

# CSS: Core Principles | 80% rule

Upload files to SLACK

Download them to folder ~5142

Open them and discuss your work

## Backgrounds

1. Background color
2. Background images
3. Background gradients
4. Opacity vs. RGBA

## Layouts

1. Box Model
2. Margin and padding
3. Border box model
4. Borders

## The Navbar

1. Block vs. inline
2. Floats (and clears)
3. Styling the list
4. Styling the links

## Typography

1. CSS Reset (Eric Meyer)
2. Google Fonts
3. H1-H6, paragraphs, blockquotes
4. Typography basics


# Background Color

Every html element is a box

Every box can have a background color

Parent elements sit behind child elements

 `<body>`

 `<div class="container">`

 `<div class="row">`



# Background Images

Every element can have a background image

The background image is in front of the background color

Background images will always tile and repeat UNLESS you tell them not to

Repeat: no-repeat | repeat-x (horizontal axis) | repeat-y (vertical axis)

Background-size: cover | contain | percentage

Background-position: Left/right/top/bottom/center | pixels | percentages

# Background Images

Both .png and .gif support transparency

If you cut part of the image away, you can see the background color through that opening

```
background-color: orange;  
background-image:  
trans_cat.png
```



# Background Gradients

```
background: linear-gradient(to right, #6A82FB,  
#FC5C7D); /* W3C, IE 10+/ Edge, Firefox 16+, Chrome  
26+, Opera 12+, Safari 7+ */
```

<https://uigradients.com>

<https://webgradients.com/>

<http://www.colorzilla.com/gradient-editor/>

# Background Gradients

```
background: #1e5799; /* Old browsers */
background: -moz-linear-gradient(top, #1e5799 0%, #2989d8 50%, #7db9e8 100%); /* FF3.6-15 */
background: -webkit-linear-gradient(top, #1e5799 0%,#2989d8 50% 51%,#7db9e8 100%); /* Chrome10-25,Safari5.1-6 */
background: linear-gradient(to bottom, #1e5799 0%,#2989d8 50%,#7db9e8 51%,#7db9e8 100%); /* W3C, IE10+, FF16+, Chrome26+, Opera12+, Safari5.1-6 */
filter: progid:DXImageTransform.Microsoft.gradient( startColorstr=#1e5799, endColorstr='#7db9e8', GradientType=0 ); /* IE6-9 */
```



# Opacity vs. RGBA

Opacity can be expressed in terms of 0 to 1, where .5 is an opacity of 50%  
But it turns the ENTIRE element and the contents transparent..

RGBA changes the transparency of ONLY the background via the alpha channel

```
#header {background: blue, opacity:.5;}
```

```
#header {background: rgba(15,93,198,.5)}
```

# Box Model

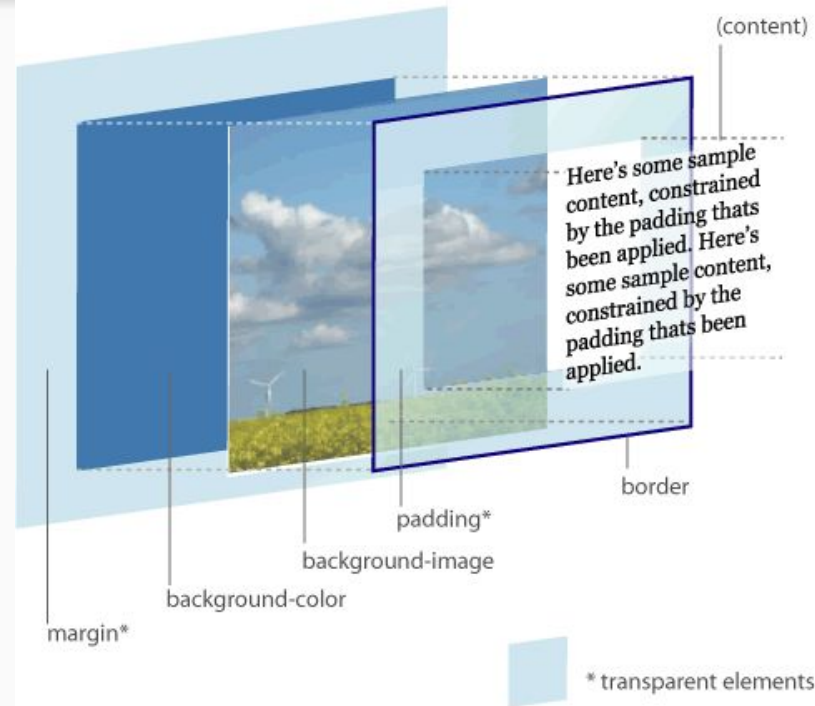
Every HTML element is a box

Semantic or Structural

A box can have padding, borders, margin

A box can have background color, image or gradient—BEHIND the padding

THE CSS BOX MODEL HIERARCHY



# Margin and Padding

Both create “white space”; visually, it doesn’t matter how you get it

Padding is inside the box

Margin is outside the box; pushes elements away; aka “the force field”

The old way: Padding makes the box bigger

```
.box {width: 200px; height: 100px; padding: 10px; border: 1px;}
```

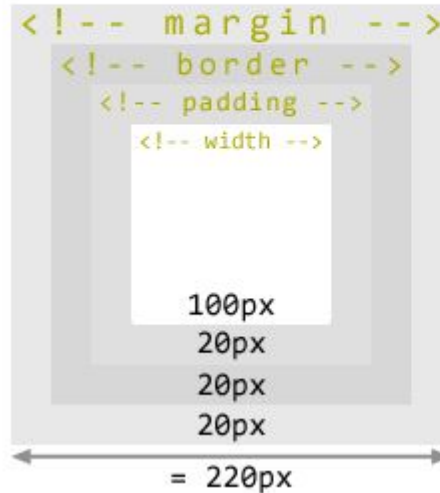
**200+10+10+1+1= 222px (so frustrating!)**

# Border Box Model | box-sizing: border-box

Includes the width of padding and borders within the total; pushes content inwards.

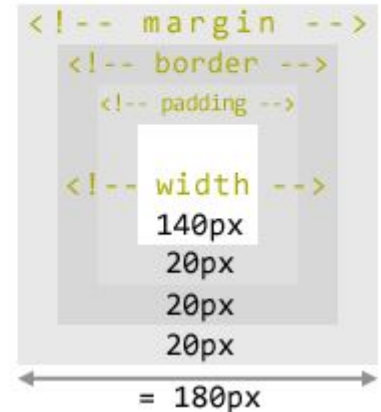
## CSS Box Model

`box-sizing: content-box;`



`box-sizing: border-box;`

As opposed to the content-box model, the border-box model includes the border and padding inside of the width.



# Borders

Border: thickness color style

2px black dashed | thick #000 solid

Border-top; border-bottom; border-left; border-right;

Border-radius: pixels or percentage to create rounded corners

# Block vs. Inline

All semantic markup (text) wants to display 'INLINE' by default. If a sentence was long enough and a browser window wide enough it would keep stretching out in a fluid, or elastic manner.

By telling an element to display: block; we can give it a specific height and width

```
.box {display: block;  
height:200px; width:300px;padding:10px;  
box-sizing:border-box;  
}
```

# Floats and Clears

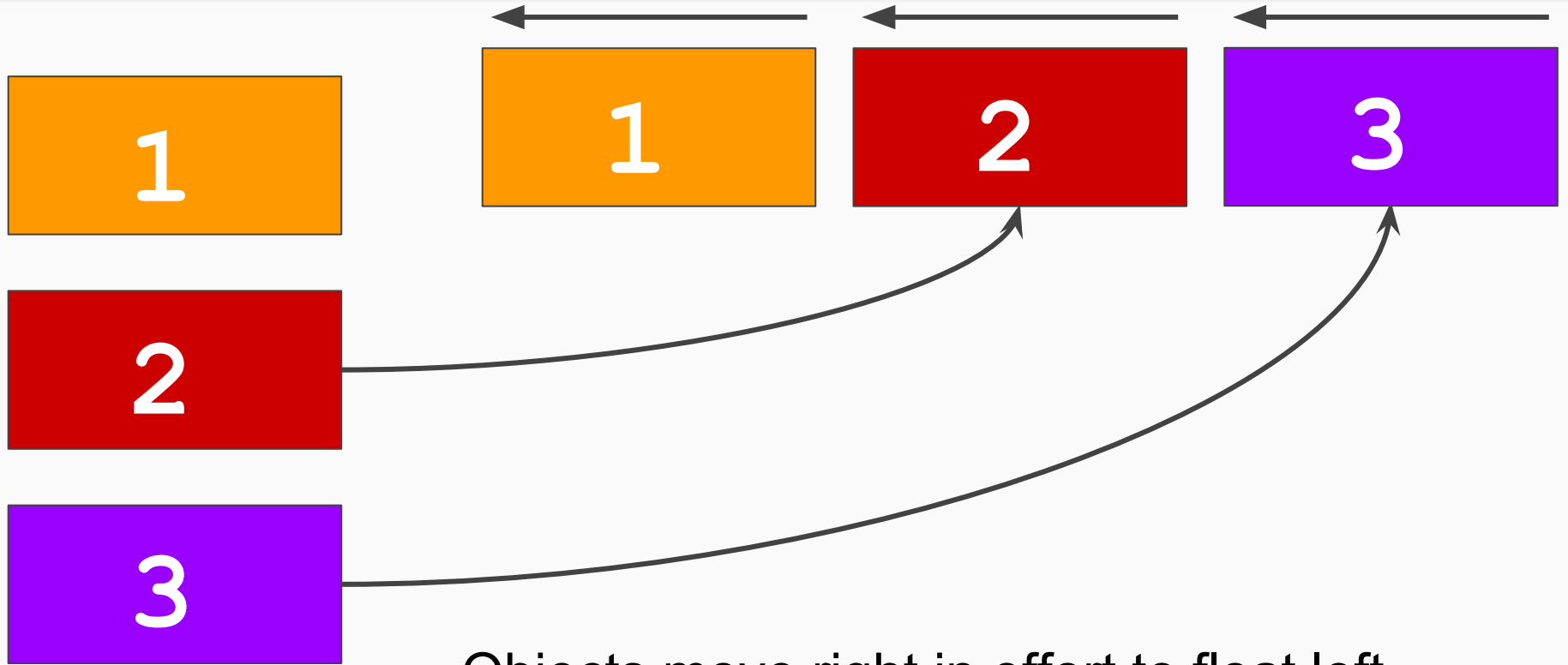
Every element is a box; all boxes stack in a 'top-to-bottom' stacking order called the DOM (Document Object Model)

When we tell an object to 'Float' it means we pull it out of that column and over to the left or right side of the parent element

Unfortunately, floats act like a gravitational pull on other elements and pull all other elements up the page

We 'break the spell' of preceding floats by applying the 'clear' property  
Float: left; float: right; clear: left; clear:right; clear:both;

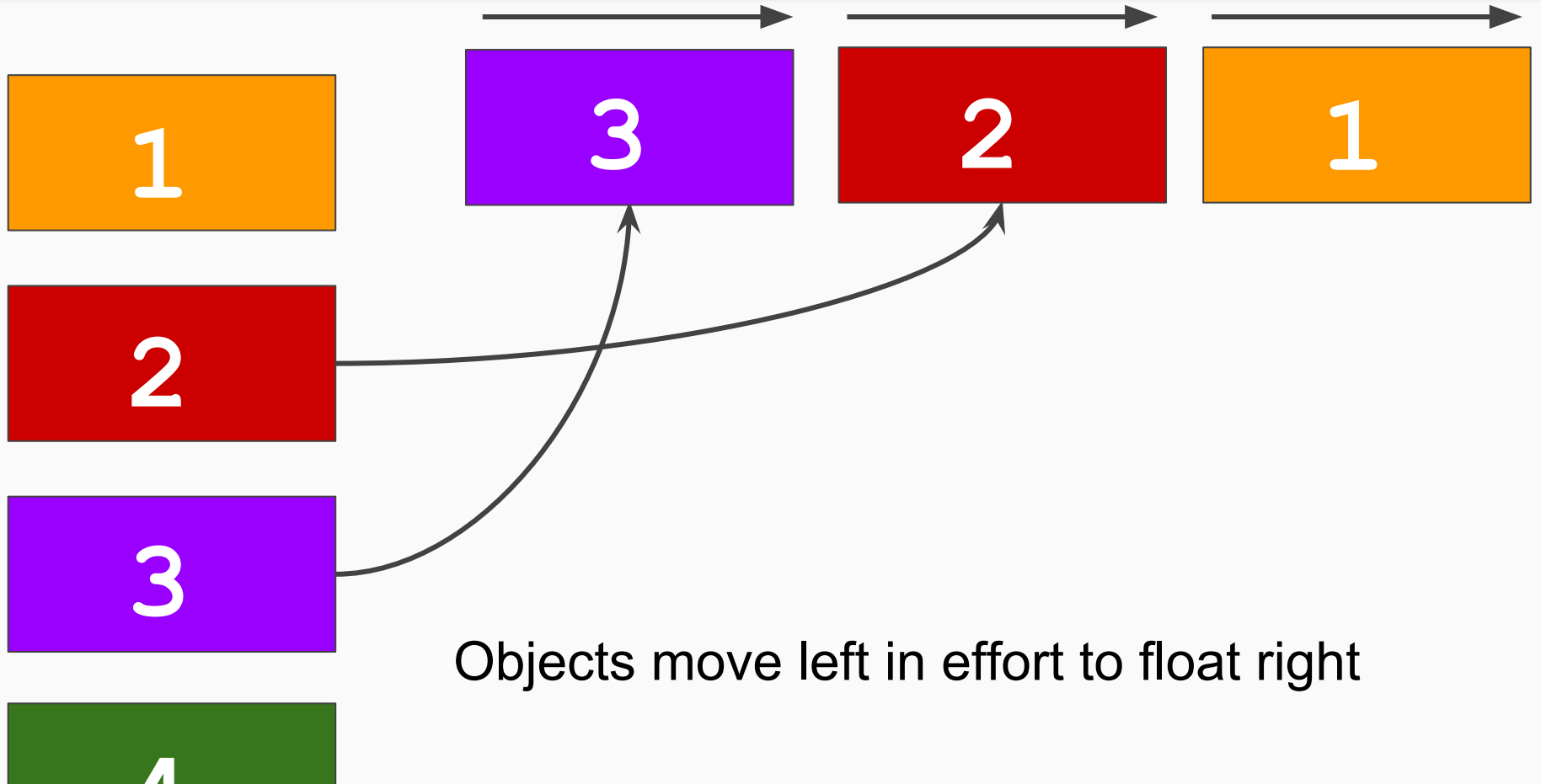
# Floats and Clears



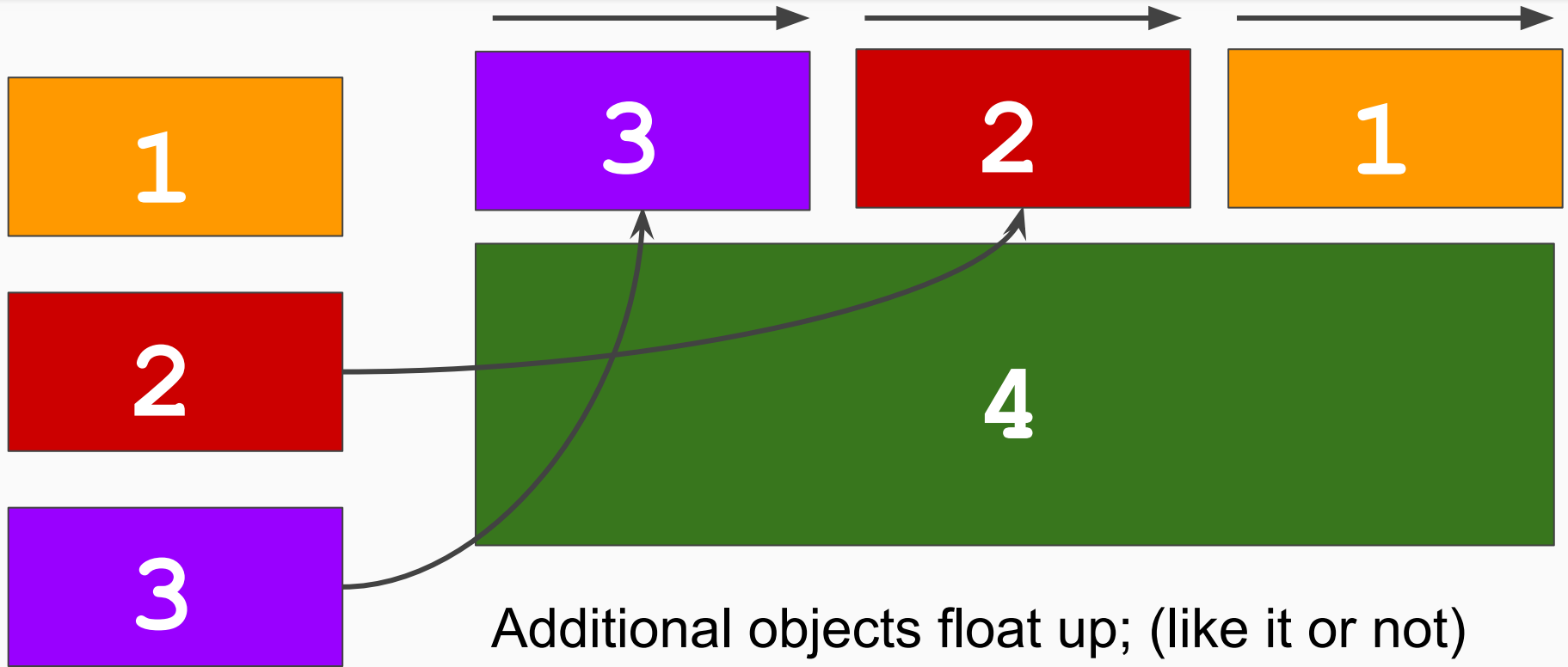
Objects move right in effort to float left



# Floats and Clears



# Floats and Clears



Additional objects float up; (like it or not)  
Unless you tell them to clear


# Styling the List

The default list stacks in a column and has ugly bullet points

Strip the list by telling the `<ul>` tag to have a `list-style-type:none;`

```
ul {list-style-type:none;}
```

```
<ul>
<li><a href="#">History</a></li>
<li><a href="#">Flavors</a></li>
<li><a href="#">Community</a></li>
<li><a href="#">Contact</a></li>
</ul>
```

- History
  - Flavors
  - Community
  - Contact
- 
- History
  - Flavors
  - Community
  - Contact

Next, we need to restyle the `<li>` tag (or list item)

```
li {display:block;  
    width:100px;  
    height:40px;  
    float:left;  
    margin-right:10px;  
    outline: thin red solid;  
}
```

History

Flavors

Community

Contact

# Styling the links

Next, we need to place most of our visual styling to the `<a>` tag

```
a {display:block;
  width:100px;
  height:40px;
  text-align:center;
  Line-height:40px;
  Background-color:white;
  color:#333;
  font-family:Georgia, serif;
  border-radius:8px;
}
```



# Styling the links

Next, we the hover state

```
a:hover {background-color:yellow;}
```

History

Flavors

Community

Contact

# CSS Reset

Remember web standards?

Every browser renders each html element a little differently

The amount of space around each tag varies as well

# I am a first level header, or h1

## I am a second level header, or h2

### I am a third level header, or h3

I am a paragraph...Hi, I'm Evgeny Suslikov. I'm 32-years old and I spend my days as a software developer. I love creating things...

Developer Tools - file:///Users/ccs6982/Documents/erik/public\_html/erikfadiman/bluebell\_demo/test.html

Elements Console Sources Network Performance Memory Application Security Audits

```
<!DOCTYPE html>
<html>
  <head>...</head>
  <body>
    <h1>I am a first level header, or h1
  </h1> == $0
    <h2>I am a second level header, or h2
  </h2>
    <h3>I am a third level header, or h3
  </h3>
    <p>...</p>
  </body>
</html>
```

html body h1

Styles Computed Event Listeners >>

margin 21.440  
border -  
padding -  
1664 x 37  
21.440

Console What's New x



I am a first level header, or h1

I am a second level header, or h2

I am a third level header, or h3

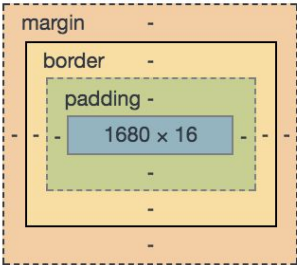
I am a paragraph...Hi, I'm Evgeny Suslikov. I'm 32-years old and I spend my days as a software developer. I love creating things...

Developer Tools - file:///Users/ccs6982/Documents/erik/public\_html/erikfadiman/bluebell\_demo/test.html

Elements Console Sources Network Performance Memory Application Security Audits

```
<!DOCTYPE html>
<html>
  <head>...</head>
  <body>
    <h1>I am a first level header, or h1
    </h1> == $0
    <h2>I am a second level header, or h2
    </h2>
    <h3>I am a third level header, or h3
    </h3>
    <p>...</p>
  </body>
</html>
```

Styles Computed Event Listeners >>



margin -  
border -  
padding -  
1680 x 16

html body h1

Console What's New x

# H1-H6, Paragraphs

```
H1, h2, h3, h4, h5, h6 {font-family: Crimson-text; font-weight:bold; color:#333;}
```

```
H1 {font-size:36px;. padding-bottom:12px;}
```

```
H2 {font-size:30px; padding-bottom:8px;}
```

```
H3 {font-size:24px; padding-bottom:4px;}
```

etc

```
23 ▼ * {
24     font-family: 'Lato', Helvetica, sans-serif;
25     color: #333447;
26     line-height: 1.5;
27 }
28
29 /* TYPOGRAPHY */
30
31 ▼ h1 {
32     font-size: 2.5rem;
33 }
34
35 ▼ h2 {
36     font-size: 2rem;
37 }
```

## Embed Font

To embed your selected fonts into a webpage, copy this code into the <head> of your HTML document.

**STANDARD**    **@IMPORT**

```
<link href="https://fonts.googleapis.com/css?family=Open+Sans" rel="stylesheet">
```

---

## Specify in CSS

Use the following CSS rules to specify these families:

```
font-family: 'Open Sans', sans-serif;
```

For examples of how fonts can be added to webpages, see the [getting started guide](#).

# Typography basics

Line-height

Letter-spacing

Text-transform

Font-weight

`<span>` tags

`<strong>` and `<em>` tags for bold and italic

## Backgrounds

1. Background color
2. Background images
3. Background gradients
4. Opacity vs. RGBA

## Layouts

1. Box Model
2. Margin and padding
3. Border box model
4. Borders

## The Navbar

1. Block vs. inline
2. Floats (and clears)
3. Styling the list
4. Styling the links

## Typography

1. CSS Reset (Eric Meyer)
2. Google Fonts
3. H1-H6, paragraphs, blockquotes
4. Typography basics

**Understanding < 80% ? MSG ME**