



PART-TIME

BOOTCAMP

INTRO



KEYS TO SUCCESS

CIRCUS
DETROIT

GRAND
CIRCUS
DETROIT

CIRCUS
DETROIT

GRAND
CIRCUS
DETROIT

GRAND
CIRCUS
DETROIT

GRAND
CIRCUS
DETROIT

KEYS TO SUCCESS

As of right now, your primary job description is:

FIGURE IT OUT



KEYS TO SUCCESS

Developers are essentially well-paid problem solvers.
If you come across something you're not sure about,
you need to make an effort to solve the issue
yourself.

KEYS TO SUCCESS

some 'rules' for when you get stuck:

1. First, ask Google.
2. Second, ask a fellow bootcamper.
3. Third, ask another fellow bootcamper.
4. If you're still stumped, ask me or your TA.

KEYS TO SUCCESS

The rule about 'the rules'

That said, don't be a hero and spin your wheels for hours without progress. It profits no one. If you're genuinely stuck, ask for help and keep asking until you get it.



KEYS TO SUCCESS

Most importantly

PRACTICE!!

We won't have any real 'homework' for this class, but any extra time you spend working on this stuff will only deepen your understanding of the material. If you finish an assignment early, look for ways to go a step further or do some extra work on the final project.

GETTING STARTED

Note that some of this should already be done. This is mostly a checklist of things we will need for smooth(ish) sailing in this course.

INSTALL GOOGLE CHROME

(If you don't already have it)

Technically, any modern browser will work but we will be using some extensions that are specific to Chrome for this course, so make sure you have Chrome on your machine.

INSTALL A TEXT EDITOR

A couple of recommendations:

- Sublime Text
- Atom.io*
- Whatever else you're comfortable with

* I'm most comfortable with Atom and I will be of the most help to you if you use it. Anything else and I won't be as quick.



SLACK CHANNEL

You've probably already been invited to the Slack channel. <http://www.grand-circus.slack.com>

If you haven't, let me know and we will get you added.



PRIVATE GROUP

`ptbc_march`

We have a private group within the Slack channel specifically for this bootcamp. It will be a place for communication and collaboration between classes. It will also be where I post slides and extra materials.

GITHUB

github.com

Make yourself a GitHub profile.

It will be a little while before we need it, but go ahead and take care of this now while we're setting up everything.

GIT

To download git, visit [git's website](#), choose your platform, and install git.

Like GitHub, it'll be a little while before we need this, but better to get all of this stuff out of the way at once.

Every website on the internet uses HTML & CSS.





...most of them use JavaScript as well, in one form or another.

GRAND
CIRCUS
DETROIT

(right click in the browser and select **Inspect**
Element)

GRAND
CIRCUS
DETROIT

GRAND
CIRCUS
DETROIT

GRAND
CIRCUS
DETROIT

GRAND
CIRCUS

GRAND
CIRCUS
DETROIT

GRAND
CIRCUS

The screenshot shows a browser's developer tools with the "Elements" tab selected. The left pane displays the DOM tree, and the right pane shows the CSS styles for the selected element. The CSS panel highlights the `reveal.css` file, which contains the reveal.js configuration options.

```
<!DOCTYPE html>
<html lang="en">
  <head></head>
  <body style="transition: -webkit-transform 0.8s ease; -webkit-transition: -webkit-transform 0.8s ease;">
    <!-- Any section element inside this container is displayed as a slide -->
    <div class="reveal convex center slide" role="application" data-transition-speed="default" data-background-transition="fade">
      <!-- END SLIDES DIV -->
      <div class="backgrounds"></div>
      <div class="progress" style="display: block;"></div>
      <aside class="controls" style="display: block;"></aside>
      <div class="slide-number"></div>
      <div class="pause-overlay"></div>
      <div id="aria-status-div" aria-live="polite" aria-atomic="true" style="position: absolute; height: 1px; width: 1px; overflow: hidden; clip: rect(1px 1px 1px 1px);">DEMO!
        (right click in the browser and select Inspect Element)
      </div>
      <::after
      </div>
      <script src="lib/js/head.min.js"></script>
      <script src="js/reveal.js"></script>
      <script>
        // Full list of configuration options available at:
        // https://github.com/hakimel/reveal.js#configuration
        Reveal.initialize({
          controls: true,
          progress: true,
          history: true,
          center: true,
          transition: 'slide', // none/fade/slide/convex/concave/zoom
          // Optional reveal.js plugins
          dependencies: [
            { src: 'lib/js/classList.js', condition: function() { return !document.body.classList; } },
            { src: 'plugin/markdown/markdown.js', condition: function() { return !!document.querySelector( '[data-markdown]' ); } },
            { src: 'plugin/markdown/markdown.js', condition: function() { return !!document.querySelector( '[data-markdown]' ); } },
            { src: 'plugin/highlight/highlight.js', condition: function() { return !!document.querySelector( '[pre code]' ); }, callback: function() { hljs.initHighlightingOnLoad(); } },
            { src: 'plugin/zoom-js/zoom.js', async: true, condition: function() { return !!document.querySelector( '[data-zoom]' ); } },
            { src: 'plugin/notes/notes.js', async: true }
          ]
        });

        </script>
        <script type="text/javascript" src="plugin/markdown/markdown.js"></script>
        <script type="text/javascript" src="plugin/markdown/markdown.js"></script>
        <script type="text/javascript" src="plugin/zoom-js/zoom.js"></script>
        <script type="text/javascript" src="plugin/notes/notes.js"></script>
      </body>
    </html>

```

Styles Computed Event Listeners »

Inherited from body

body { reveal.css:30

position: relative;
line-height: 1;
background-color: #fff;
color: #000;

Inherited from html

html, body, .reveal div, .reveal.css:11

.reveal span, .reveal applet, .reveal object, .reveal iframe, .reveal h1, .reveal h2, .reveal h3, .reveal h4, .reveal h5, .reveal h6, .reveal p, .reveal blockquote, .reveal pre, .reveal q, .reveal abbr, .reveal acronym, .reveal address, .reveal b, .reveal cite, .reveal code, .reveal del, .reveal dfn, .reveal em, .reveal img, .reveal ins, .reveal kbd, .reveal q, .reveal s, .reveal samp, .reveal small, .reveal strike, .reveal strong, .reveal sub, .reveal sup, .reveal tt, .reveal var, .reveal b, .reveal u, .reveal center, .reveal dl, .reveal dt, .reveal dd, .reveal ol, .reveal ul, .reveal li, .reveal fieldset, .reveal form, .reveal label, .reveal legend, .reveal table, .reveal caption, .reveal tbody, .reveal tfoot, .reveal thead, .reveal tr, .reveal th, .reveal td, .reveal article, .reveal aside, .reveal canvas, .reveal details, .reveal embed, .reveal figcaption, .reveal figure, .reveal footer, .reveal header, .reveal hgroup, .reveal menu, .reveal nav, .reveal output, .reveal ruby, .reveal section, .reveal summary, .reveal time, .reveal mark, .reveal audio, video { margin: 0;
padding: 0;
border: 0;
font-size: 100%;
font: inherit;
vertical-align: baseline;

Find in Styles

TERMINOLOGY

TERMINOLOGY

Some key terms or phrases that are used as a matter of course in the software industry. You may know some or all of these, or you may have heard the terms but be unclear about their actual meaning. They are common jargon among developers

WEB DEVELOPMENT

Web development is a broad term for the work involved in developing a web site for the Internet. In industry parlance, 'web development' usually refers to the more code-related tasks such as programming JavaScript, coding HTML and CSS. It can even extend to tasks related to the back end server infrastructure such as creating web services and handling business logic for a company or product.

WEB DESIGN

The process of planning & structuring a website; specifically, the visual aspects and assets for the site.

Recently, this job description has also begun to include interaction design. That is, designing the user experience (UX), information architecture, and the flow of the application or site.

WEB SITE

A largely informational web page. While they may include dynamic elements and react to user inputs.

The general purpose of a web site is to provide information about a person, business, product, or service.

WEB APPLICATION

A more recent term to indicate a web site whose sole purpose is not just informational, but rather functional. Web applications have become robust enough to do everything from our taxes, manage our personal calendars, or even do standard desktop publishing tasks.

FRONT END

The visible and interactive parts of a website or application.

BACK END

The 'invisible' or inner functionality of a website or application. Examples include costly calculations, interacting with a database or making use of web service end points. While we're at it...

WEB SERVICE

A software function provided at a network address over the Web. The W3C defines a Web service generally as:

"A software system designed to support interoperable machine-to-machine interaction over a network."

DATABASE

A software system for storing data long term. This is also sometimes referred to as a 'persistance layer'.

VERSION CONTROL

A software tool for managing changes to a set of files, website, application or any collection of files and for reconciling the differences between those files when conflicts emerge.

APPLICATION PROGRAMMING INTERFACE (API)

An API is a set of routines, protocols, and tools for building software applications. An API expresses a software component in terms of its operations, inputs, outputs, and underlying types. An API defines functionalities that are independent of their respective implementations, *which allows definitions and implementations to vary without compromising each other.*



CSS POSITIONING



BLOCK VS. INLINE ELEMENTS



BLOCK ELEMENTS

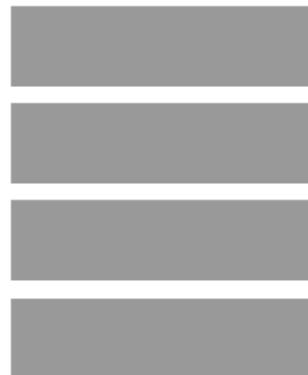
Block elements each appear on a new line of a web page, like paragraphs. Spatially, what is happening is that the block element takes up all of the horizontal space it can. It stretches to fill all the space to the left and right of the element within its parent container.

INLINE ELEMENTS

Inline elements are rendered without starting a new line. They appear side by side until reaching the edge of its parent container. Then it will start a new line.

BLOCK VS. INLINE ELEMENTS

BLOCK:



INLINE:



BLOCK VS. INLINE ELEMENTS

- *Inline element:* `<a> `
- *Block element:* `<div> <p> <table>`
...and just about everything else.



THE BOX MODEL



THE BOX MODEL

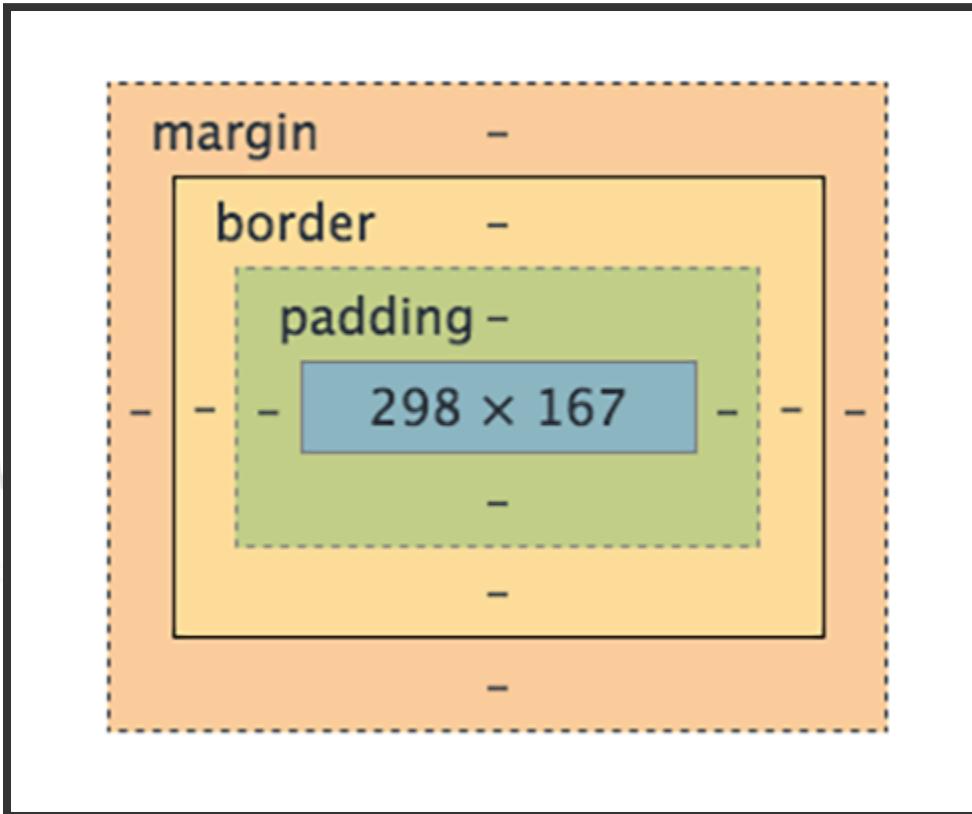
Every HTML element is in a box, regardless of its visible shape.

THE BOX MODEL

The total size of an element is a combination of the following:

- Content
- Padding
- Border
- Margin

THE BOX MODEL



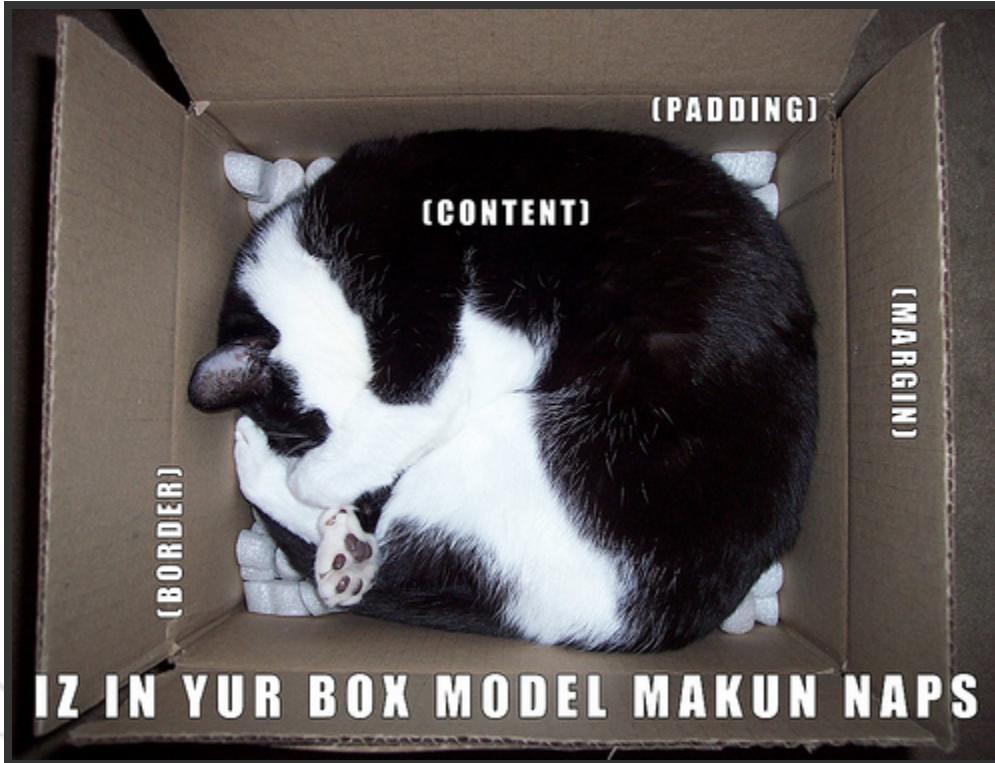
CIRCUS
DETROIT

GRAND
CIRCUS
DETROIT

CIRCUS
DETROIT

GRAND
CIRCUS
DETROIT

THE BOX MODEL



GRAND
CIRCUS
DETROIT

PADDING

PADDING

Padding is the space between the content within an element and the border. Padding can be specified in units of pixels (px), Ems (em), or points (pt) or percentage (%) of the containing element.

PADDING

Padding is a CSS property (just like color or font-family) and is defined using the same rules. The only difference is that there are a few different ways to write them.

PADDING

```
p {  
    padding: 10px; /* 10 px of padding on all sides */  
}  
p {  
    padding: 10px 5px;  
    /* 10 px on top/bottom, 5px on left/right */  
}  
p {  
    padding-top: 5px;  
    padding-right: 10px;  
    padding-bottom: 15px;  
    padding-left: 20px;  
}  
p {  
    padding: 5px 10px 15px 20px;  
    /* top, right, bottom, left */}
```

GRAND
CIRCUS
DETROIT

CIRCUS
DETROIT

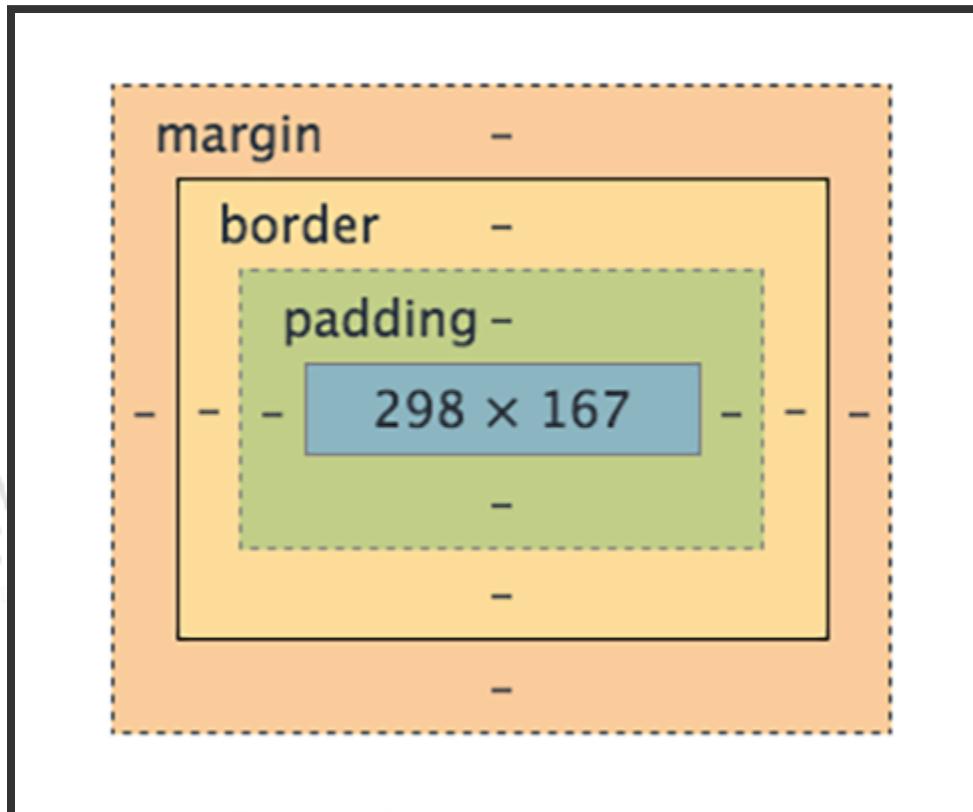
GRAND
CIRCUS

GRAND
CIRCUS
DETROIT

GRAND
CIRCUS

BORDER

THE BOX MODEL





BORDER

Border is the edge around the element. It has elements of thickness, style, and color.

BORDER

Like padding, border is a CSS property and is defined using CSS rules. Also like padding, there are a few different ways to specify a border.

GRAND
CIRCUS
DETROIT

GRAND
CIRCUS
DETROIT

CIRCUS
DETROIT

GRAND
CIRCUS
DETROIT

BORDER

```
p { /* each property specified separately */
  border-width: 10px; /* thickness */
  border-style: solid;
  border-color: blue;
}
p { /* all properties specified in one rule */
  border: 1px dashed black;
}
p { /* Two separate styles on one element */
  border-top: 1px dotted #FF0000;
  border-bottom: 1px groove #FF0;
}
```

BORDER STYLES

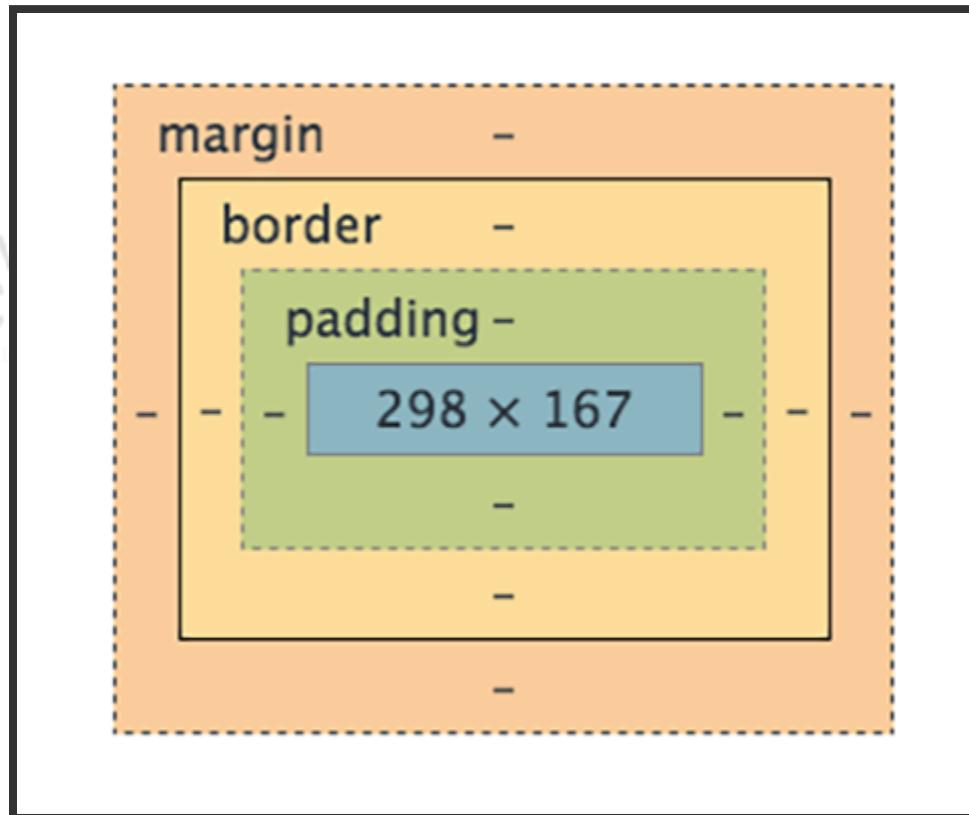
none hidden dotted dashed
solid double groove ridge
inset outset

GRAND
CIRCUS
DETROIT

MARGIN

GRAND
CIRCUS
DETROIT

THE BOX MODEL





MARGIN

Margins are the space outside the element that separates it from other elements. It is 'outside the fence' in terms of its relationship to the border.

MARGIN

Like padding & border is a CSS property and is defined using CSS rules. Much like padding, there are a few different ways to specify it.



MARGIN



```
p {  
margin: 10px; /* 10 pixels of margin on all sides */  
}  
p {  
margin: 10px 5px; /* 10 px on top/bottom, 5px on left/right */  
}  
p {  
margin-top: 5px;  
margin-right: 10px;  
margin-bottom: 15px;  
margin-left: 20px;  
}  
p {  
margin: 5px 10px 15px 20px; /* top, right, bottom, left */  
}
```



AUTO MARGIN

If margin is set to `auto` on an element that has a set width, it will take up as much space as possible. You can use this to center or left or right justify an element. This only works for horizontally centering.

CIRCUS
DETROIT

GRAND
CIRCUS
DETROIT

GRAND
CIRCUS
DETROIT

GRAND
CIRCUS
DETROIT

GRAND
CIRCUS
DETROIT

AUTO MARGIN

```
p {  
margin: 0 auto;  
width: 300px;  
}  
p {  
margin-right: auto;  
margin-left: 10px;  
width: 25%;  
}
```

GRAND
CIRCUS
DETROIT



F



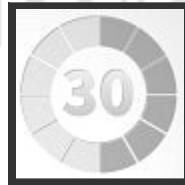
EXERCISE MESSING WITH THE BOX MODEL



F



GRAND
CIRCUS
DETROIT



GRAND
CIRCUS
DETROIT

BOX MODEL SHENANIGANS

- New Project (index.html, main.css)
- Create a basic HTML site
- Add 3 div elements with IDs and put a paragraph tag in each.
- Add some content
- Apply the following CSS properties
- Using the ID selectors, specify padding, border, and margin for each div
- try different border styles
- try setting some different widths and heights just to see how they're affected (if at all)

GRAND
CIRCUS
DETROIT

CSS POSITIONING

GRAND
CIRCUS
DETROIT

GRAND
CIRCUS

GRAND
CIRCUS
DETROIT

GRAND
CIRCUS

CSS POSITIONING

Every HTML element has a property called **position** which dictates how that element flows on a document. This property can be set to a number of different values, each of which behave slightly differently. This gives a web developer greater control over the design and overall look of a page.



STATIC POSITIONING

STATIC POSITIONING

Static positioning is the default for all HTML elements. And it adheres to the previously discussed behavior of block and inline elements. Block elements flow from top to bottom and each element appears on a new line. Inline elements flow from left to right, wrapping as needed.



GRAND
CIRCUS
DETROIT



GRAND
CIRCUS
DETROIT



RELATIVE POSITIONING

RELATIVE POSITIONING

Relative positioned elements appear in the normal flow of the document but can be offset by using the top, bottom, left and right properties.

GRAND
CIRCUS
DETROIT

RELATIVE POSITIONING

Demo

GRAND
CIRCUS
DETROIT

ABSOLUTE POSITIONING

ABSOLUTE POSITIONING

Absolutely positioned are removed from the normal flow of the document. They are not affected by and do not affect other elements in the flow, regardless of their position in the document.

ABSOLUTE POSITIONING

Absolute positioned elements appear relative to their parent container elements, using the top/bottom/right/left properties for offsets.

ABSOLUTE POSITIONING

You can think of absolutely positioned elements as being stuck onto a page with a magnet. Once placed, they don't move.



ABSOLUTE POSITIONING

Demo

FIXED POSITIONING

Fixed positioned elements are very similar to absolutely positioned elements. Fixed positioned elements are also removed from the normal flow of the document. They are not affected by and do not affect other elements in the flow, regardless of their position in the document.



FIXED POSITIONING

Fixed positioned elements appear relative to the view port (the browser window), using the top/bottom/right/left properties for offsets.



FIXED POSITIONING

Fixed elements do not move when you scroll up or down the page. They seem fixed to the window.

GRAND
CIRCUS
DETROIT

FIXED POSITIONING

Demo

GRAND
CIRCUS
DETROIT

FLOAT

GRAND
CIRCUS
DETROIT



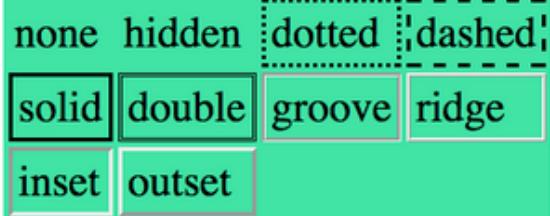
FLOAT

"Floating" takes an element in the normal flow and pushes it as far to the left or right of its parent element as possible. When an element is floated, other elements will wrap around it

FLOAT

- Always specify a width when floating an element
- Specify whether an element should float left or right
- If two or more elements are floated, they will 'stack up' on whichever side they are floated to.

FLOAT



Lorem ipsum dolor sit amet, consectetur adipisicing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat. Duis aute irure dolor in reprehenderit in voluptate velit esse cillum dolore eu fugiat nulla pariatur. Excepteur sint occaecat cupidatat

non proident, sunt in culpa qui officia deserunt mollit anim id est laborum. Lorem ipsum dolor sit amet, consectetur adipisicing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat. Duis aute irure dolor in reprehenderit in voluptate velit esse cillum dolore eu fugiat nulla pariatur. Excepteur sint occaecat cupidatat non proident, sunt in culpa qui officia deserunt mollit anim id est laborum. Lorem ipsum dolor sit amet, consectetur adipisicing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat. Duis aute irure dolor in reprehenderit in voluptate velit esse cillum dolore eu fugiat nulla pariatur. Excepteur sint occaecat cupidatat non proident, sunt in culpa qui officia deserunt mollit anim id est laborum.



CLEAR

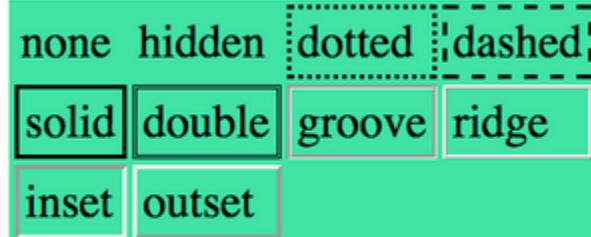




CLEAR

Clear specifies on which side of an element other elements cannot appear. You can clear left, right, or both.

FLOAT



Not cleared Item
Not cleared Item
Not cleared Item

Z-INDEX

When elements are moved out of the normal flow of content, they can overlap. Z-index can be used to define the order of overlapping elements. The element with the highest z-index goes on top.

GRAND
CIRCUS
DETROIT

Z-INDEX

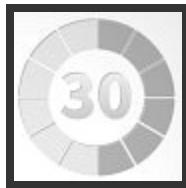
Demo



EXERCISE

CSS POSITIONING





CSS POSITIONING EXERCISE

- New Project (index.html, main.css)
- Create a basic HTML site
- Add 4 divs and add some content to each
- Float 2 of the divs so that they rest side-by-side
- In another of the divs, use relative positioning to offset some content inside the div
- In the last div, attach the content to the viewport using fixed positioning. Add enough height to the body tag to scroll up and down

GRAND
CIRCUS
DETROIT

CIRCUS
DETROIT

GRAND
CIRCUS
DETROIT

GRAND
CIRCUS
DETROIT

GRAND
CIRCUS
DETROIT

GRAND
CIRCUS
DETROIT

GR
CIR

GRAND
CIRCUS
DETROIT

LAB 1

IMAGE GALLERY

GRAND
CIRCUS
DETROIT

GRAND
CIRCUS
DETROIT

GRAND
CIRCUS
DETROIT

GRAND
CIRCUS
DETROIT

GRAND
CIRCUS

GRAND
CIRCUS
DETROIT

GRAND
CIRCUS
DETROIT

GRAND
CIRCUS

CSS LAB

- Make a website!
- Your website should include 3 rows of images, 3 images per row
- Each image should say something about you (hobby, vacation spot, favorite book, etc.)
- Each image should have an explanatory caption
- Each row of images should be contained in a div with padding and margin
- Each image should have a border, padding, and margin
- Your page should include a header and footer
- Use at least one absolutely positioned element
- Use at least one fixed positioned element

CIRCUS
DETROIT

GRAND
CIRCUS
DETROIT

CIRCUS
DETROIT

GRAND
CIRCUS
DETROIT

NEW
CONTENT
AHEAD!



HTML5

HTML5

HTML5 is the most recent spec from the W3C that expands the language of HTML and adds a bunch of new features.

HTML5

We don't have time to go into all the details of all of HTML5's new features. Here are some highlights:

- New semantic tags
- Multimedia (new Audio and Video tags)
- Offline storage (local storage)
- 3d graphics support
- CSS3 animations and effects

HTML5

We will take a very brief look at a number of the HTML semantic tags that are available for use in your projects.

NEW HTML5 TAGS

class Description

section Container tag used for page organization

header Container for introductory and navigational stuff

footer Container for footer content (site map, internal links, etc.)

nav Container for a major block of navigation links

audio Multimedia tag for playing audio files

video Multimedia tag for playing video files

canvas element can be used to draw graphics via JavaScript

RESOURCES

- [HTML5 Please](#)
- [HTML5 Rocks](#)
- [CanIUse Compatibility Tables](#)

CSS PSUEDO-CLASSES AND PSUEDO- SELECTORS

CSS PSUEDOCASSES

Psuedo-classes are used to add special effects to some selectors.

PSUEDO-CLASS SYNTAX

```
selector:pseudo-class { property:value }
```

```
selector.class:pseudo-class { property:value }
```

ANCHOR PSUEDO-CLASSES

```
a:link { color: aliceblue; } /* unvisited link */
a:visited { color: darkblue; } /* visited link */
a:hover { color: lightblue; } /* mouse over link */
a:active { color: yellow; } /* mouse click link */
```

OTHER PSEUDO-CLASSES

class	Description
:first-child	Any element which is the first child of its parent
:first-letter	First letter of an element's content
:first-line	First line of an element's content
:last-child	Any element which is the last child of its parent
:only-child	Any element which is the only child of its parent
:empty	Any element which has no content
:nth-child	Any element which is the nth child of its parent

NTH-CHILD SYNTAX

`:nth-child` takes a mathmatic express to determine which children to select.

```
element:nth-child(an + b) { property:value; }
```

NTH-CHILD EXAMPLES

These expressions can be as complex as you want.

Feel free to mess around with it.

```
div:nth-child(3) { color:red; }
```

```
p:nth-child(4n) { color:blue; }
```

```
ul:nth-child:(2n + 5) { color:yellow; }
```

NTH-CHILD EXAMPLES

Realistically though, in practice `nth-child` is usually only used to alternate styles for large groups of data like table rows, etc.

```
element:nth-child(even) { property:value; }
```

```
element:nth-child(odd) { property:value; }
```

CSS3 FEATURES

CSS3 VENDOR PREFIX

Not all browsers support all of the newer CSS3 properties, which means we sometimes need to create rules for certain browsers

GRAND
CIRCUS
DETROIT

GRAND
CIRCUS
DETROIT

GRAND
CIRCUS
DETROIT

BORDER-RADIUS USAGE

```
.example {  
-moz-border-radius: purple;  
-ms-border-radius: purple;  
-webkit-border-radius: purple;  
-o-border-radius: purple;  
border-radius: purple;  
}
```

GRAND
CIRCUS
DETROIT

CIRCUS
DETROIT

GRAND
CIRCUS
DETROIT

GRAND
CIRCUS
DETROIT

GRAND
CIRCUS
DETROIT

GRAND
CIRCUS
DETROIT



VENDOR PREFIXES

Order matters! The non-prefixed property should always go last.

HANDY CSS3 PROPERTIES

There are a great many very handy CSS3 properties that you may need to use on your current and future projects. They can add a level of polish to your project that really sells it. That said, we dont have a time to go over even a very few of them in details so I'll just create a list that you can investigate further.

CSS3 PROPERTIES

Property	Description
<code>border-radius</code>	creates rounded corners or circular borders for elements
<code>rgba</code>	An alternate way to define color and alpha
<code>opacity</code>	Controls the transparency of an element
<code>box-shadow</code>	Adds a calculated shadow effect to an element
<code>text-shadow</code>	Adds a calculated shadow effect to an element
<code>transforms</code>	Adds the ability to transform HTML elements (rotation, scale, translate)
<code>animations</code>	Allows for one or more property changes to be applied as an animation

GRAND
CIRCUS
DETROIT

RESPONSIVE DESIGN

GRAND
CIRCUS
DETROIT



RESPONSIVE DESIGN

Changes your website presentation in relation to the screen size

MEDIA QUERIES

This is done via Media Queries. Media queries are able to detect a bunch of things about the device with which you're viewing a site. The browser can use specifically developed CSS styles for those different screen sizes.

DETROIT

GRAND
CIRCUS

DETROIT



MEDIA QUERY EXAMPLE

```
@media (max-width: 600px) {  
  /* specific styles for this device size */  
}
```

CIRCUS
DETROIT

GRAND
CIRCUS

CIRCUS
DETROIT

GRAND
CIRCUS

GRAND
CIRCUS

GRAND
CIRCUS

GRAND
CIRCUS

GRAND
CIRCUS

THINGS TO DO WITH MEDIA QUERIES

- resize images / font
- collapse navigation
- remove some larger content (video / images)



HTML FORMS

FORMS

Forms are used to send data back to a server from user inputs



FORM TAGS

```
<form action="something" method="get">  
    <!-- Form inputs and other html -->  
</form>
```

METHODS AND ACTIONS

Action is the URL you want the data to be sent to

Method is the HTTP methods used to transfer data

- GET
- POST
- PUT
- DELETE

FORM TAGS

```
<input type="text" method="get" name="firstname" placeholder="Name?">
```

TYPES OF INPUTS

- Text (type='text')
- Radio Button (type='radio')
- Checkbox (type='checkbox')
- Password (type='password')
- Date (type='date')
- Time (type='time')
- Date & Time (type='datetime')
- Email (type='email')
- Submit
 - `<input type='submit' />`
 - `<button type='submit' />`



DEMO





BOOTSTRAP

BOOTSTRAP

HTML5 & CSS framework that combines

- Responsive Design
- Grid Positioning
- Iconography
- Mobile Templating
- Javascript plugins



ABOUT BOOTSTRAP

Bootstrap is a CSS framework that makes making halfway—decent—looking sites a snap!

It was created by Twitter and was made open source
a few years ago



BOOTSTRAP GRIDS

The best of CSS positioning and table layouts

- 12 columns wide
- Infinite rows
- Responsive-ready out of the box
- Uses only CSS



ROWS & COLUMNS

```
<div class="container">
  <div class="row">This is Row 1</div>
  <div class="row">
    <div class="col-md-6"> This is Row 2 Col 1 </div>
    <div class="col-md-6"> This is Row 2 Col 2 </div>
  </div>
</div>
```



CIRCUS
DETROIT

GRAND
CIRCUS
DETROIT

CIRCUS
DETROIT

GRAND
CIRCUS
DETROIT

DEMO

GRAND
CIRCUS
DETROIT



WHERE TO GET IT

<http://www.getbootstrap.com> > Getting Started

Hit the "Download Bootstrap" button



HOW TO USE IT

1. Unzip the folder and rename it "bootstrap"
2. Save "bootstrap" in the folder containing your html and css files
3. Link to bootstrap in your index.html file



GRAND
CIRCUS
DETROIT



```
<link href="bootstrap/css/bootstrap.min.css" rel="stylesheet">  
<script src="https://ajax.googleapis.com/ajax/libs/jquery/1.11.3/jquery.min.js"></script>  
<script src="bootstrap/js/bootstrap.min.js"></script>
```



GR

```
<!-- Latest compiled and minified CSS -->
<link rel="stylesheet" href="//maxcdn.bootstrapcdn.com/bootstrap/3.2.0/css/bootstrap.min.css">

<!-- Optional theme -->
<link rel="stylesheet" href="//maxcdn.bootstrapcdn.com/bootstrap/3.2.0/css/bootstrap-theme.min.css">

<!-- Latest compiled and minified JavaScript -->
<script src="//maxcdn.bootstrapcdn.com/bootstrap/3.2.0/js/bootstrap.min.js"></script>
```

Copy

Like jQuery, Bootstrap can also be linked via CDN.

Just linking to the Bootstrap CDN will make a noticeable difference on your page

Before

THIS IS MY HEADER

Some text Some text Some text



After

THIS IS MY HEADER

Some text Some text Some text



CSS COMPONENTS

Bootstrap provides a bunch of CSS Components

- Grid layout
- Buttons
- Primary and Secondary headings
- Form elements
- Automatic table styling
- All the things basically

CSS COMPONENTS

Bootstrap provides great docs which demonstrate each CSS component and how to add them to your site.

There are so many things available in just the CSS portion of the Bootstrap. We don't have time to look at them all.

USING BOOTSTRAP

As said before, all of the bootstrap components are shown on their docs but for the most part it's as simple adding some classes to your html.

ex:

```
<a href="#" role="button" class="btn btn-primary">A button!</a>
```

GRAND
CIRCUS
DETROIT

GRAND
CIRCUS
DETROIT

CIRCUS
DETROIT

GRAND
CIRCUS
DETROIT

GRAND
CIRCUS
DETROIT

GRAND
CIRCUS
DETROIT

GRAND
CIRCUS
DETROIT

LAB 2

BRINGING IT ALL TOGETHER

LAB 2

Create a blog page using Bootstrap Grid. The page should have the following:

Requirements

- (optional) Nav section on Left, 1/3rd page width
- (optional) Main section on Right, 2/3rd page width
- At least 1 faked article of blog content
- At least 1 floated image with text wrapping around it in the article
- Use bootstrap glyphicons
- An HTML form with at least 3 elements, one of which must be radio button group, check box group, date picker, or dropdown, i.e. **select**
- Footer with copyright, contact link, some other content
- Use CSS psuedo classes
- BONUS: Research and build in a Bootstrap JavaScript Plugin (ex. Carousel, Accordion, Collapse, Tooltip, Popover, or Modal)