Build a Webpage

Anatomy of a beautiful webpage

By morriswmz

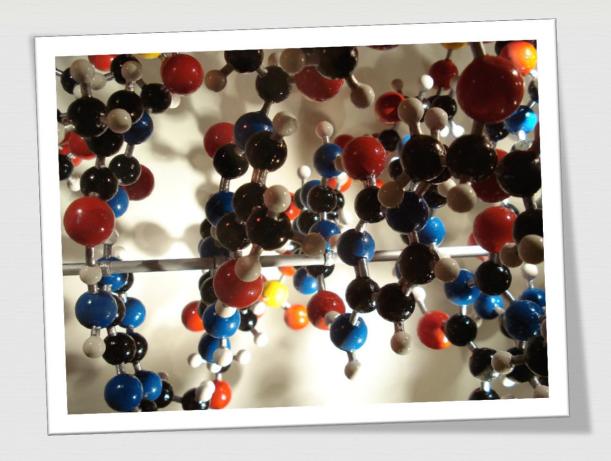
About last homework



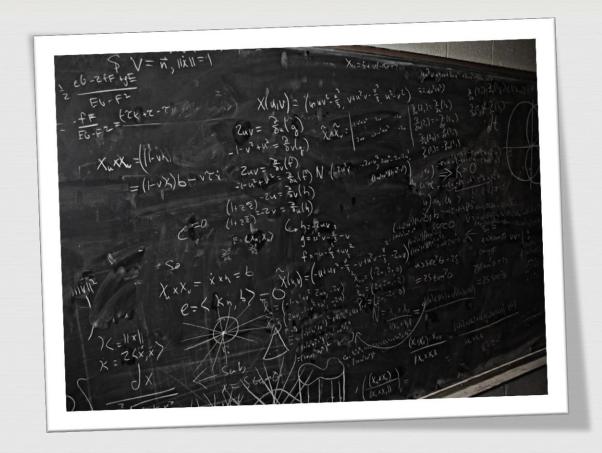


```
<l
      <a href="#"><img src="img/genericApp.png">App Na</a>
      <a href="#"><img src="img/appBut">
      <a href="#"><img src="img/genericApp.png">App Na</a>
      <a href="#"><img src="img/genericApp.png">App Na</a>
      <a href="#"><img src="img/genericApp.png">App Na</a>
   <a href="#">
       <img src="img/iPhoneGraphic.png" alt="App Store" id=</pre>
   </a>
                                     14kb
</div> <!-- end apps-->
<div id="banner" class="grid_12">
   <h2> Light Up Your iPhone </h2>
    >
       Lorem ipsum dolor sit amet, <strong>Aurora App</stro
```

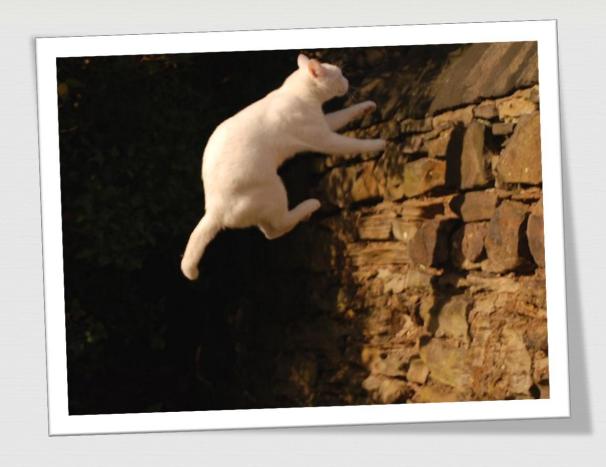
You wrote about 14kb of code



You encountered a bunch of new elements



You felt confused about them



However, you got an astonishing result. You built a beautifully designed, organized and structured webpage.



It is much better and more complex than THIS.



And it's HTML5!

Anatomy of a beautifully coded webpage

03

HTML Tag usage
Attributes Usage
CSS Naming
CSS Syntax

• • •

Use Correct HTML Tag

CB

```
<div id="content_layout">
 <h2>TT</h2>
 cli class="nav"><a href="#"></a>
   cli class="nav"><a href="#"></a>
   cli class="nav"><a href="#"></a>
   cli class="nav"><a href="#"></a>
 </div>
 <div id="content main">
   >
   Ut sit amet placerat nibh.
   >
   Mauris ultrices fringilla dignissim.
   </div>
</div>
```

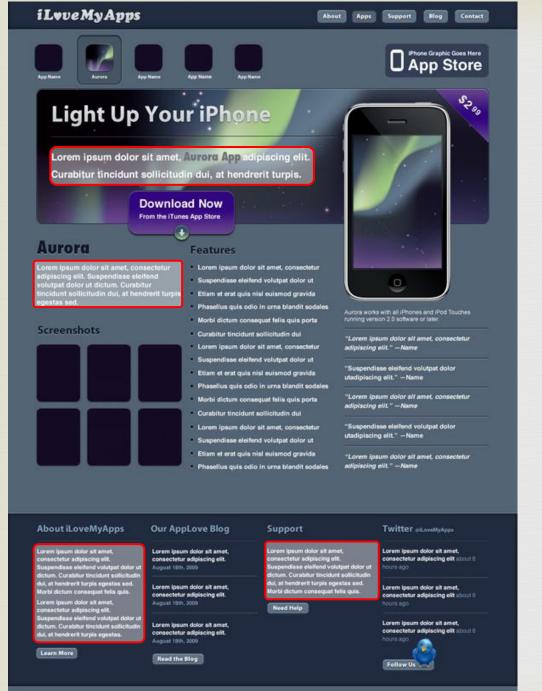
```
<div id="content_layout">
  <div class="content_title">TT</div>
  <div id="content nav container">
    <div class="nav"><a href="#"></a></div>
    <div class="nav"><a href="#"></a></div>
    <div class="nav"><a href="#"></a></div>
    <div class="nav"><a href="#"></a></div>
  </div>
  <div id="content main">
    <div class="paragraph">
    Ut sit amet placerat nibh.
    </div>
    <div class="paragraph">
    Mauris ultrices fringilla dignissim.
    </div>
  </div>
</div>
```



These are all lists. So use and



These are titles. So using <h1> ~ <h6> is preferred.



These are paragraphs. So we'd better use .

10000 ILove MyAzon, All Rights Reserved

*bout

ps :

upport

contact



Always use proper HTML Tags.

It will help maintainers understand the structure of the HTML page. HTML tags are meaningful.

Well-structured layout

03

Structure should be concise and flexible.

For large websites with lots of HTML pages, a well-done layout structure helps both when coding and maintaining.





Lorem ipsum dolor sit amet, consectetur adipiscing elit. rem ipsum dolor sit amet, nsectetur adipiscing elit. spendisse eleifend volutpat dolor u tum. Curabitur tincidunt sollicitude nust talls 2000 onsectetur adipiscing elit. Justinordisse eleitead volutra Suspendisse eleifend volutpat dolor u dictum. Curabitur tincidunt sollicitudi dui, at hendrerit turpis egestas. orem ipsum dolor sit amet, onsectetur adipiscing elit. ugust 18th 2009

ca header navigator

ca feature

content

3 column

ca footer

cs 4 column

About CSS Template



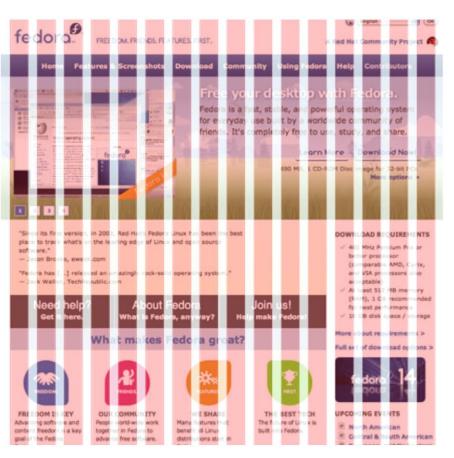
Simplify the layout. We don't need to write our framework from scratch if we follow the template.

960 grid system is commonly used in our design.

| container_12 | | | | | |
|--------------|--------|--------|--|--|--|
| grid_12 | | | | | |
| grid_4 | grid_4 | grid_4 | | | |
| | | | | | |

| container_12 | | | | |
|--------------|--------|--------|--------|--|
| grid_3 | grid_3 | grid_3 | grid_3 | |
| | | | | |





Use ID & Class Wisely

03

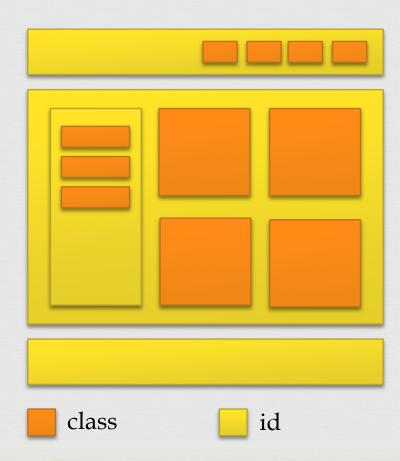
```
<div id="content_layout">
 <h2>TT</h2>
 cli class="nav"><a href="#"></a>
   cli class="nav"><a href="#"></a>
   cli class="nav"><a href="#"></a>
   cli class="nav"><a href="#"></a>
 </div>
 <div id="content main">
 </div>
</div>
<div id="footer_layout">
 <div id="footer content">
 </div>
</div>
```

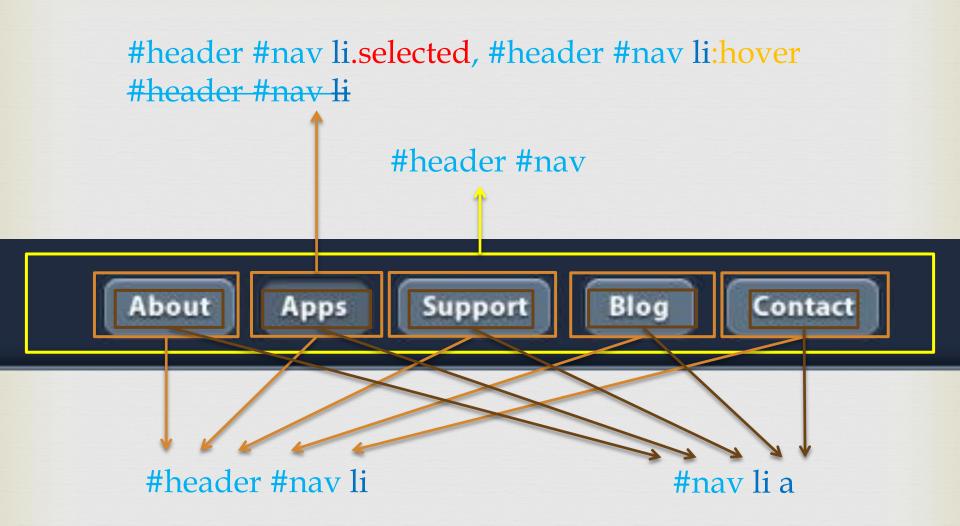
```
<div class="content layout">
 <h2>TT</h2>
 cli class="nav"><a href="#"></a>
   class="nav"><a href="#"></a>
   cli class="nav"><a href="#"></a>
   cli class="nav"><a href="#"></a>
 </div>
 <div class="content main">
 </div>
</div>
<div class="footer_layout">
 <div class="footer_content">
 </div>
</div>
```

Use ID & Class Wisely



- characteristics
- class for common or shared styles
- Generally, #ID will override properties specified in .class. Use this wisely.





Don't Use Inline Style

03

```
<div id="content_layout">
 <h2>TT</h2>
 </div>
 <div id="content main">
 </div>
</div>
<div id="footer_layout">
 <div id="footer content">
 </div>
</div>
```

```
<style>
</style>
<div style="width: 960px;">
  <h2 style="font-size: 20px">TT</h2>
  </div>
 <div style="width: 840px; margin-left: 20px; ...">
 </div>
</div>
<div style="width: 960px;">
  <div style="width: 940px; margin: 0 auto;">
 </div>
</div>
```

Don't Use Inline Style

CS

CR They

- cs make HTML file larger
- cs make HTML file messy
- cs lead to a lot of duplications
- are difficult to maintain



Naming Properly

03

```
<div id="content_layout">
 <h2>TT</h2>
 cli class="nav"><a href="#"></a>
   cli class="nav"><a href="#"></a>
   cli class="nav"><a href="#"></a>
   cli class="nav"><a href="#"></a>
 </div>
 <div id="content main">
   >
   Ut sit amet placerat nibh.
   >
   Mauris ultrices fringilla dignissim.
   </div>
</div>
```

```
<div id="middle part">
 <h2>TT</h2>
 cli class="style1"><a href="#"></a>
   cli class="style1"><a href="#"></a>
   cli class="style1"><a href="#"></a>
   cli class="style1"><a href="#"></a>
 </div>
 <div id="right_part">
   >
    Ut sit amet placerat nibh.
   >
    Mauris ultrices fringilla dignissim.
   </div>
</div>
```



Think carefully when you name them.

Write Clean CSS

03

```
#header_container {
  width: 940px;
  height: 80px;
  #header_logo {
    width: 220px;
    height: 80px;
    float: left;
    background-image: url('../images/logo.png');
    background-position: 0 16px;
    background-repeat: no-repeat;
  #header_nav {
    float: right;
    padding-top: 16px;
```

```
#header_container { width: 940px; height: 80px; }
#header_logo { width: 220px; height: 80px; float: left;
background-image: url('../images/logo.png');
background-position: 0 16px; background-repeat:
no-repeat; }
#header_nav {
  float: right;
  padding-top: 16px;
}
```



You can easily pick up mistakes in a clean CSS.

Keep It Short

03

```
top right bottom left
```

margin: 10px 20px 30px 40px;

top right bottom left

padding: 10px 20px 30px 40px;

background: #fff url('images/aaa.png') norepeat 0 0;

```
margin-top: 10px;
margin-right: 20px;
margin-bottom: 30px;
margin-left: 40px;
```

```
padding-top: 10px;
padding-right: 20px;
padding-bottom: 30px;
padding-left: 40px;
```

background-color: #fff;

background-image: url('images/aaa.png');

background-repeat: no-repeat;

background-position: 0 0;



It saves the server a lot if you just make the CSS file a little smaller and your browser love short ones too.

Homework



Sorry, but I have to...
And no tutorial this time



Zachery Mocha's Portfolio arcu posuere

arcu posuere rhoncus sed non justo

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Etiam hendrerit massa id lacus semper eu auctor sem lacinia. Proin fringilla magna id ipsum rhoncus ultrices. Proin ipsum quam, accumsan in accumsan fermentum.







HOME
GALLERY
STORIES
CONTACT

INKS

2011 9/22

Nam in ipsum ac arcu posuere rhoncus sed non justo

Category: Feelings Tags: None

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Etiam hendrerit massa id lacus semper eu auctor sem lacinia. Proin fringilla magna id ipsum rhoncus ultrices. Proin ipsum quam, accumsan in accumsan eget, aliquet pellentesque libero. Suspendisse condimentum faucibus fermentum. Sed ut justo dui. Praesent laoreet elementum mollis. Aliquam egestas feugiat ultrices. Duis consequat magna eget ante volutpat et bibendum nibh faucibus. Integer justo quam, pretium id cursus eu, pulvinar volutpat risus. Fusce fringilla auctor metus sed tristique.

Sed quis sem a nisl volutpat molestie. Nullam at augue elit, sit amet sodales erat. Fusce nulla est, accumsan quis faucibus rutrum, semper et eros. Nam a rutrum quam. Integer tortor justo, consequat quis dictum vitae, porta nec sapien. Nulla hendrerit mi in tortor sodales a aliquam orci laoreet. Donec feugiat magna accumsan urna hendrerit quis consectetur nibh laoreet. Suspendisse molestie facilisis tellus eget feugiat. Morbi sed mauris non sem bibendum iaculis sollicitudin nec nulla. Aenean ligula diam, facilisis id varius nec, vestibulum sed risus. Suspendisse potenti. Cras aliquet odio vel tortor convallis ullamcorper. Nulla facilisi. Aliquam nisl orci, congue id eleifend quis, iaculis in leo. In eu bibendum tellus. Pellentesque habitant morbi tristique senectus et netus et malesuada fames ac turpis egestas.

Cum sociis natoque penatibus et magnis dis parturient montes, nascetur ridiculus mus. Curabitur quis orci enim, sit amet vehicula justo. Pellentesque nibh lorem, convallis et semper nec, iaculis in nulla. Sed orci tellus, consectetur id sodales a, feugiat et sem. Nunc non nunc nec sapien dapibus posuere. Praesent magna dolor, volutpat vitae iaculis ac, accumsan et nibh. Sed sem felis, tincidunt vel egestas a, volutpat ac lectus. Nulla fermentum pulvinar dictum.

Cras elementum facilisis odio pellentesque congue. Nullam vitae molestie dui. Nulla sollicitudin

-- Massa Feugiat

Class aptent taciti sociosqu ad litora torquent per conubia nostra, per inceptos himenaeos. Proin id pretium mi. Pellentesque eleilend eros auctor diam condimentum convallis.

-- Lorem Ipsum

Further Reading

03

○ Do's and Don'ts of Writing Better CSS and HTML

 http://www.1stwebdesigner.com/tutorials/dos-donts-better-css-and-html/