
    </p>
  </section>
  <section>
    <p>
    Lorem ipsum dolor sit amet, consectetur adipiscing elit. Phasellus
    imperdiet, nulla et dictum interdum, nisi lorem egestas odio, vitae
    scelerisque enim ligula venenatis dolor. Maecenas nisl est, ultrices nec
    congue eget, auctor vitae massa. Fusce luctus vestibulum augue ut aliquet.
    Mauris ante ligula, facilisis sed ornare eu, lobortis in odio. Praesent
    convallis urna a lacus interdum ut hendrerit risus congue. Nunc sagittis
    dictum nisi, sed ullamcorper ipsum dignissim ac. In at libero sed nunc
    venenatis imperdiet sed ornare turpis. Donec vitae dui eget tellus gravida
    venenatis. Integer fringilla congue eros non fermentum. Sed dapibus
    pulvinar nibh tempor porta. Cras ac leo purus. Mauris quis diam velit.
    </p>
  </section>
  </div>
</div>
```

```css
nav{
    display: inline-block;
    vertical-align: top;
    width: 25%;
}
.column{
    display: inline-block;
    vertical-align: top;
    width: 74%;
}
```

column

- css columns are very new, so you need to use the prefixes and it won't work in IE9 or Opera Mini

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Phasellus imperdiet, nulla et dictum interdum, nisi lorem egestas odio, vitae scelerisque enim ligula venenatis dolor. Maecenas nisl est, ultrices nec congue eget, auctor vitae massa. Fusce luctus

vestibulum augue ut aliquet. Mauris ante ligula, facilisis sed ornare eu, lobortis in odio. Praesent convallis urna a lacus interdum ut hendrerit risus congue. Nunc sagittis dictum nisi, sed ullamcorper ipsum dignissim ac. In at libero sed nunc venenatis

imperdiet sed ornare turpis. Donec vitae dui eget tellus gravida venenatis. Integer fringilla congue eros non fermentum. Sed dapibus pulvinar nibh tempor porta. Cras ac leo purus. Mauris quis diam velit.

```html
<body>
  <section>
    <p class="three-column">
      Lorem ipsum dolor sit amet, consectetur adipiscing elit. Phasellus
      imperdiet, nulla et dictum interdum, nisi lorem egestas odio, vitae
      scelerisque enim ligula venenatis dolor. Maecenas nisl est, ultrices nec
      congue eget, auctor vitae massa. Fusce luctus vestibulum augue ut aliquet.
      Mauris ante ligula, facilisis sed ornare eu, lobortis in odio. Praesent
      convallis urna a lacus interdum ut hendrerit risus congue. Nunc sagittis
      dictum nisi, sed ullamcorper ipsum dignissim ac. In at libero sed nunc
      venenatis imperdiet sed ornare turpis. Donec vitae dui eget tellus gravida
      venenatis. Integer fringilla congue eros non fermentum. Sed dapibus
      pulvinar nibh tempor porta. Cras ac leo purus. Mauris quis diam velit.
    </p>
  </section>
</body>
```

```css
.three-column {
  padding: 1em;
  -moz-column-count: 3;
  -moz-column-gap: 1em;
  -webkit-column-count: 3;
  -webkit-column-gap: 1em;
  column-count: 3;
  column-gap: 1em;
}
```

flexbox

- The new flexbox layout mode is poised to redefine how we do layouts in CSS.

- The following examples only work with some browsers that use the latest version of the standard.

- https://css-tricks.com/snippets/css/a-guide-to-flexbox/

flexbox

- simple layout.

```html
<div class="container dashed">
  <nav>
    <ul>
      <li> <a href="#">Home</a>        </li>
      <li> <a href="#">Taco Menu</a>   </li>
      <li> <a href="#">Draft List</a> </li>
      <li> <a href="#">Hours</a>       </li>
      <li> <a href="#">Directions</a> </li>
      <li><a href="#">Contact</a>      </li>
    </ul>
  </nav>
  <div class="flex-column dashed">
  <section>
    <p>Flexbox is so easy!</p>
  </section>
  <section>
    <p>
      Lorem ipsum dolor sit amet, consectetur adipiscing elit. Phasellus imperdiet, nulla et dictum interdum, nisi
      lorem egestas odio, vitae scelerisque enim ligula venenatis dolor. Maecenas nisl est, ultrices nec congue
      eget, auctor vitae massa. Fusce luctus vestibulum augue ut aliquet. Mauris ante ligula, facilisis sed ornare
      eu, lobortis in odio. Praesent convallis urna a lacus interdum ut hendrerit risus congue. Nunc sagittis
      dictum nisi, sed ullamcorper ipsum dignissim ac. In at libero sed nunc venenatis imperdiet sed ornare turpis.
      Donec vitae dui eget tellus gravida venenatis. Integer fringilla congue eros non fermentum. Sed dapibus
      pulvinar nibh tempor porta. Cras ac leo purus. Mauris quis diam velit.
    </p>
  </section>
  </div>
</div>
```

- Home
- Taco Menu
- Draft List
- Hours
- Directions
- Contact

Flexbox is so easy!

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Phasellus imperdiet, nulla et dictum interdum, nisi lorem egestas odio, vitae scelerisque enim ligula venenatis dolor. Maecenas nisl est, ultrices nec congue eget, auctor vitae massa. Fusce luctus vestibulum augue ut aliquet. Mauris ante ligula, facilisis sed ornare eu, lobortis in odio. Praesent convallis urna a lacus interdum ut hendrerit risus congue. Nunc sagittis dictum nisi, sed ullamcorper ipsum dignissim ac. In at libero sed nunc venenatis imperdiet sed ornare turpis. Donec vitae dui eget tellus gravida venenatis. Integer fringilla congue eros non fermentum. Sed dapibus pulvinar nibh tempor porta. Cras ac leo purus. Mauris quis diam velit.

```css
.container {
    display: -webkit-flex;
    display: flex;
}

nav {
    width: 200px;
}

.flex-column {
    -webkit-flex: 1;
            flex: 1;
}
```

flexbox

```css
.container {
    display: -webkit-flex;
    display: flex;
}
.initial {
        width: 200px;
    min-width: 100px;
}

.none {
  -webkit-flex: none;
          flex: none;
  width: 200px;
}
.flex1 {
  -webkit-flex: 1;
          flex: 1;
}
.flex2 {
  -webkit-flex: 2;
          flex: 2;
}
```

```html
<body>
<div class="container dashed">
  <div class="initial dashed">
    I will be 200px when there is room, and I will shrink down to 100px if there is not room, but no smaller.
  </div>
  <div class="none dashed">
    I will always be 200px, no matter what.
  </div>
  <div class="flex1 dashed">
    I will fill up 1/3 of the remaining width.
  </div>
  <div class="flex2 dashed">
    I will fill up 2/3 of the remaining width.
  </div>
</div>
</body>
```

• more complex layout, one flex container with several div's each with its own properties set for sizing.

| I will be 200px when there is room, and I will shrink down to 100px if there is not room, but no smaller. | I will always be 200px, no matter what. | I will fill up 1/3 of the remaining width. | I will fill up 2/3 of the remaining width. |

flexbox

```
.vertical-container {
    height: 300px;
    display: -webkit-flex;
    display: flex;
    -webkit-align-items: center;
    align-items: center;
    -webkit-justify-content: center;
    justify-content: center;
}
```

Finally it's easy to vertically center something in CSS!

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <title></title>
  <link type="text/css" rel="stylesheet" href="style.css" media="screen"/>
</head>
<body>
<div class="vertical-container dashed">
  <div>
   Finally it's easy to vertically center something in CSS!
  </div>
</div>
</body>
</html>
```

- Centering using Flexbox makes the process a lot easier and exact than alternative methods.

Learn CSS Layout

Table of Contents

http://learnlayout.com
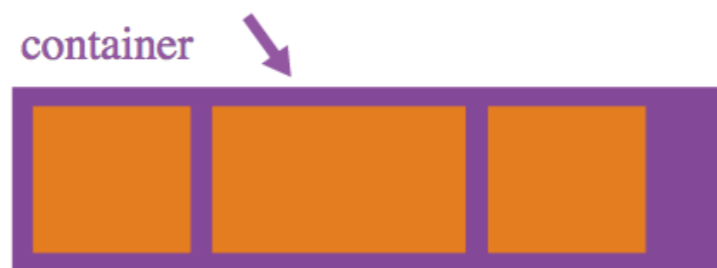
Code Snippets » CSS »

A Complete Guide to Flexbox

BY **CHRIS COYIER** LAST UPDATED ON OCTOBER 12, 2016

▶ **Background**

▶ **Basics & Terminology**

container

items

Properties for the Parent

(flex container)

Properties for the Children

(flex items)

Code Snippets » CSS »

A Complete Guide to Grid

BY CHRIS HOUSE LAST UPDATED ON APRIL 13, 2016

This article was ported over from Chris House's guide, by Chris himself, who is keeping both up-to-date.

▶ **Introduction**

▶ **Basics and Browser Support**

▶ **Important Terminology**

▶ **Grid Properties Table of Contents**