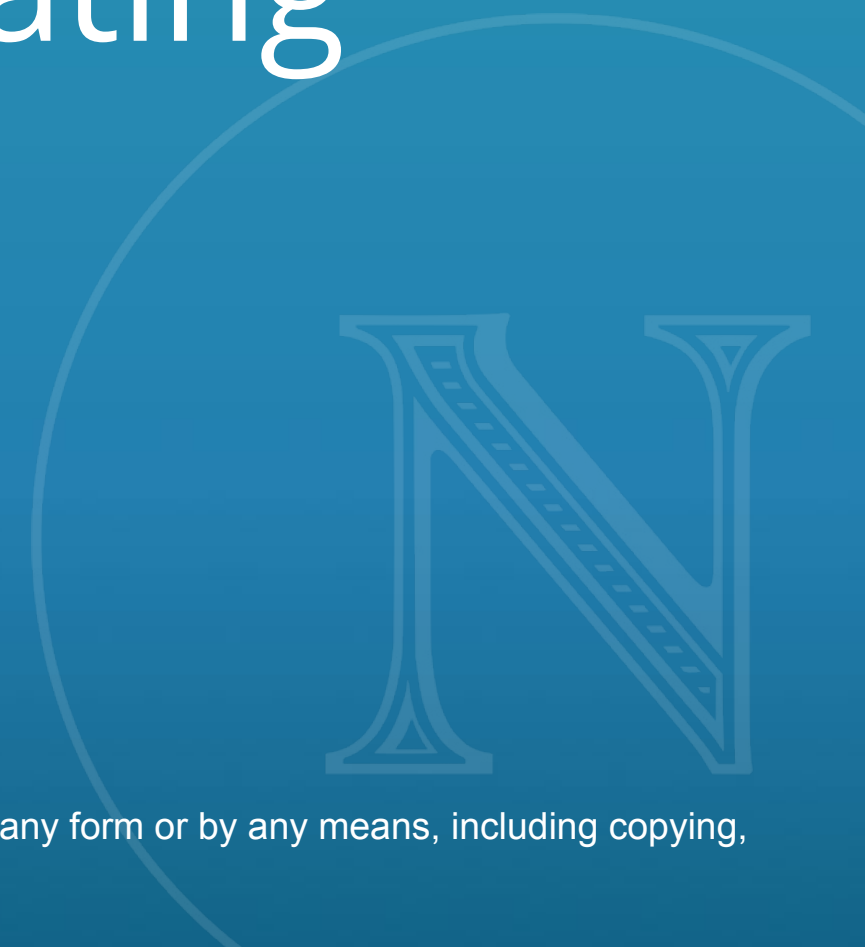# Introduction to Web Development

## Week 3: Creating layouts and creating responsive pages

# Agenda: Week 3

- **Layout Basics**

    ○ Class and id

    ○ Display and float

    ○ Positioning

- **Responsive web**

# Useful Things to Know

# Creating More Pages

## How do you create more pages?

Sublime Text > File > New (opens new file) >
Save As > new.html (in same folder as index.html)

## How do you link to them from the main page?

```
<!-- Like any other link! -->
<a href="new.html">New Page</a>
```

# Important Tidbits

Preparation for the main bits!

# Some Terminology

- **Element**
  - **One thing on a page that is surrounded by tags**

    **e.g. a paragraph is an element**

```
<p>
    hello world
</p>
```

# Some Terminology

- **Parent - Child**
  - **Parent:** an element that is **containing another**

  - **Child:** an element that is **contained by another**

  e.g.
    - `<li>Item 1</li>` is a **child** of a `<ul>`
    - `<ul>` is a **parent** of 3 `<li>`

```
<ul>
    <li>Item 1</li>
    <li>Item 2</li>
    <li>Item 3</li>
</ul>
```

# Class and Id

**Ways to identify unique elements:**

- **class**: to identify a **group** of element types
- **id**: to identify **ONE** special element

**Don't mix them up or your site might not look like how you expect it to be**

# Class

What if you want different colored headings?

E.g.

**I'm a red heading 2**

**I'm a blue heading 2**

# Class

## HTML

```html
<h2 class="red-text">
    I'm a red heading 2
</h2>
<h2 class="blue-text">
    I'm a blue heading 2
</h2>
```

## CSS

```css
.red-text {
    color: red;
}
.blue-text {
    color: blue;
}
```

to select for a class, it is fullstop followed by class-name (no space)

## OUTPUT

### I'm a red heading 2

### I'm a blue heading 2

# Class

What if you want same styles for headings and paragraphs?

**Fancy title** ⚬--------⚬ **\<h2\>**

**Look at me! I'm important** ⚬--------⚬ **\<p\>**

Both are red
and underlined

# CLASSES

## GROUP & LABEL DIFFERENT ELEMENTS TOGETHER

# Class

## HTML

```
<h2 class="attention">
    Fancy Title </h2>
<p class="attention">
    Look at me! I'm important
</p>
```

## CSS

```
.attention {
  color: red;
  text-decoration: underline;
}
```
/* So you don't have to repeat the styles for both h2 and p! */

## OUTPUT

# Fancy title

**Look at me! I'm important**

# Id

## What if you need to **id**entify one specific element?

`<div id="login-area"> Log me in! </div>`

*Not as important now, but will be when we talk about Javascript.*

# Id

## HTML

```
<button id="login">
    Log me in!
</button>
```

## CSS

```
#login {
    background-color: lightblue;
}
```

## OUTPUT

Log me in!

# Using div

## HTML

```
<div class="welcome">
    <p> paragraph a </p>
    <p> paragraph b </p>
</div>
```

## CSS

```
.welcome {
  border: 1px solid red;
}
```

## OUTPUT

paragraph a

paragraph b

# Using span

## HTML

```
<p> Welcome to divs and
    <span class="highlight">
      spans
    </span>
</p>
```

## CSS

```
.highlight {
    background-color:lightsalmon;
}
```

## OUTPUT

Welcome to divs and spans

# Bonus : Rainbow Poem

Using spans, make each line of a limerick a different color



There was an old man with a beard
Who said, "it's just how I feared!
Two owls and a hen
Four larks and a wren
Have all built their nests in my beard.
- Anonymous

# Layouts - Pt 1

The art of putting things into places where they are supposed to be

# Box Model Review

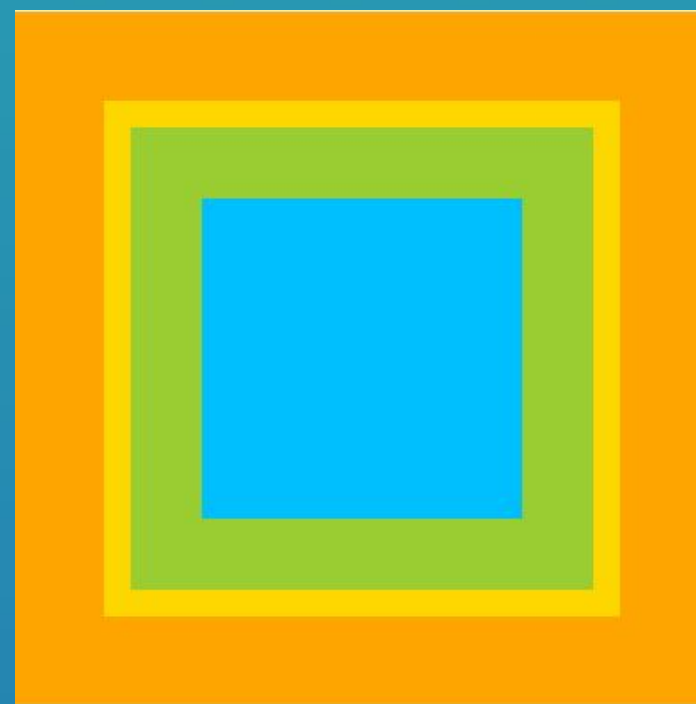# Box Sizing

When you set `width: 200px`, what are you **really** setting?

What happens if you add padding? Margin? Borders?

# Challenge: Boxes in boxes

How would you achieve this?



Bonus: I did it with 3 divs. Can you do it with less?

Hint: Use background-color, padding, margin and borders
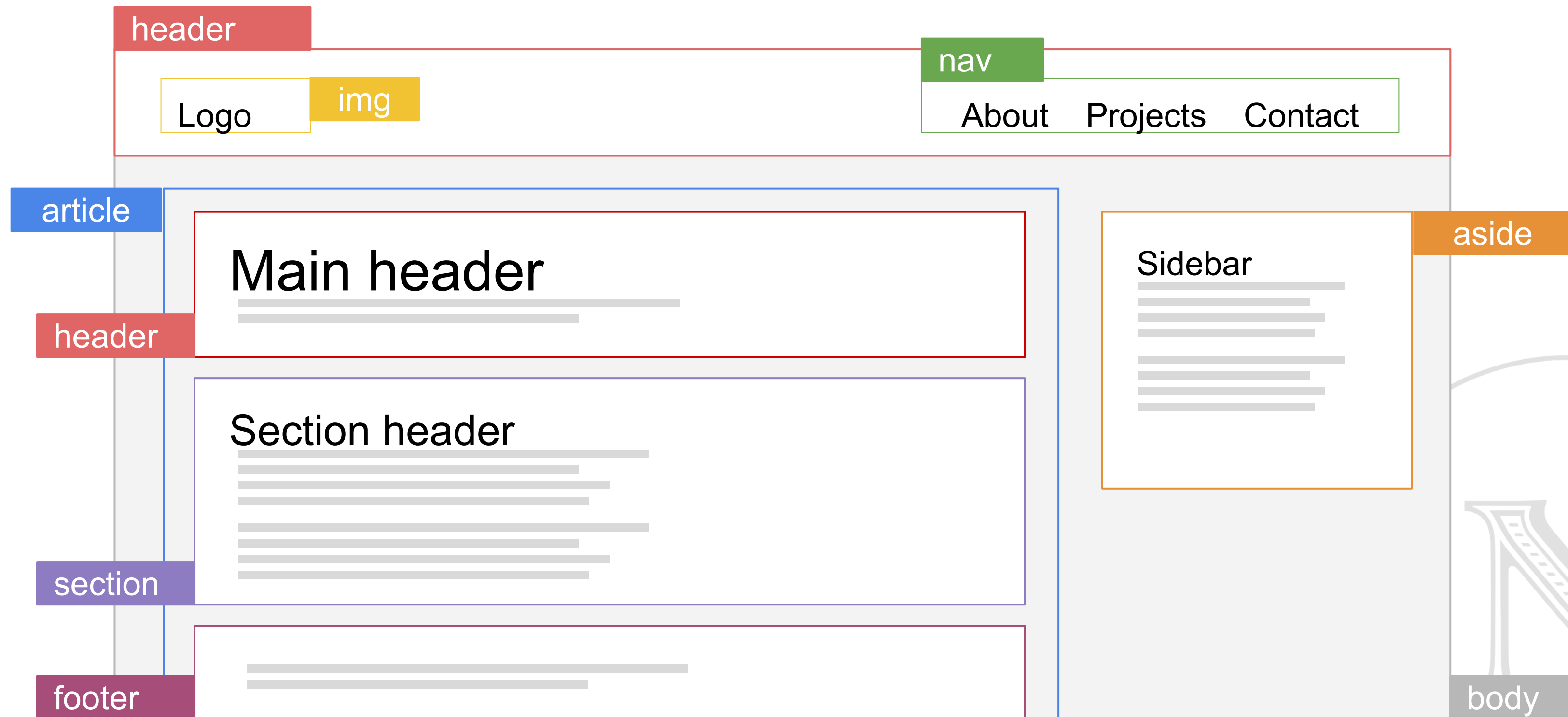
# Grouping stuff in meaningful ways

- **Semantic layout**
  - article
  - header, section, footer
  - nav
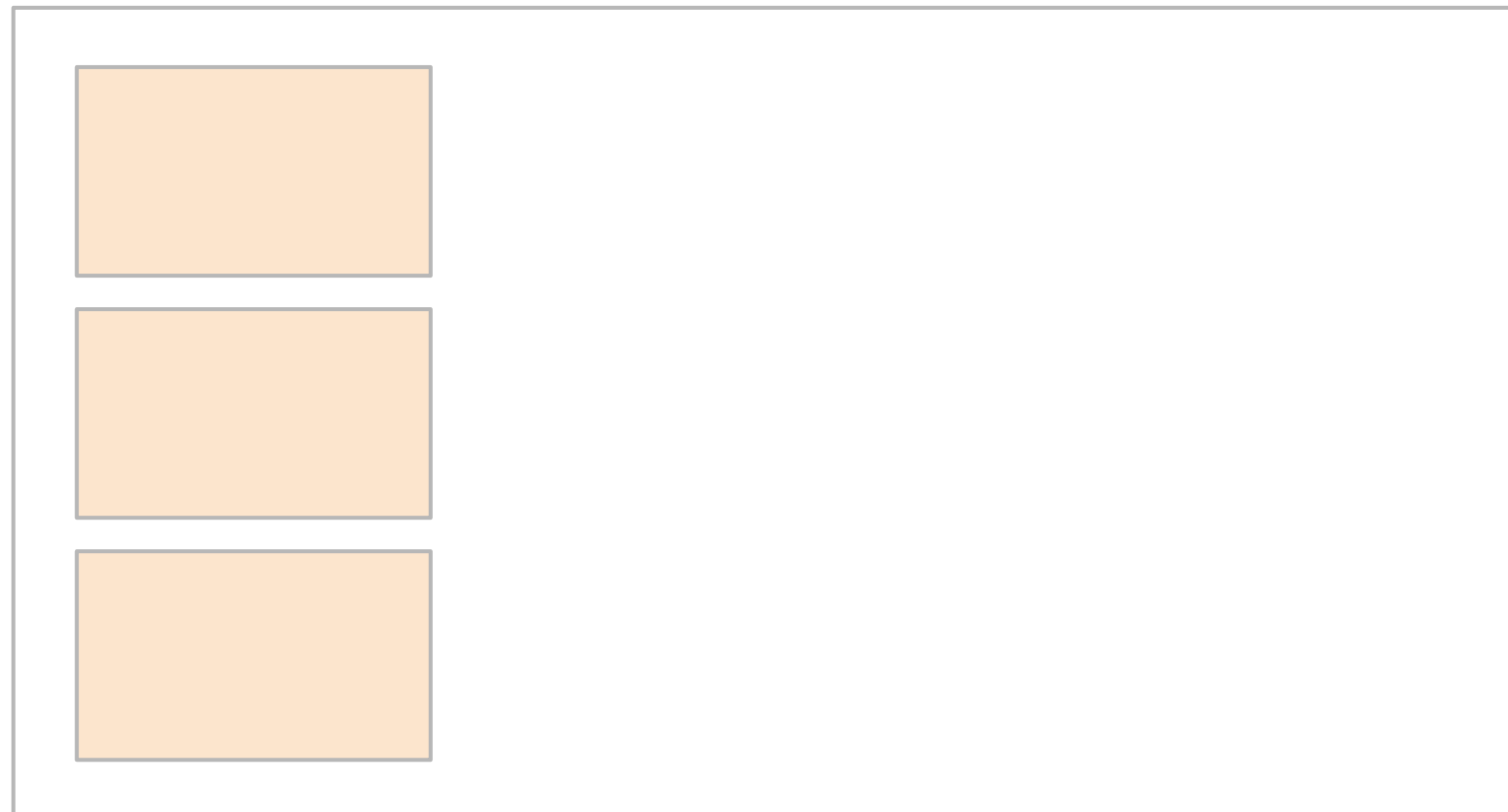  - aside
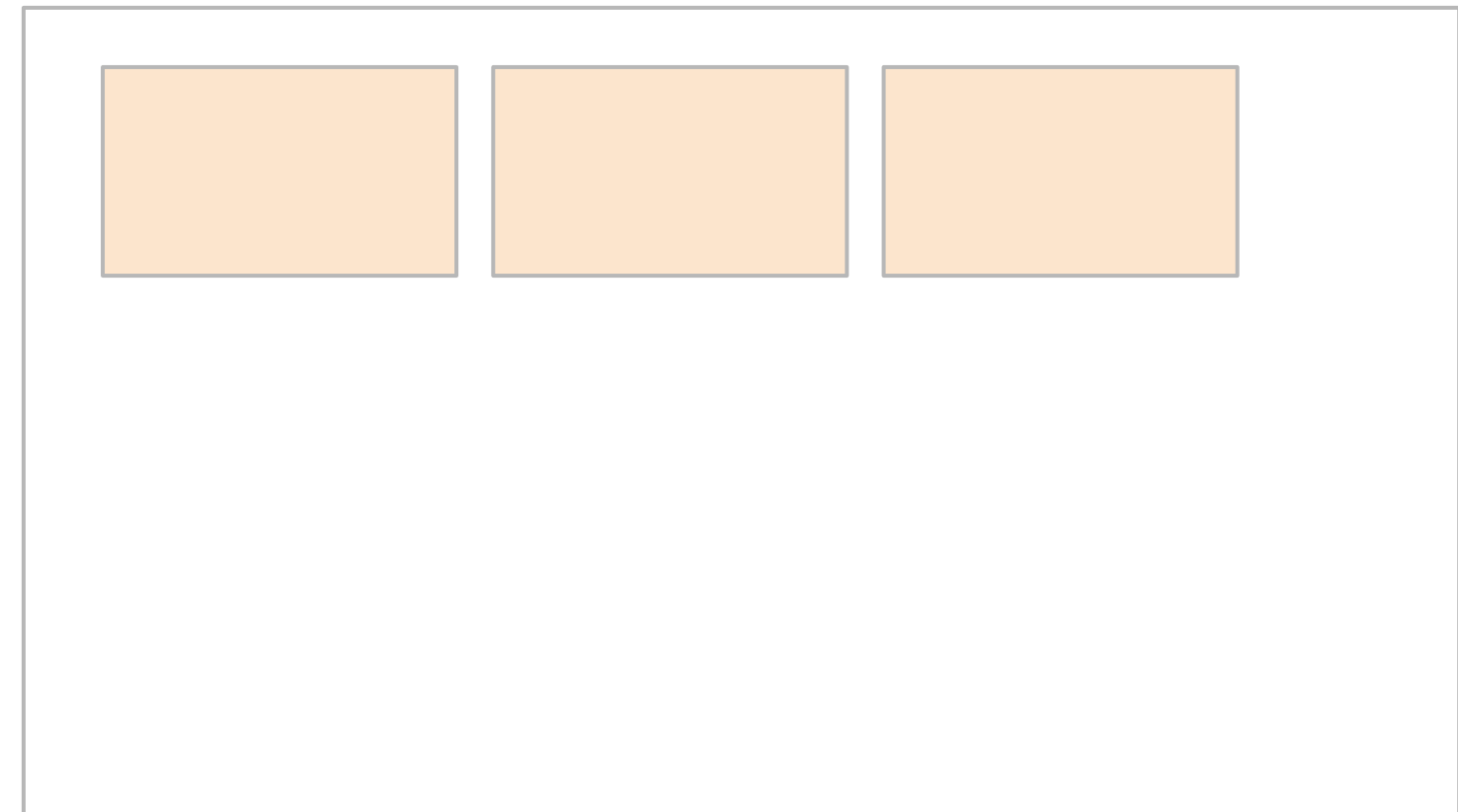- **For everything else**
  - div
  - span

# Semantic Layout

**header**

**nav**

Logo **img**

About    Projects    Contact

**article**

# Main header

**header**

## Section header

**section**

Sidebar **aside**

**footer**

**body**

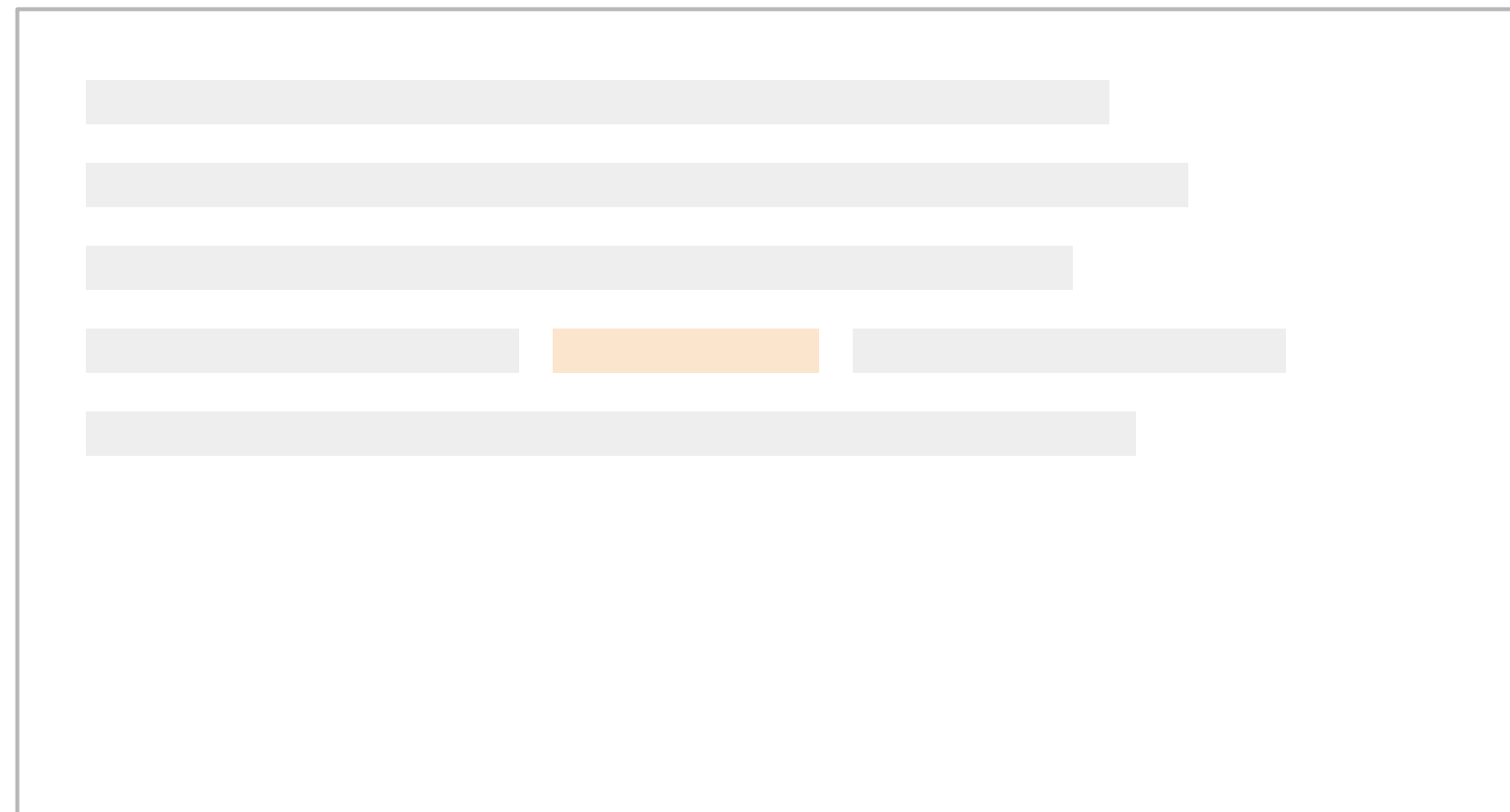# Display

- For setting the **flow** of your elements



one after another?
`display: block`

in a line?, but still individual 'blocks'?
`display: inline-block`

# Display

are they in lines of text?
display: inline

don't wanna display it?
display: none

# Display

- Options

    - block

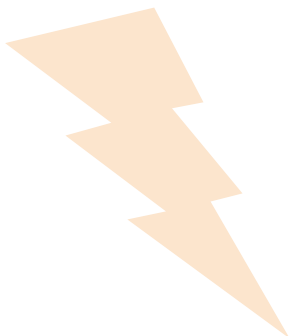    - inline-block

    - inline

    - none

    - and so on.

# Floats

- ## For pulling elements to the left or right of view
- ## Options
  - `float: left | right`
- ## Clearing floats
  - `clear: left | right | both`
  - ## Further reading: https://css-tricks.com/all-about-floats/

float:right;

| Logo | | About   Projects   Contact |

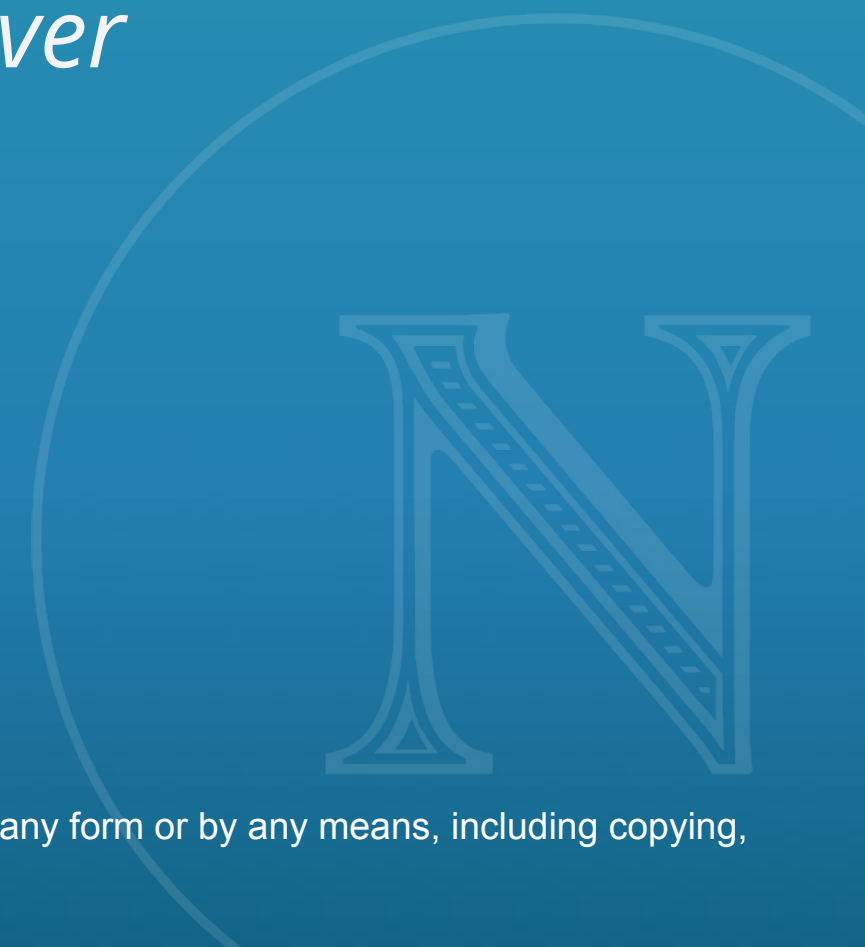# Display + Float Hands-on

# Recap

So far, we covered:

- class and id
- div and span
- box model and box-sizing
- display and float
- how to layout

# Layouts - Pt 2

On the `position` attribute, or, the hardest part of CSS, *ever*

# Positioning

- **When do you need it?**
  - Fixed position on screen
  - Specific positions
    - **20px from the right**
    - **15% from the top of the parent**

- **Options**

  - `position: static | absolute | relative | fixed`

# Positioning

- **When do you need it?**
  - **Fixed position on screen**
  - **Specific positions**
    - **20px from the right**
    - **15% from the top of the parent**

- **Options**

  *default*

  ↓
  - `position: static | absolute | relative | fixed`

# Getting into position...

Positioning is tricky! We won't go into too much details today, but...

I highly recommended reading to fully understand positioning:

http://learnlayout.com/position.html

# Back to top button

Most sites have a back-to-top link/button that scrolls to the top of a page. This can be achieved with

```
<a href="#">Back to top</a>
```

These buttons are also usually stuck at the bottom of a page. Add a back to top button to your new site with layout.

Hint: those buttons look like they're uh.. in a pretty fixed position on the page

NEXT ACADEMY

# A Special on Centering Elements

What happens when you want to position something EXACTLY in the center of the page?

Try adding this to `.kitty`:

`left: 50%`

`top: 50%`

Does it work? If not, why?

How can you get it to be exactly in the center?

# Responsive Web

# How do you make something responsive?

1. `max-width` **and** `min-width` **instead of** `width`

2. **set viewport meta**

3. **use** `@media` **queries**

4. `%` **instead of** `px`

# max-width instead of width
# + use % instead of px

```
article {

  width: 500px;

}
```

```
article {

  max-width: 500px;

  width: 100%;

}
```

# set viewport meta

```html
<head>

    <meta name="viewport"

            content="width=device-width"/>

</head>
```
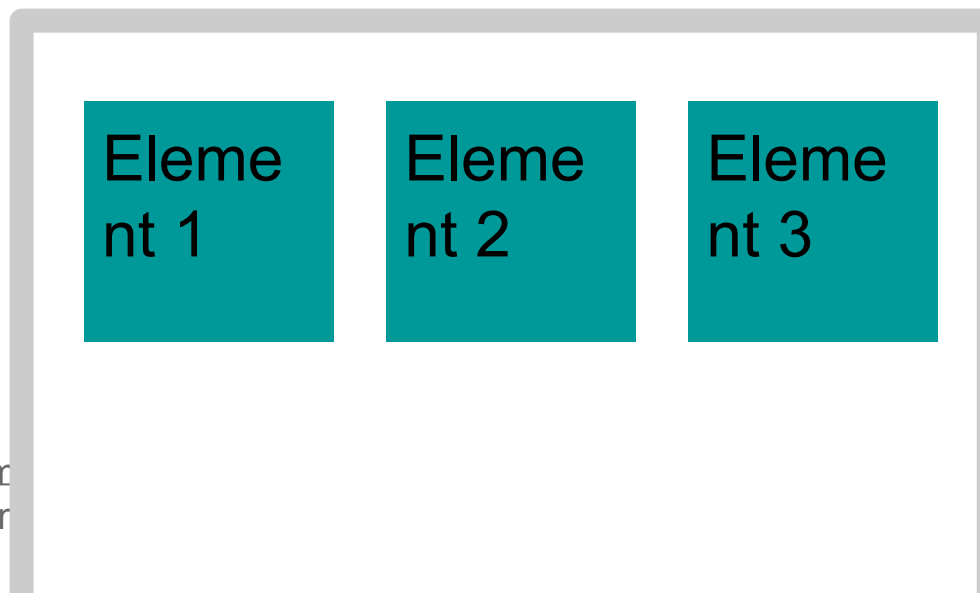
# @media queries: max-width

## HTML

```
<div class="elem">Element 1</div>
<div class="elem">Element 2</div>
<div class="elem">Element 3</div>
```
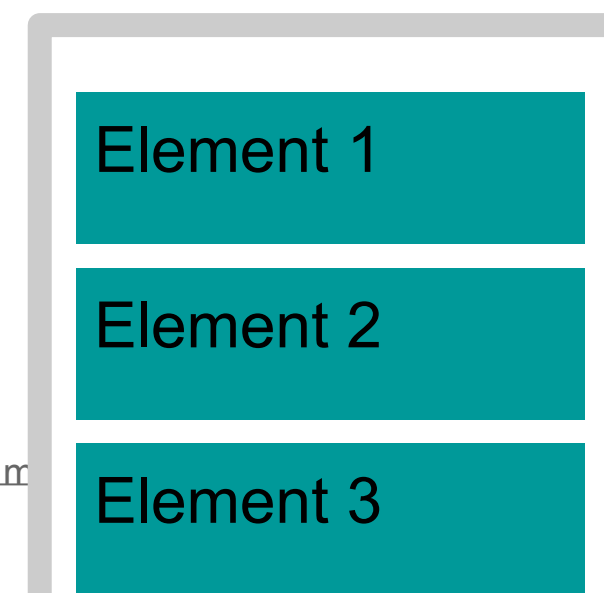
## CSS

```
.elem{
    display: inline-block;
}


@media(max-width: 480px){
  .elem{
    display: block;
  }
}
```

## OUTPUT

screen width  > 480px

| Eleme nt 1 | Eleme nt 2 | Eleme nt 3 |

screen width  <= 480px

Element 1

Element 2

Element 3

# @media queries: min-width

## HTML

```
<div class="elem">Element 1</div>
<div class="elem">Element 2</div>
<div class="elem">Element 3</div>
```

## CSS

```
.elem{
    display: inline-block;
}


@media(min-width: 480px){
  .elem{
    display: block;
  }
}
```

## OUTPUT

screen width  <= 480px

| Element 1 | Element 2 |
| Element 3 | |

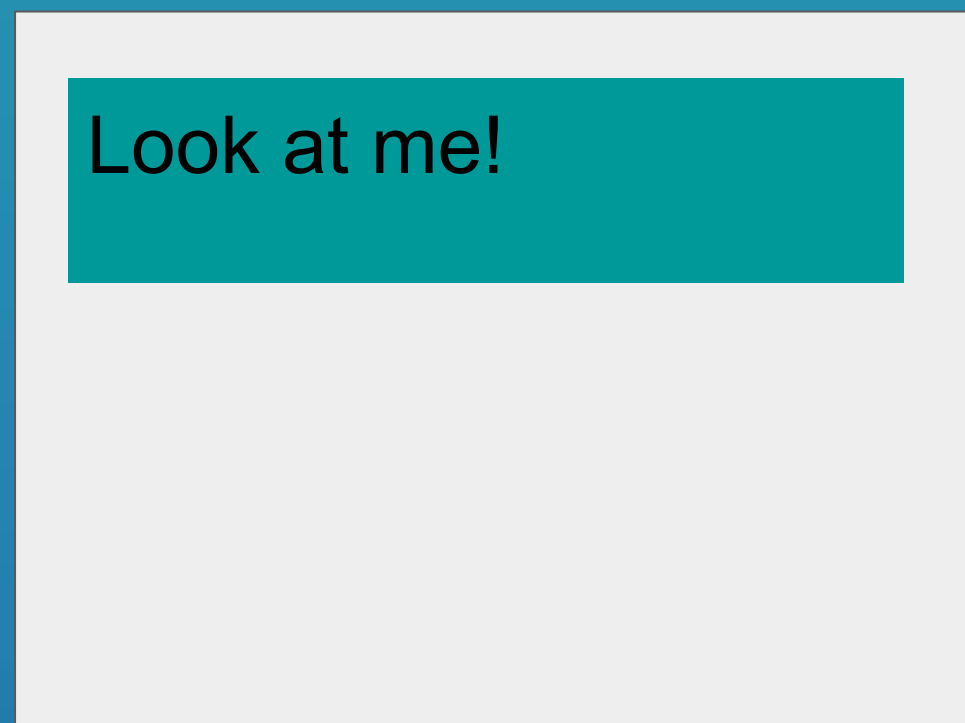screen width  >= 480px

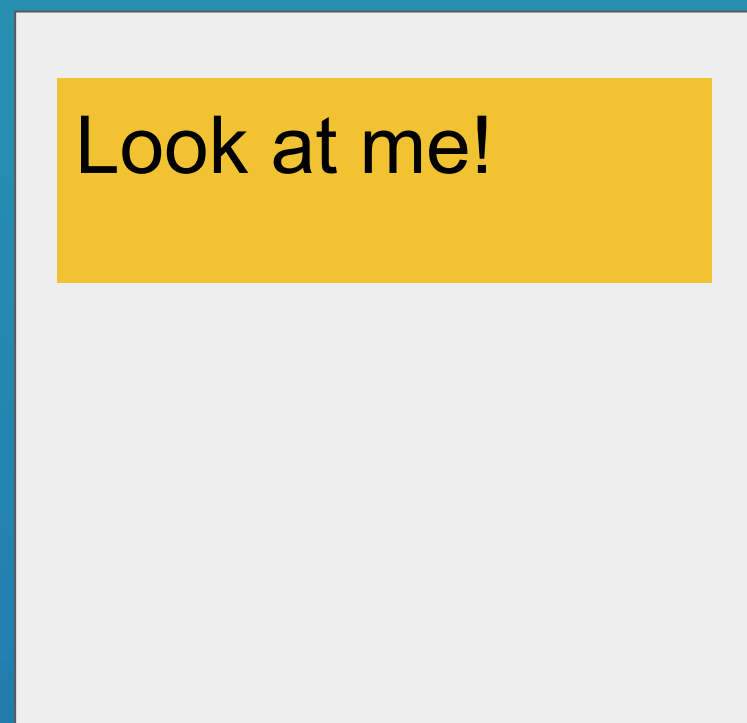| Element 1 |
| Element 2 |
| Element 3 |

# Bonus: Combining min-width and max-width

How can we select a specific range of screen size? Try only changing the background color when the screen size is in between 600px to 800px:
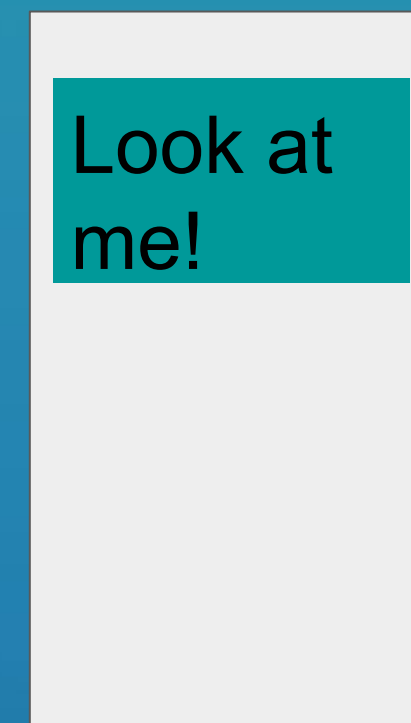
screen width > 800px

Look at me!

600px < screen width <= 800px

Look at me!

screen width < 600px

Look at me!

Hint: You can combine min-width and max-width in the media query :)

# Recap

- **Layout Basics**

  ○ Class and id

  ○ Creating layouts with display and float

  ○ Positioning and z-index

- **Responsive web**

# The End

Thank you for coming!