HACK YALE

< WEB DEVELOPMENT />

WWW.HACKYALE.COM

HACK‹YALE›

< WEB DEVELOPMENT />

week_3

BUILDING OUR WIREFRAME

AGENDA

Today

- Oscars
- The Oldest College Daily
- Sample good solution
- Sample amazing solution
- Questions / CSS review
- Beginning JavaScript

HACK YALE

HACKYALE OSCARS

WEEK 3

CUTEST

HACK YALE

SOPHIA SANCHEZ

EARLY BIRD

JEREMY LIU

HACK YALE

BEST FAVICON

HACK YALE

DIEGO TORRES

HACK YALE

MOST NUTRITIOUS

HACK YALE

TIE!

ANNA YOUNG
&
JESSICA YUAN

HACK YALE

MOST SPEC-TACULAR

HACK YALE

LIZ ASAI

MOST BUMPIN’

HACK YALE

JACK MULROW

HACK YALE

MOST EDUCATIONAL

HACK YALE

TIE!

HACK YALE

CHARLOTTE PARKER
&
CHANDRIKA SRIVASTAVA

HACK YALE

BEST CAPTIONS

HACK YALE

KEVIN XU

HACK YALE

MOST LAUNCH-READY

HACK YALE

ERIC PARRIE

HACK YALE

DISPLAY PROPERTY

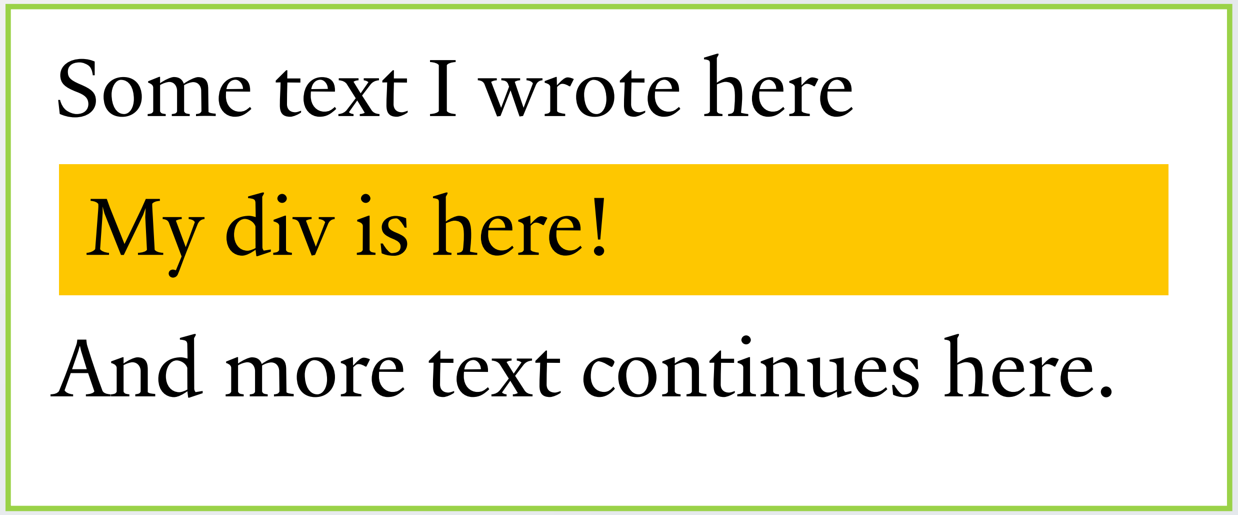
DISPLAY: SICKNASTY;

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!

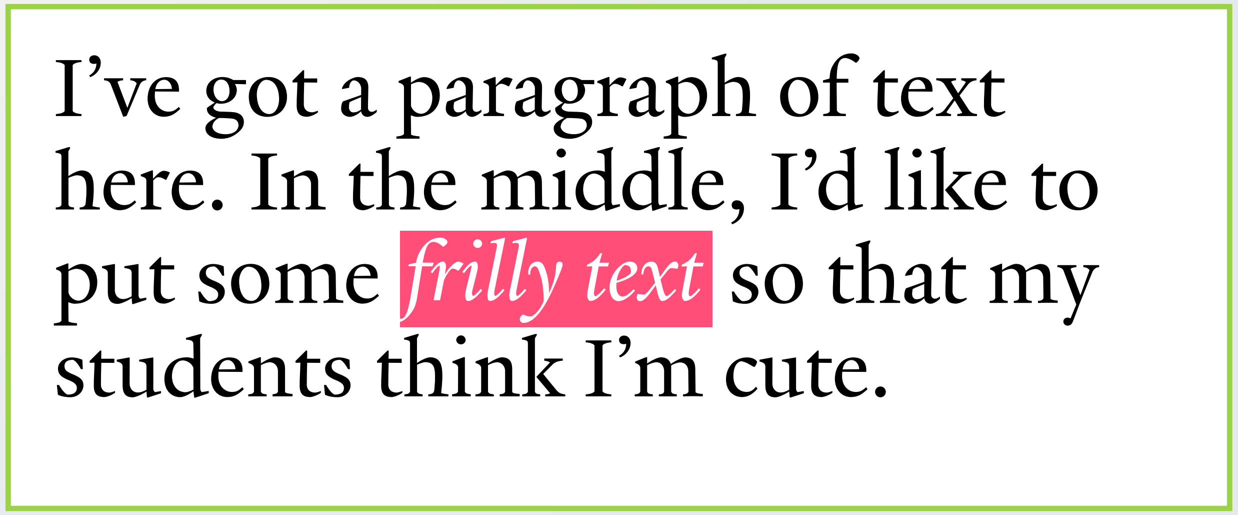
And more text continues here.

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>
```



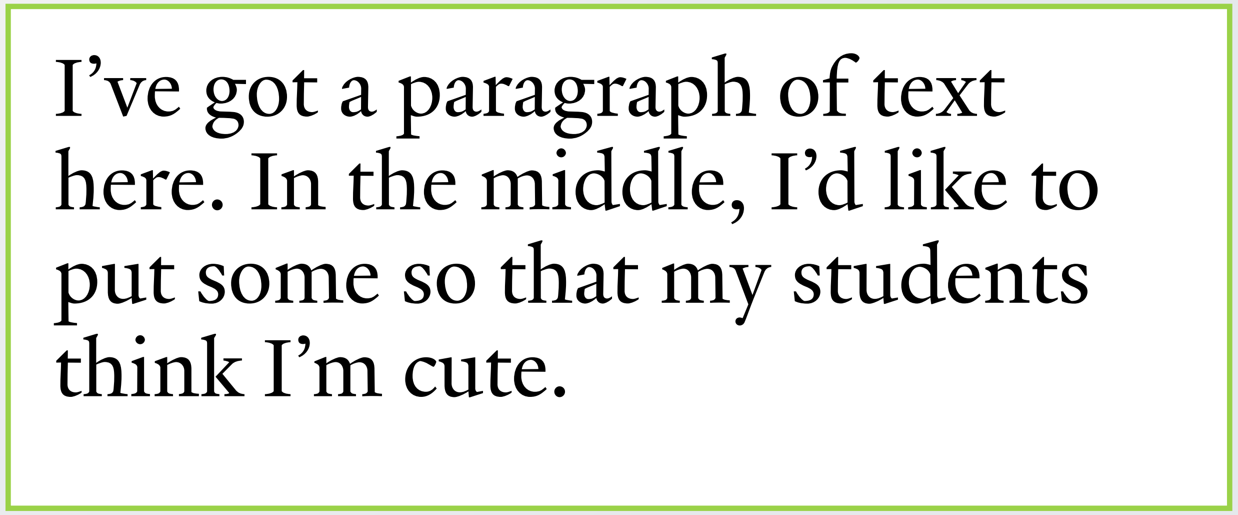
I’ve got a paragraph of text here. In the middle, I’d like to put some *frilly text* so that my students think I’m cute.

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

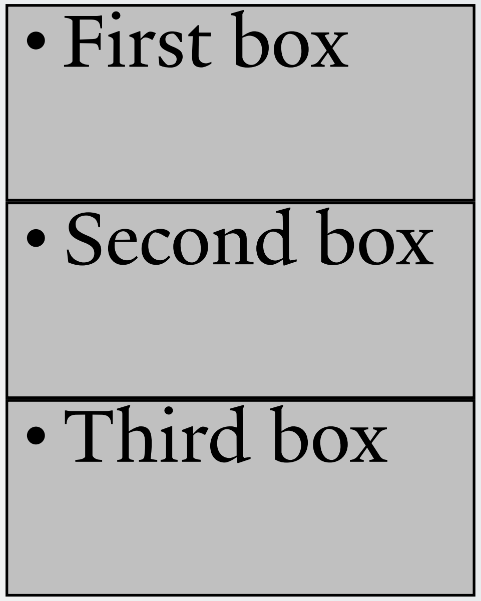
```
display: inline-block;
```

- 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

HACK YALE

DISPLAY VALUES: PRACTICE

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

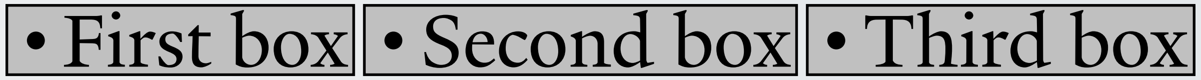


- First box
- Second box
- Third box

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

DISPLAY VALUES: PRACTICE

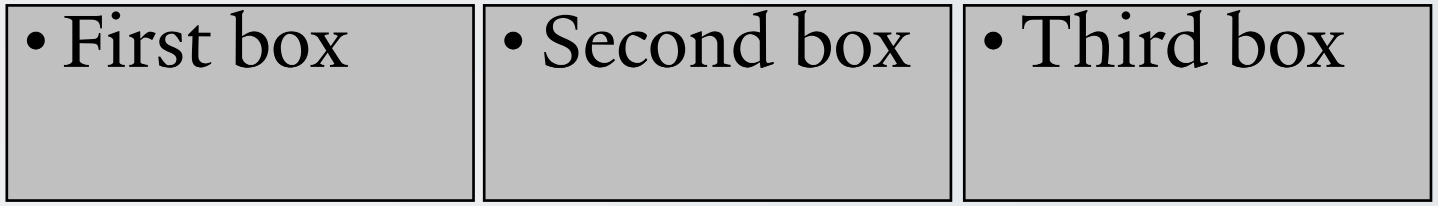
```
<ul id=”my-boxes”>
    <li>First box</li>
    <li>Second box</li>
    <li>Third box</li>
</ul>

#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>
```



- First box
- Second box
- Third box

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

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.

HACK YALE

❮ The position property ❯

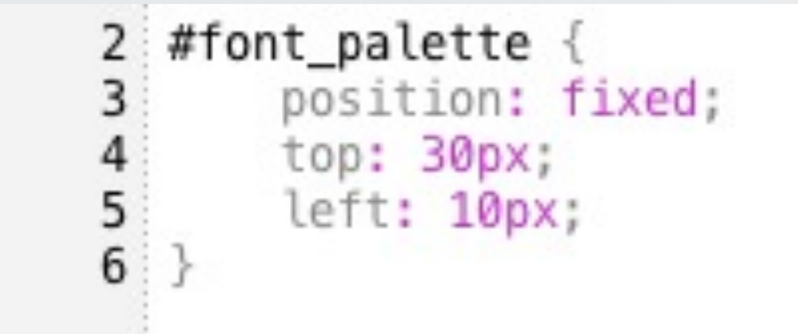
THE POSITION PROPERTY SPECIFIES A WAY FOR AN ELEMENT TO POSITION ITSELF WITH REGARD TO THE BROWSER WINDOW, THE PARENT ELEMENT, OR RELATIVE TO SIBLING ELEMENTS.

POSITION: FIXED

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

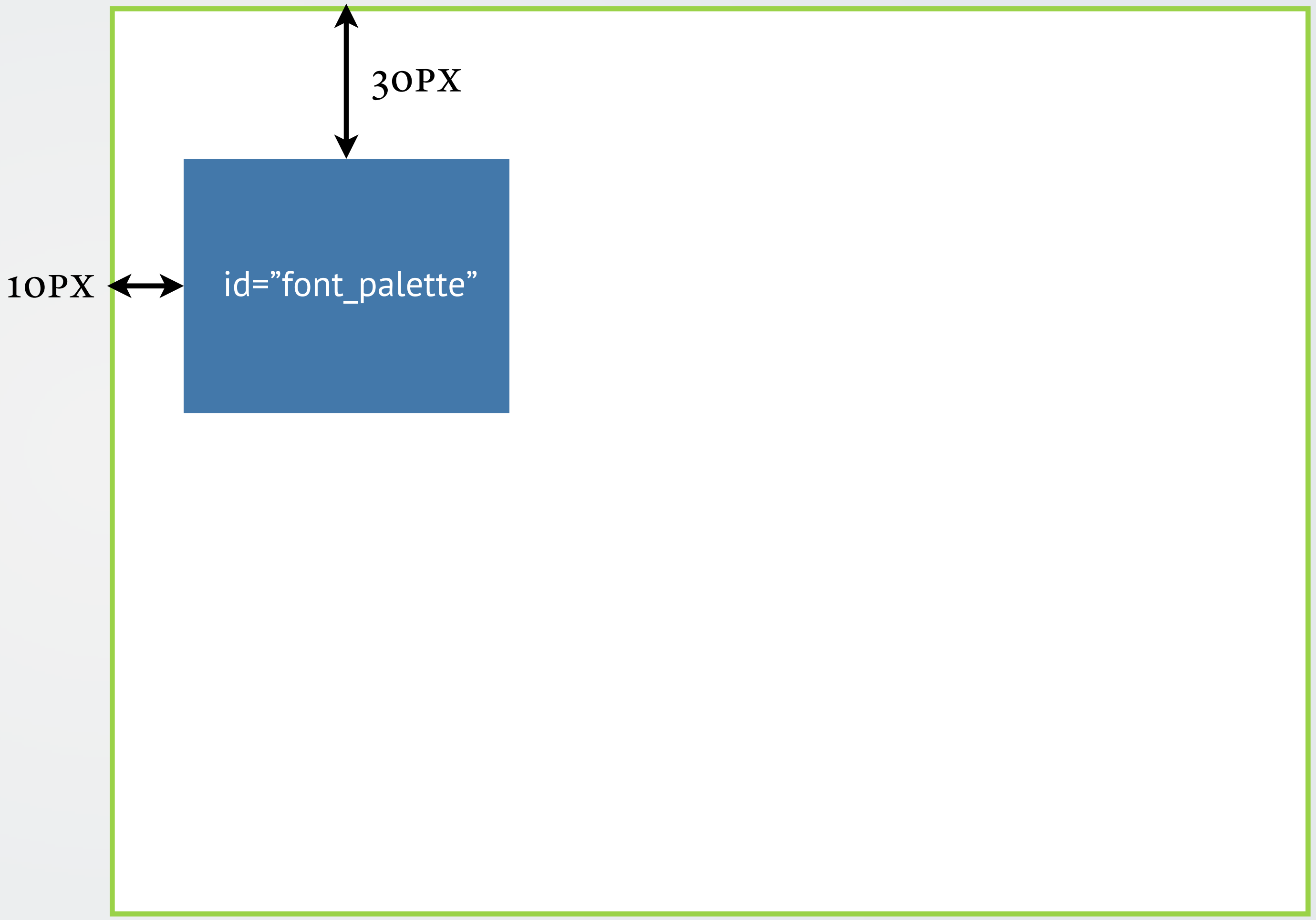
- top, left, bottom, and right properties tell the element where to position itself in the browser window

POSITION: FIXED



```
#font_palette {
    position: fixed;
    top: 30px;
    left: 10px;
}
```

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



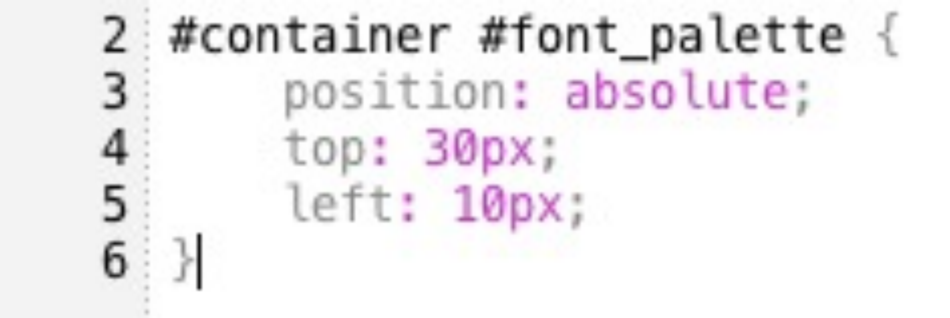
YOUR BROWSER WINDOW

POSITION: ABSOLUTE

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

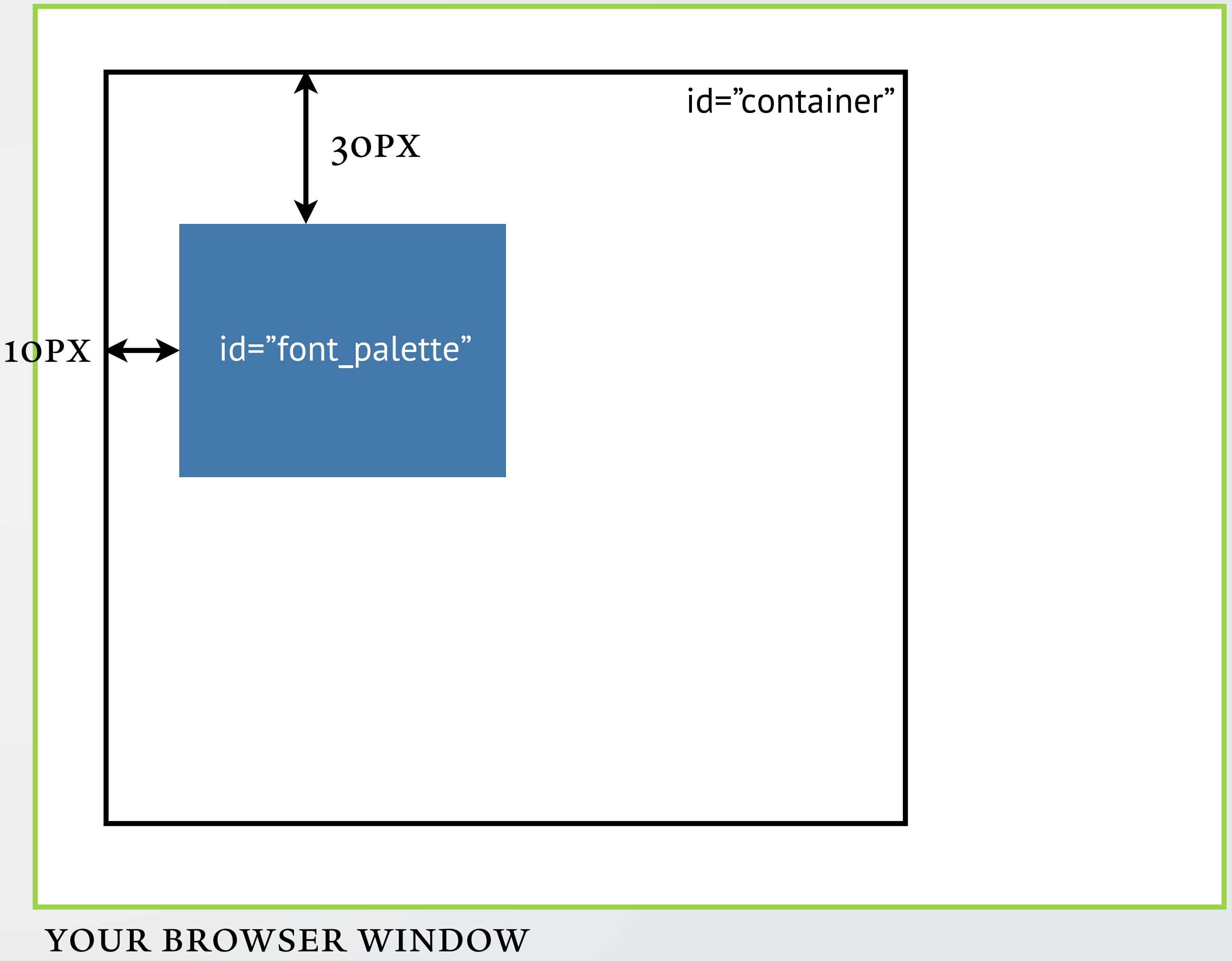
- top, left, bottom, and right properties tell the element where to position itself in the parent element
- **The parent element must not be position static!!**
  - Quick fix: make it position: relative;

POSITION: ABSOLUTE



```
#container #font_palette {
    position: absolute;
    top: 30px;
    left: 10px;
}
```

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

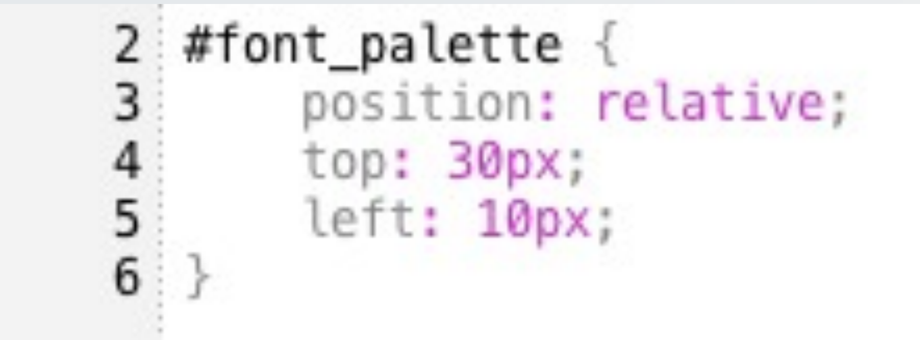


POSITION: RELATIVE

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

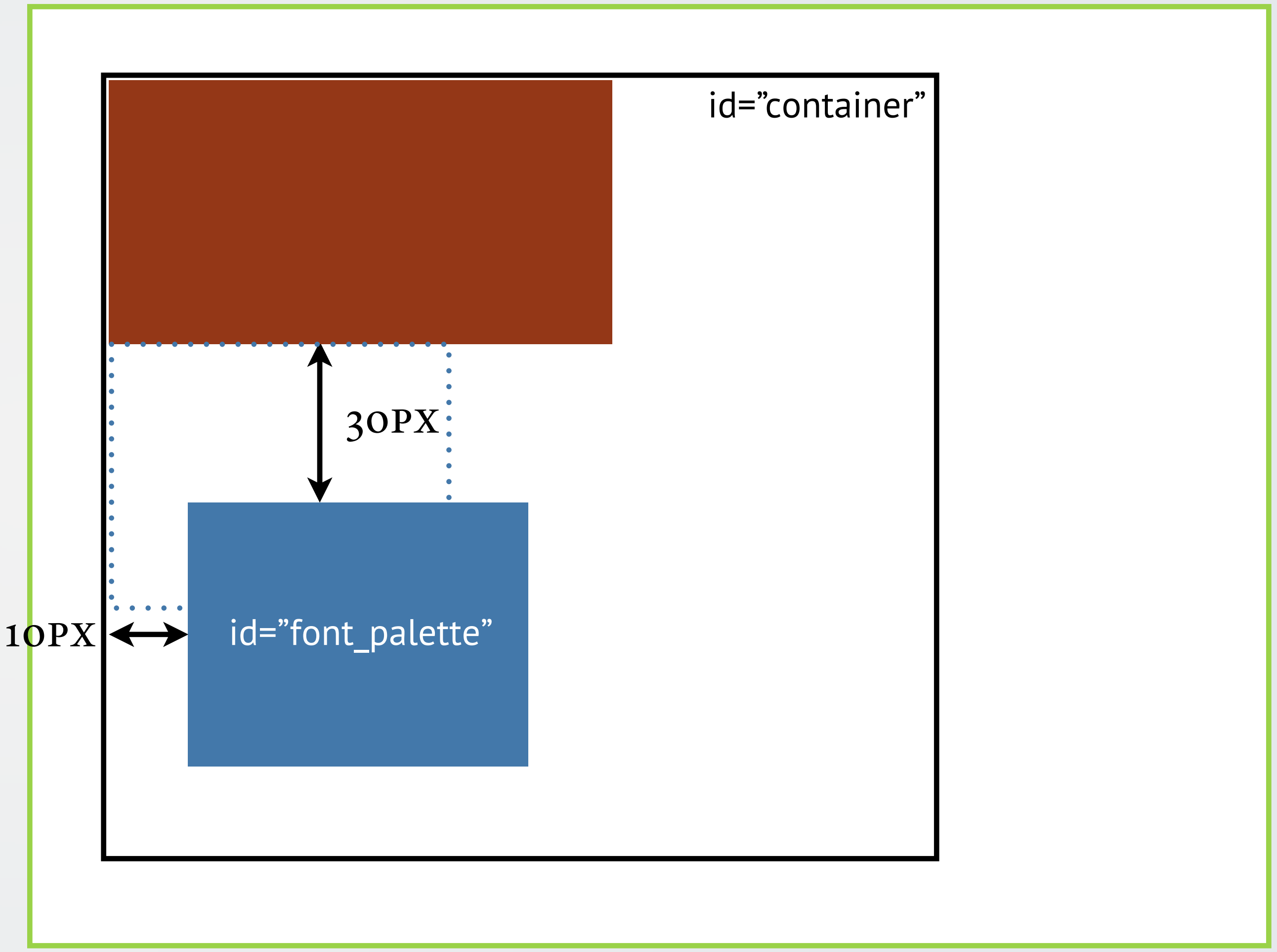
- top, left, bottom, and right properties tell the element where to position itself relative to where it would normally be

POSITION: RELATIVE



```
#font_palette {
    position: relative;
    top: 30px;
    left: 10px;
}
```

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



YOUR BROWSER WINDOW

POSITION: STATIC

Render the element relative to its neighbor with no offset

- The default setting (so you never need to declare this in CSS)
- You can imagine this as “Tetris”, where the gravity is up instead of down,
    - And you’re holding the left key

The float property

FLOAT: LEFT;

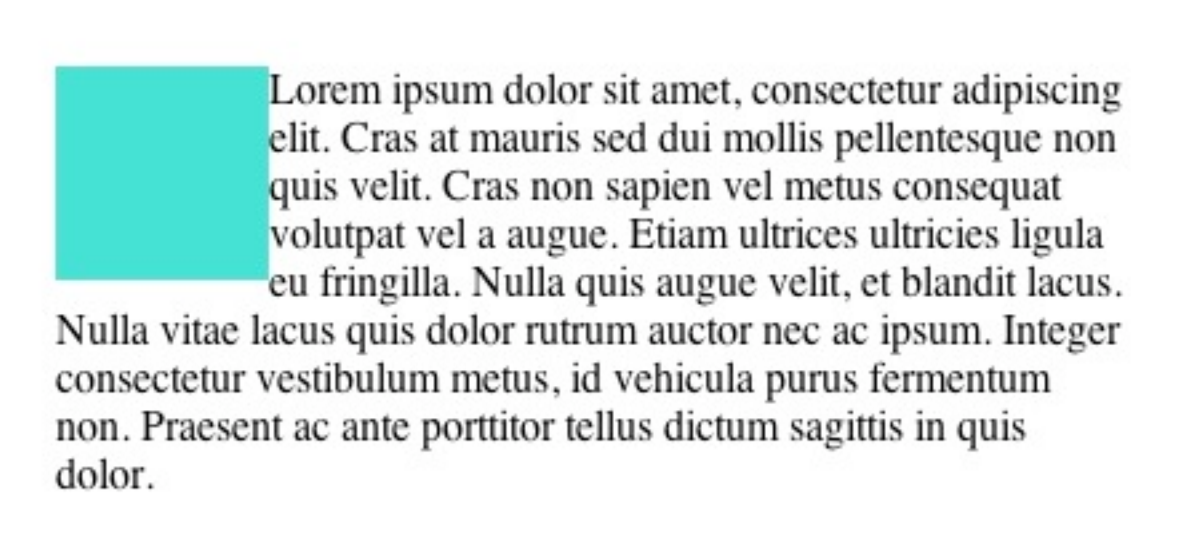
THE FLOAT PROPERTY

Causes element to ‘float’ toward one side of its parent

- Four values: left, right, none, and inherit
- IMPORTANT: the width of the element collapses to only the needed width of the content (think *inline*!)
- Multiple floated siblings will line up side by side

FLOAT: LEFT;

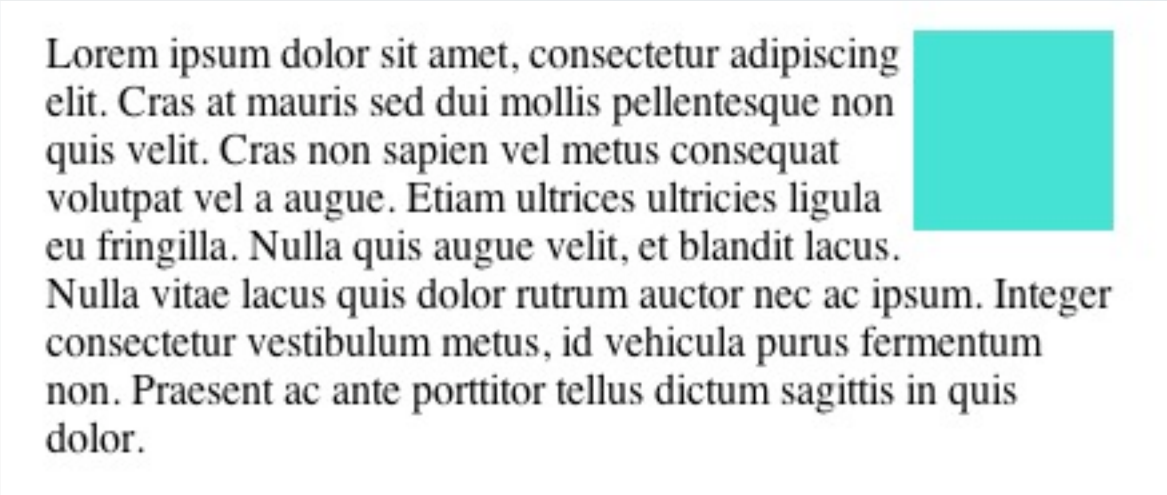
```
.floatleft {
  display: block;
  float: left;
  width: 75px;
  height: 75px;
  background: turquoise;
}

<p>
  <span class="floatleft"></span>
  Lorem ipsum dolor sit amet,
  consectetur adipiscing elit ...
</p>
```



Lorem ipsum dolor sit amet, consectetur adipiscing elit. Cras at mauris sed dui mollis pellentesque non quis velit. Cras non sapien vel metus consequat volutpat vel a augue. Etiam ultrices ultricies ligula eu fringilla. Nulla quis augue velit, et blandit lacus. Nulla vitae lacus quis dolor rutrum auctor nec ac ipsum. Integer consectetur vestibulum metus, id vehicula purus fermentum non. Praesent ac ante porttitor tellus dictum sagittis in quis dolor.

FLOAT: RIGHT;

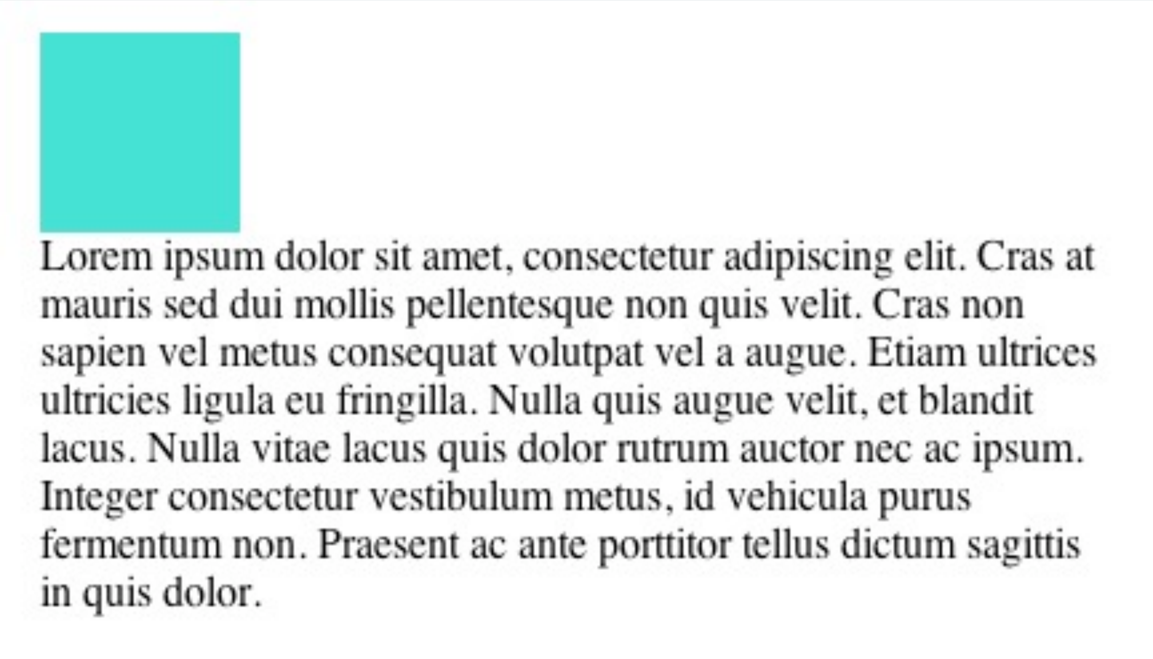
```
.floatright {
  display: block;
  float: right;
  width: 75px;
  height: 75px;
  background: turquoise;
}

<p>
  <span class="floatright"></span>
  Lorem ipsum dolor sit amet,
  consectetur adipiscing elit ...
</p>
```



Lorem ipsum dolor sit amet, consectetur adipiscing elit. Cras at mauris sed dui mollis pellentesque non quis velit. Cras non sapien vel metus consequat volutpat vel a augue. Etiam ultrices ultricies ligula eu fringilla. Nulla quis augue velit, et blandit lacus. Nulla vitae lacus quis dolor rutrum auctor nec ac ipsum. Integer consectetur vestibulum metus, id vehicula purus fermentum non. Praesent ac ante porttitor tellus dictum sagittis in quis dolor.

CLEAR: BOTH;

```
.floatleft {
  display: block;
  float: left;
  width: 75px;
  height: 75px;
  background: turquoise;
}
p { clear: both; }

<div>
  <span class="floatleft"></span>
  <p>Lorem ipsum dolor sit amet,
  consectetur adipiscing
  elit ...</p>
</div>
```



Lorem ipsum dolor sit amet, consectetur adipiscing elit. Cras at mauris sed dui mollis pellentesque non quis velit. Cras non sapien vel metus consequat volutpat vel a augue. Etiam ultrices ultricies ligula eu fringilla. Nulla quis augue velit, et blandit lacus. Nulla vitae lacus quis dolor rutrum auctor nec ac ipsum. Integer consectetur vestibulum metus, id vehicula purus fermentum non. Praesent ac ante porttitor tellus dictum sagittis in quis dolor.

Pseudo-classes

A:HOVER {COLOR:GREEN;}

LINKS

Probably the most common use for a pseudo-class

- `a:visited`
  - Styling applied to links that have been visited by the user
- `a:hover`
  - Styling applied while the mouse is hovered over the link
- `a:active`
  - Styling applied while the user’s mouse button is pressed down on the link
- Order matters! Always apply your CSS styles in this order: a, a:visited, a:hover, a:active

HOMEWORK

HACK YALE

HOMEWORK

JS Practice
http://eloquentjavascript.net/
http://www.codecademy.com/#!/exercises/0
http://cloudchill.in/a/misc/javascript_intro/index.html

ASSIGNMENT
Make a website about different JavaScript skills you've learned, and upload it to cloudchill.in. This assignment is more open-ended than the past two, so we're looking to see you experiment more and take some chances. Here was my first homework assignment, from way back when I first took HackYale last year:
http://cloudchill.in/a/misc/zacks_old_hw1/

HACK YALE

< FRONT END />

WWW.HACKYALE.COM