

HACK YALE

< WEB DEVELOPMENT />

WWW.HACKYALE.COM

HACK YALE

< WEB DEVELOPMENT />

week 1
TIME TO GET PRETTY

WELCOME BACK!

Agenda

- Oscars!
- Common slip-ups
- Filling in some gaps
- Onward to CSS – today is coding intensive!
 - Concept / implementation
 - Stylesheets (the `head` and `link` tags)
 - The DOM and the Box Model

HACK YALE
< FRONT END />

HACKYALE OSCARS
YOU'LL GROW TO LOVE THEM

BEST RÉSUMÉ

HUNTER FORD

BEST COLOR SCHEME



MEREDITH REDICK

BEST TROLL

PRANAV MADDI



MOST DEVELOPED

ERIC PARRIE

CLEANEST PROFILE

KEVIN XU



BEST CHILDHOOD PICS

TIE!

**CHARLEY LOCKE
&
CHARLOTTE PARKER**

COMMON PROBLEMS

- Linking to files on your own computer, not in the folder
- Self closing tags
 - Partly our bad :(
- Not closing tags!



COMMON OOPSIES

LET'S NIP THESE BAD HABITS IN THE BUD

FILLING IN SOME GAPS

- Finishing up HTML: attributes
- Inspecting elements in Chrome
- Concept: Modularity

HACK YALE

< FRONT END />

CSS
THE CONCEPT

SO YOU'VE BUILT A SITE...

But it's ugly. Not to be mean, but by that I mean:

- Aside from images, not much color
- Not much control of where to put the “boxes”
- Spacing and alignment may not be ideal
- Not too responsive: hovers and clicks don’t do anything

THAT'S WHY WE HAVE CSS!

Style specifies how your HTML elements look

- We want to **select** some elements
- And apply a **style** to them
 - like color, size, alignment...

WAYS TO SELECT ELEMENTS

By tag name:

- Very easy, just `p`, `img`, `div` or what have you
- Very rarely used, and often to define a global ‘feel’ for the site
 - For example, make all links red instead of the default blue
 - Make all images have a border
- But what happens when you don’t want *all* of the `p` tags?

WAYS TO SELECT ELEMENTS

By class name:

- A **class** is an attribute given to one or more HTML elements
 - `<p class="blue-text">Some text</p>`
- Many elements may have the same class
- An element may have many classes
 - Just separate with spaces!
 - `<p class="blue underline">More text</p>`

WAYS TO SELECT ELEMENTS

By ID:

- An ID is an attribute given to **only one** element
 - `<p id="my-only-red-paragraph">ELM0</p>`
- Only one element may have a particular ID
- An element may only have one ID

Why IDs and not classes?

PRACTICAL USE CASES

Tag name

- To override defaults or define global styles

Class

- The most widely used, allows for nice element selection

ID

- HackYale discourages using IDs in CSS selectors
 - Some other people do too

HACK YALE

< FRONT END />

CSS
THE IMPLEMENTATION

ENTER, CSS

```
<p> Hello World! </p>
<p> Paragraphs are great! </p>
<p> Totally. </p>
```

Hello World!
Paragraphs are great!
Totally.

ADDING STYLES

```
<p>  
  Hello World!  
</p>  
<p>  
  Paragraphs are great!  
</p>  
<p>  
  Totally.  
</p>
```

+

```
p {  
  color: red;  
}
```



Hello World!
Paragraphs are great!
Totally.

YALE

CSS SYNTAX

```
p {  
    font-size: 18px;  
    color: blue;  
}
```

CSS SYNTAX

```
p {  
    font-size: 18px;  
    color: blue;  
}
```

SELECTOR

CSS SYNTAX

```
p {  
    font-size: 18px;  
    color: blue;  
}
```

SELECTOR

DEFINITION BLOCK

CSS SYNTAX

```
p {  
    font-size: 18px;  
    color: blue;  
}
```

SELECTOR
DEFINITION BLOCK
PROPERTY

CSS SYNTAX

```
p {  
    font-size: 18px;  
    color: blue;  
}
```

SELECTOR
DEFINITION BLOCK
PROPERTY
VALUE

CSS SYNTAX

```
p {  
    font-size: 18px;  
    color: blue;  
}
```

SELECTOR
DEFINITION BLOCK
PROPERTY
VALUE
END OF DEFINITION

THE SOURCE

```
<!DOCTYPE html>
<html>
<head>
    <title>Welcome To HackYale | HTML</title>
</head>
<body>
    <h1>Welcome To HackYale!</h1>
    <h3>This course will focus on front end technologies</h3>
    <ul>
        <li>HTML</li>
        <li>CSS</li>
        <li>JavaScript</li>
    </ul>
    <h2>HTML</h2>
    <p>
        HTML is the <em>content</em> of the web. Without it, none of your favorite sites would exist. You can write links to places like <a href="http://www.google.com/">Google</a>, <a href="http://www.nytimes.com/">The New York Times</a> or anywhere you like!
    </p>
    <p>
        If you go to the 'view' menu in chrome, scroll to 'developer', and click 'view source', you can see the HTML (among other things) that your browser rendered. (shortcut: command + option + u.)
    </p>
    <p>
        This page is <b>just</b> HTML (except for some browser defaults, but we'll get to those later).
    </p>
    <p>
        Next, let's check out <a href="/welcome/css">CSS</a>.
    </p>
</body>
</html>
```

THE BORING RESULT

Welcome To HackYale!

This course will focus on front end technologies

- HTML
- CSS
- JavaScript

HTML

HTML is the *content* of the web. Without it, none of your favorite sites would exist. You can write links to places like [Google](#), [The New York Times](#) or anywhere you like!

If you go to the 'view' menu in chrome, scroll to 'developer', and click 'view source', you can see the HTML (among other things) that your browser rendered. (shortcut: command + option + u.)

This page is **just** HTML (except for some browser defaults, but we'll get to those later).

Next, let's check out [CSS](#).

A DASH OF CSS

```
body {  
    padding: 0;  
    font: 14px "Lucida Grande", Helvetica, Arial, sans-serif;  
    background-color: #ffffff;  
    background-image: -moz-radial-gradient(50% 50%, circle farthest-side, white, #eaeaea 70%);  
    background-image: -webkit-radial-gradient(50% 50%, circle farthest-side, white, #eaeaea 70%);  
    background-image: -o-radial-gradient(50% 50%, circle farthest-side, white, #eaeaea 70%);  
    background-image: -ms-radial-gradient(50% 50%, circle farthest-side, white, #eaeaea 70%);  
    background-image: radial-gradient(50% 50%, circle farthest-side, white, #eaeaea 70%);  
    margin: 0;  
}  
  
a {  
    color: #00B7FF;  
    text-decoration: none;  
}  
  
a:hover {  
    text-decoration: underline;  
}  
  
ul, li {  
    list-style-type: none;  
}  
  
p {  
    margin-left: 40px;  
}  
  
.emph {  
    font-size: 20px;  
    color: #20AA20;  
    font-family: "Zapf Chancery", Parkavenue, cursive;  
}
```

THE LESS BORING RESULT

Welcome To HackYale!

This course will focus on front end technologies

HTML
CSS
JavaScript

CSS

CSS is the *style* of the web.

With CSS, you can color, arrange, and generally style your page. In addition, you can add small amounts of dynamic activity, like the underline you'll see as the mouse hovers over the "JavaScript" link.

Next, let's check out [JavaScript](#).

SO HOW DO I ADD CSS?

ADDING CSS

We've seen one way to add it: <style> tags.

However,

- Style tags break the principle of **modularity**.
- Style tags don't allow us to stay **DRY** (Don't Repeat Yourself)

So...we use stylesheets!

LINKING TO A STYLESHEET

Writing all our styles in a **separate** file.

- Usually called “style.css”
- We can include this style on all our pages without copy-pasting or rewriting code
- Allows us to stay modular

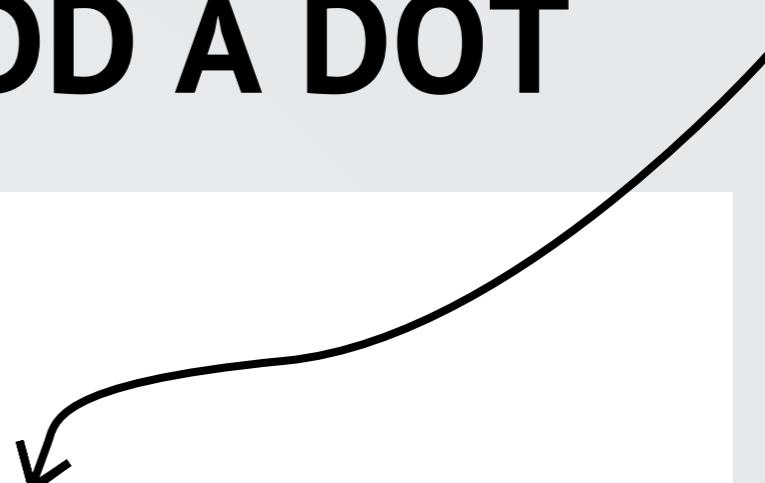
So how do we add it..?

THE LINK TAG

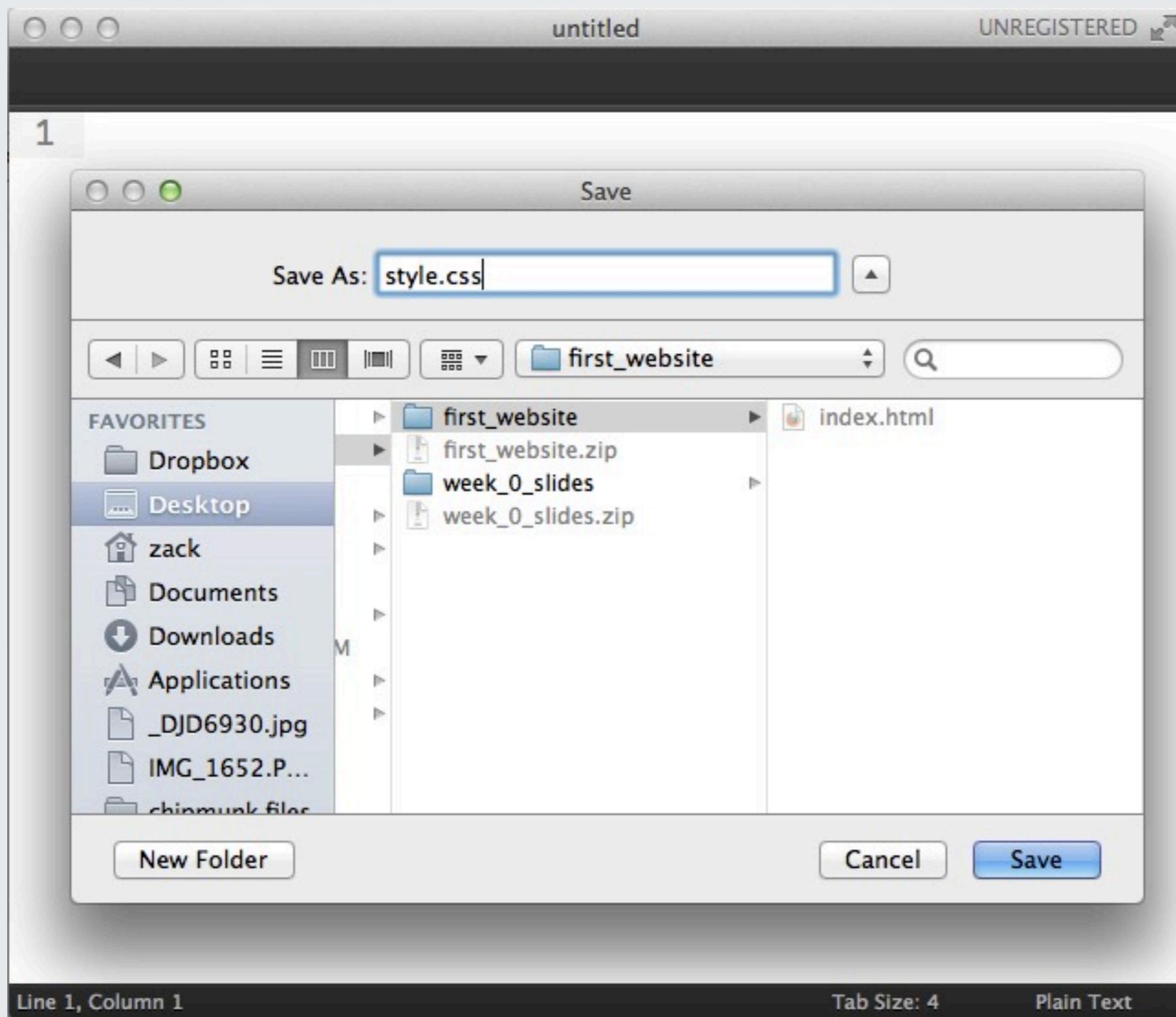
```
<html>
<head>
    <!-- some HTML -->
    <link rel="stylesheet" href="/style.css">
    <!-- some more HTML -->
</head>
<body>
    <!-- even more HTML -->
</body>
</html>
```

WE'RE GOING TO ADD A DOT

```
<html>
  <head>
    <!-- some HTML -->
    <link rel="stylesheet" href="./style.css">
    <!-- some more HTML -->
  </head>
  <body>
    <!-- even more HTML -->
  </body>
</html>
```



SAVE CSS FILE IN PROJECT DIRECTORY



ADD A STYLE DECLARATION



A screenshot of a Mac OS X application window titled "style.css". The window contains the following CSS code:

```
1 p {  
2     color: red;  
3 }
```

The code is numbered from 1 to 8 on the left side. The word "color" is highlighted in blue, and "red" is highlighted in green.

WOOHOO!

Welcome To HackYale!

This course will focus on front end technologies

- HTML
- CSS
- JavaScript

HTML

HTML is the *content* of the web. Without it, none of your favorite sites would exist. You can write links to places like [Google](#), [The New York Times](#) or anywhere you like!

If you go to the 'view' menu in chrome, scroll to 'developer', and click 'view source', you can see the HTML (among other things) that your browser rendered. (shortcut: command + option + u.)

This page is just HTML (except for some browser defaults, but we'll get to those later).

Next, let's check out [CSS](#).

WWW.HACKYALE.COM



WE'VE ADDED TO OUR WORKFLOW

STYLESHEETS ARE BIG STEP!

THE REAL DEAL



BACK TO SELECTORS FOR A SEC

- Tag name: just use the tag, e.g `p`
- Class: prefix a `.`
 - e.g. `".my-class"`
- ID: prefix a `#`
 - e.g. #my-id"
 - But try not to use IDs; stick to classes

COMMON STYLES

- **color, background-color**
 - Takes names (red, white, blue), rgb or hex values
- **text-decoration**
 - “none” or “underline”
- **width, height, font-size**
 - Specify as a percentage, or pixel value (10% or 10px)
- **border**
 - Size style color
 - E.g: border: 1px double red;

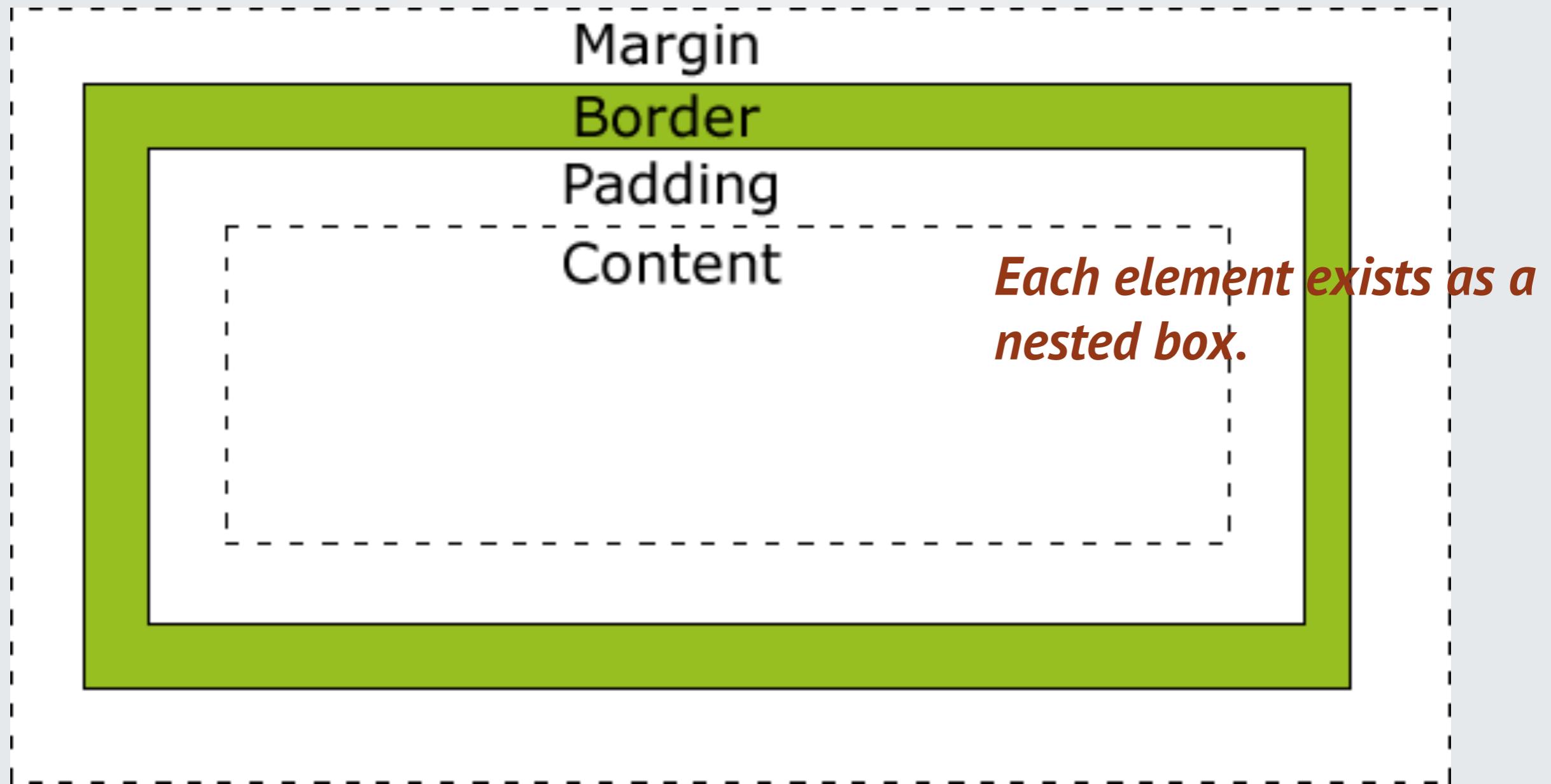


LIVE EXAMPLE!

AS ALWAYS, FOLLOW ALONG

THE BOX MODEL

THE BOX MODEL





DISPLAY
DISPLAY: INLINE;



DISPLAY VALUES: BLOCK

display: block;

- Element takes up the full available width
- Begins on a new line, forces following content onto a new line

Some text I wrote here.

<div>My div is here!</div>

And more text continues here.

Some text I wrote here

My div is here!

And more text continues here.

DISPLAY VALUES: INLINE

display: inline;

- Element takes up only the width of its content
- Remains in the flow of the document (does not start a new line)

```
<p>I've got a paragraph of  
text here. In the middle, I'd  
like to put some <span>frilly  
text</span> so that my  
students think I'm cute.</p>
```

I've got a paragraph of text here. In the middle, I'd like to put some *frilly text* so that my students think I'm cute.

DISPLAY VALUES: NONE

display: none;

- Element is not rendered in the browser
- Removed from the flow of the document
 - i.e. it does not affect the positioning of other elements

```
<p>I've got a paragraph of  
text here. In the middle, I'd  
like to put some <span  
style="display:none;">frilly  
text</span> so that my  
students think I'm cute.</p>
```

I've got a paragraph of text here. In the middle, I'd like to put some so that my students think I'm cute.

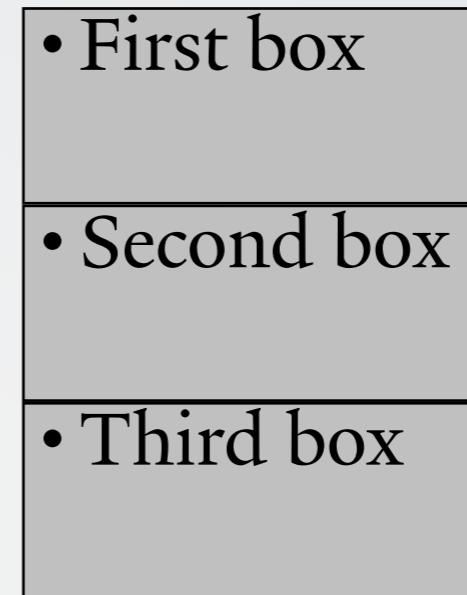
DISPLAY VALUES: INLINE-BLOCK

`display: inline-block;`

- Rendered like an inline element (only takes up needed width, doesn't disrupt document flow)
- Allows us to set block-display properties like width, height, and top and bottom padding / margin

DISPLAY VALUES: PRACTICE

```
<ul id="my-boxes">  
  <li>First box</li>  
  <li>Second box</li>  
  <li>Third box</li>  
</ul>  
  
#my-boxes li {  
  display: block; /* default */  
  width: 175px;  
  height: 75px;  
  background: gray;  
  border: 1px solid black;  
}
```



DISPLAY VALUES: PRACTICE

```
<ul id="my-boxes">  
  <li>First box</li>  
  <li>Second box</li>  
  <li>Third box</li>  
</ul>
```

```
#my-boxes li {  
  display: inline;  
  width: 175px;  
  height: 75px;  
  background: gray;  
  border: 1px solid black;  
}
```

- First box
- Second box
- Third box

DISPLAY VALUES: PRACTICE

```
<ul id="my-boxes">  
  <li>First box</li>  
  <li>Second box</li>  
  <li>Third box</li>  
</ul>
```

```
#my-boxes li {  
  display: inline-block;  
  width: 175px;  
  height: 75px;  
  background: gray;  
  border: 1px solid black;  
}
```

- First box
- Second box
- Third box

EACH ELEMENT TYPE HAS A DEFAULT DISPLAY PROPERTY

Defaults

- usually ‘*inline*’ or ‘*block*’
 - *inline*
 - *span*, *a*, *em*, *img*, most text modifiers
 - *block*
 - *div*, *p*, *ul*, *form*, etc.
-



The position property

THE POSITION PROPERTY SPECIFIES A WAY
FOR AN ELEMENT TO POSITION ITSELF
WITH REGARD TO THE BROWSER WINDOW,
THE PARENT ELEMENT, OR RELATIVE TO
SIBLING ELEMENTS.

POSITION: STATIC

Render the element relative to its neighbor with no offset

- The default setting (so you never need to declare this in CSS)

POSITION: FIXED

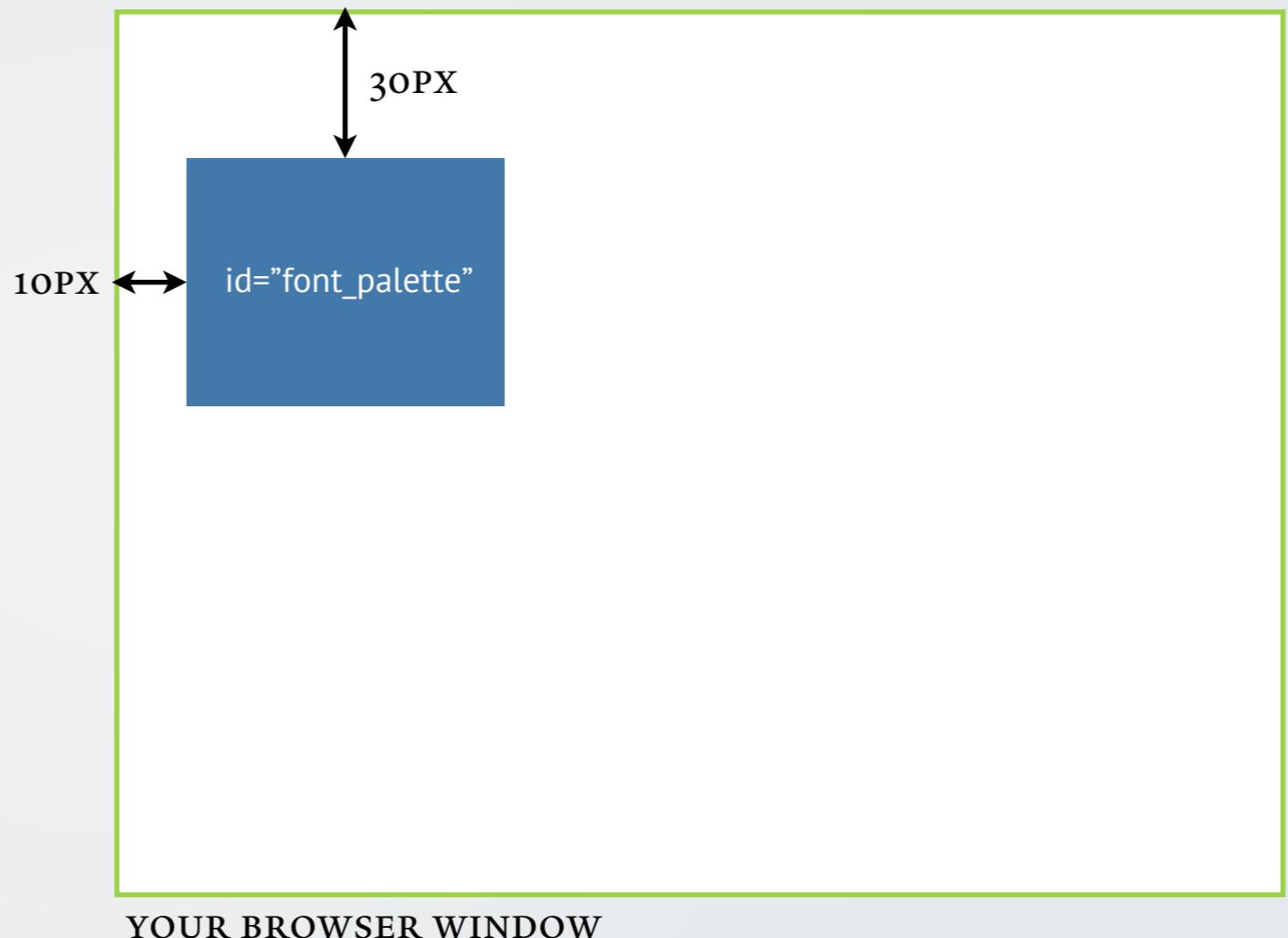
Sets the element to be rendered at a fixed location in the browser window, regardless of page scrolling

- top, left, bottom, and right properties tell the element where to position itself in the browser window

POSITION: FIXED

```
2 #font_palette {  
3     position: fixed;  
4     top: 30px;  
5     left: 10px;  
6 }
```

Take the element with **id="font_palette"** and render it 30px from the top of the window and 10px from the left of the window



POSITION: ABSOLUTE

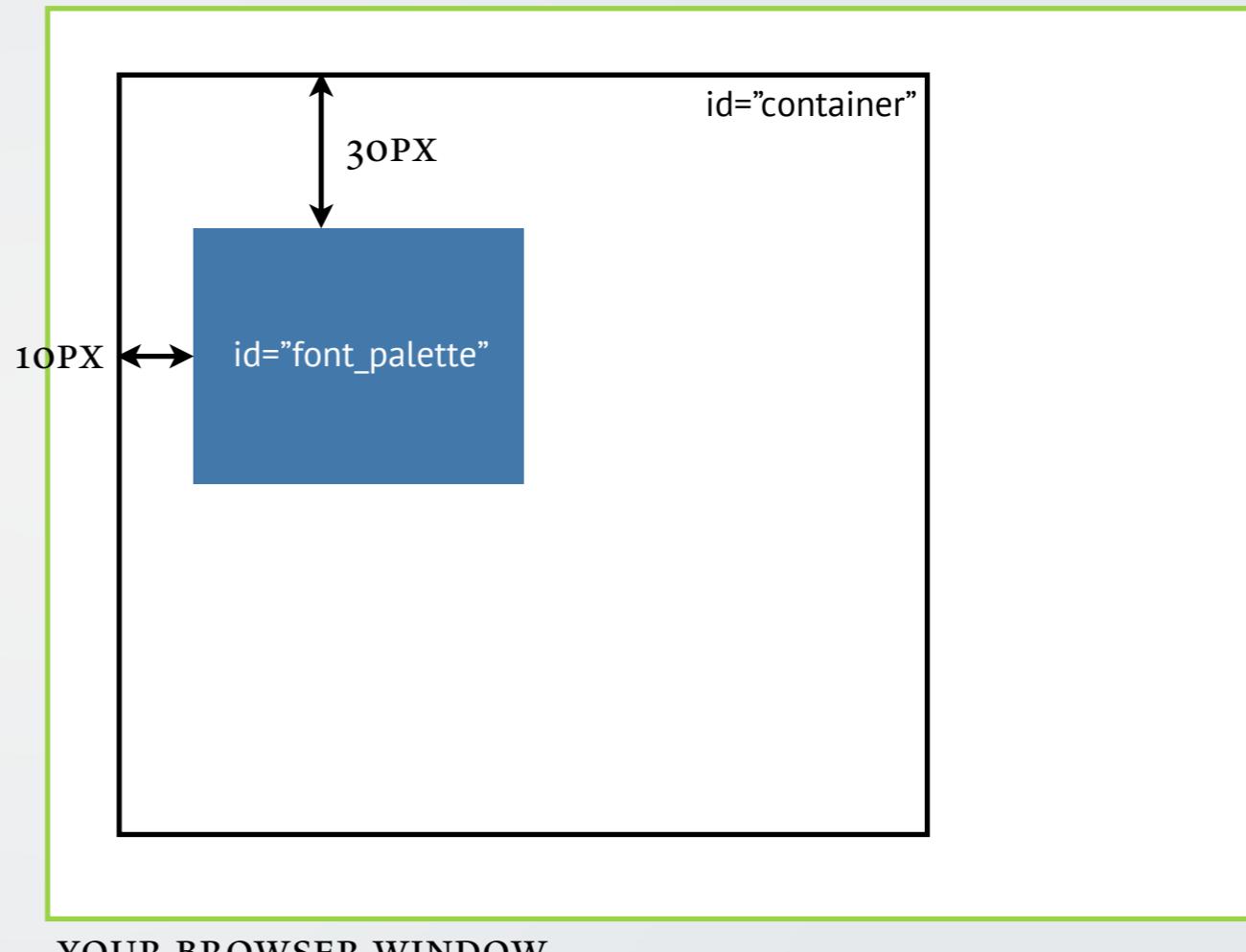
Sets the element to be rendered at a specific location in the parent element

- top, left, bottom, and right properties tell the element where to position itself in the parent element

POSITION: ABSOLUTE

```
2 #container #font_palette {  
3     position: absolute;  
4     top: 30px;  
5     left: 10px;  
6 }
```

Take the element with **id="font_palette"** and render it 30px from the top and 10px from the left of its parent (**id="container"**)



POSITION: RELATIVE

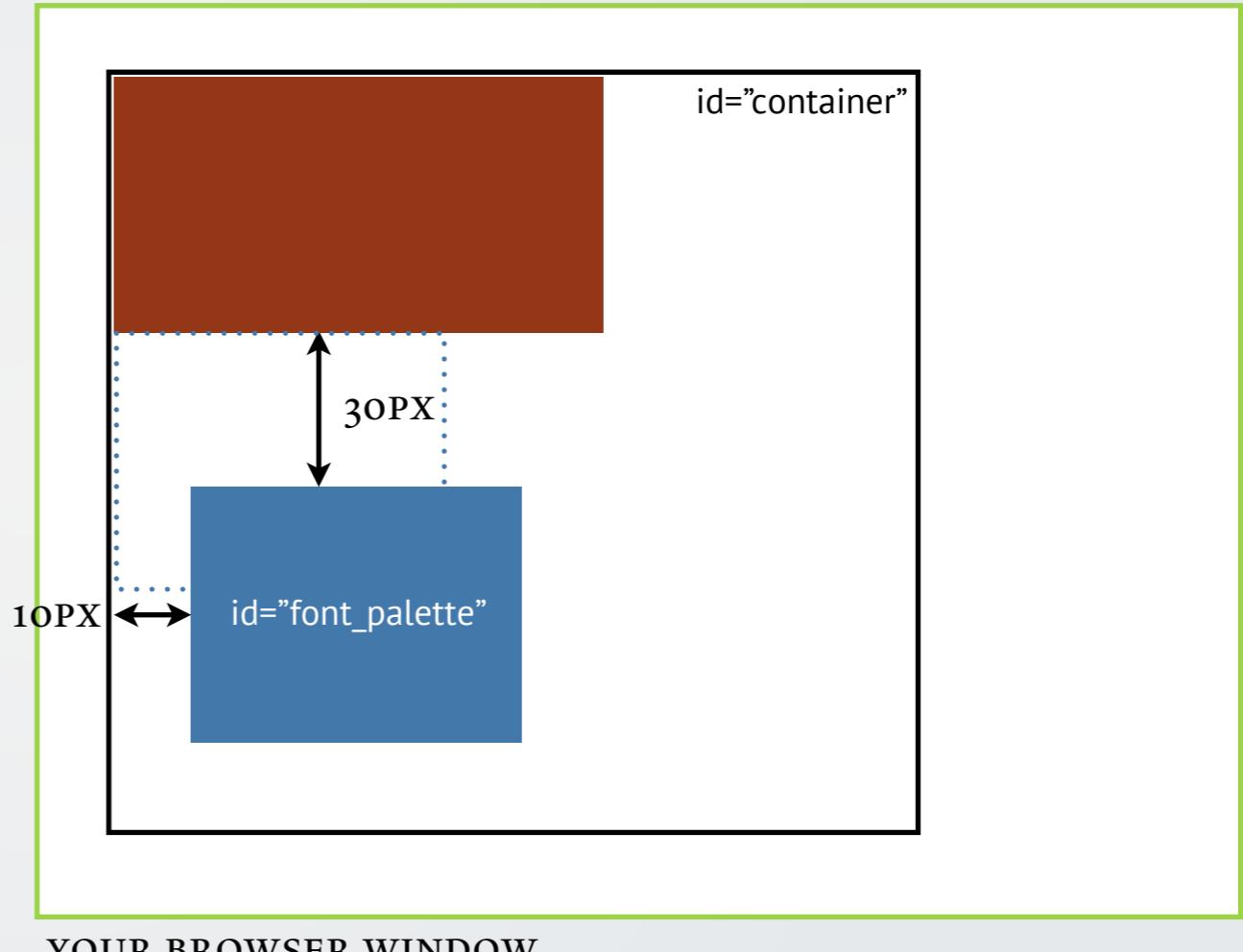
Tells the element how to position itself relative to neighboring sibling elements (i.e. elements with the same parent)

- top, left, bottom, and right properties tell the element where to position itself relative to where it would normally be

POSITION: ABSOLUTE

```
2 #font_palette {  
3     position: relative;  
4     top: 30px;  
5     left: 10px;  
6 }
```

Take the element with **id="font_palette"** and render it 30px from the top and 10px from the left of where it would normally be



HOMEWORK

Adopt your previous website and make it pretty!

- At least 50 lines of HTML – use div's, p's, img's, a's...you know a lot of tags now, and should be comfortable using them!
- At least 100 lines of CSS. This is your chance to practice, crack open your design sense!
- **Everything must be well-formatted:** That means tabs and newlines where they belong. No ugly code!

NUGGETS

- Change Sublime Theme
 - Preferences...Color Scheme...
- Designer's inspiration
 - <http://line25.com/>
 - <http://www.thebestdesigns.com/>
- Designer's resources
 - <http://subtlepatterns.com/>
 - <http://www.freeiconsweb.com/>



QUESTIONS EVEN GOOGLE CAN'T ANSWER?

TEAM@HACKYALE.COM

WWW.HACKYALE.COM