A: There is one crucial set of rules we haven't learned yet...

block vs **inline** display

What is HTML?

HTML (**H**ypert**e**xt **M**arkup **L**anguage)

- Describes the **content** and **structure** of a web page
- Made up of building blocks called **elements**.

```
<p>
  HTML is <em>awesome!!!</em>
  <img src="puppy.png" />
</p>
```

And there are 3 basic types.

Types of HTML elements

Each HTML element is categorized by the HTML spec into one of three-ish categories:

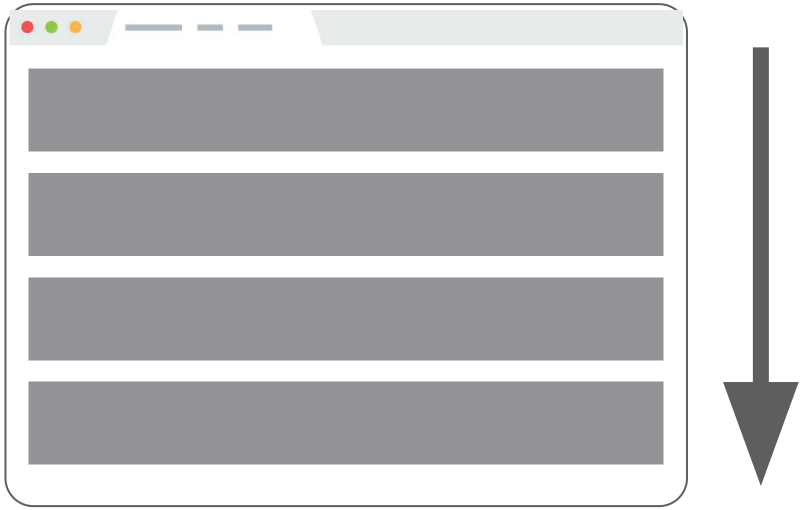
1. **block:** large blocks of content, has height and width
   `<p>`, **`<h1>`**, `<blockquote>`, `<ol>`, `<ul>`, `<table>`
2. **inline:** small amount of content, no height or width
   `<a>`, `<em>`, **`<strong>`**,`<br>`
   a. **inline block:** inline content with height and width
      `<img>`
3. metadata: information about the page, usually not visible
   `<title>`, `<meta>`

Block elements

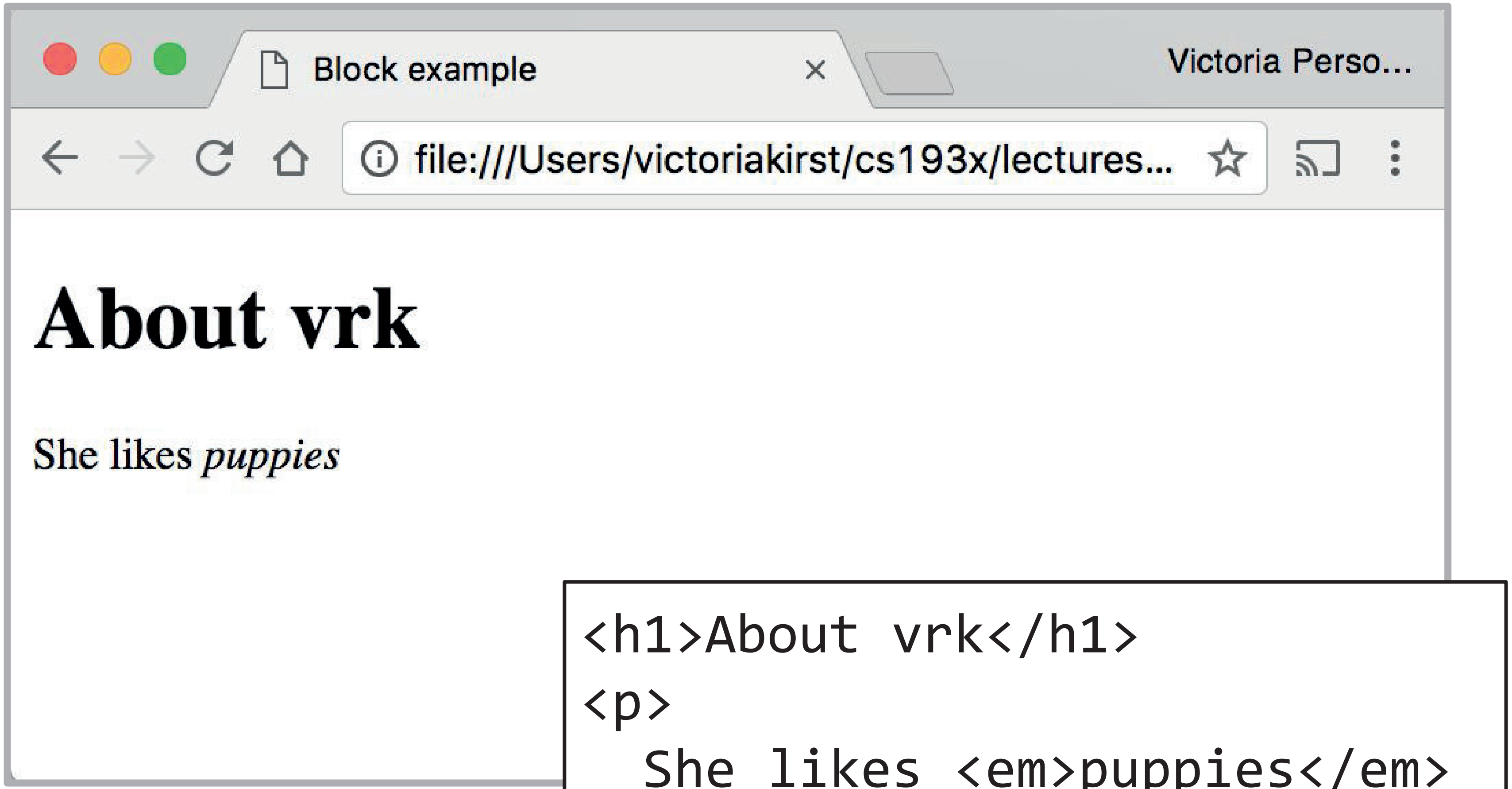
Examples:

`<p>, <h1>, <blockquote>, <ol>, <ul>, <table>`

- Take up the full width of the page (**flows top to bottom**)
- Have a `height` and `width`
- Can have block or inline elements as children



Example: Block



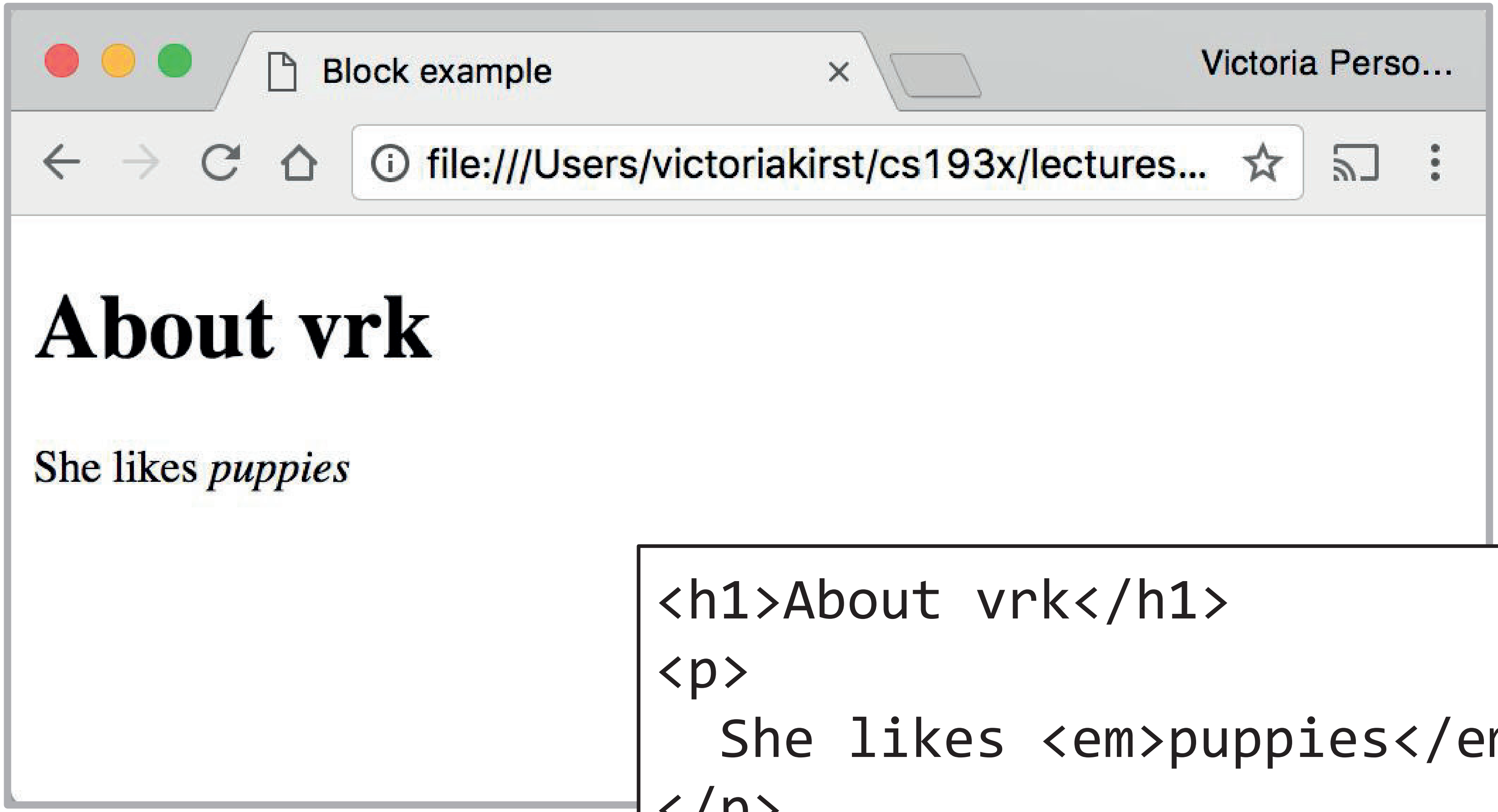
About vrk

She likes *puppies*

```
<h1>About vrk</h1>
<p>
  She likes <em>puppies</em>
</p>
```

Q: What does this look like in the browser?

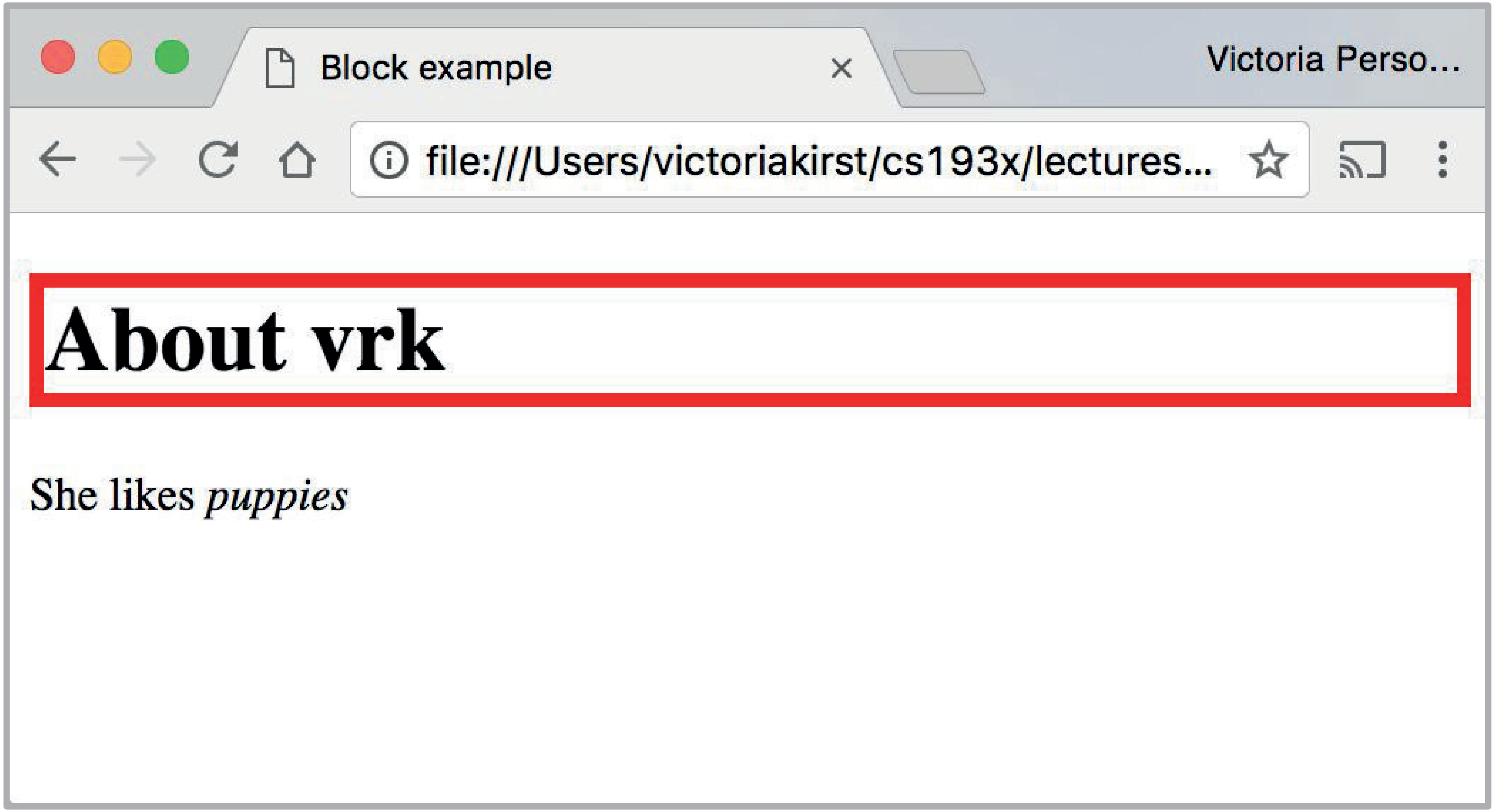
```
h1 {
  border: 5px solid red;
}
```



About vrk

She likes *puppies*

```
<h1>About vrk</h1>
<p>
  She likes <em>puppies</em>
</p>
```



About vrk

She likes *puppies*

(Codepen)

Block-level:

extends the full width of the page

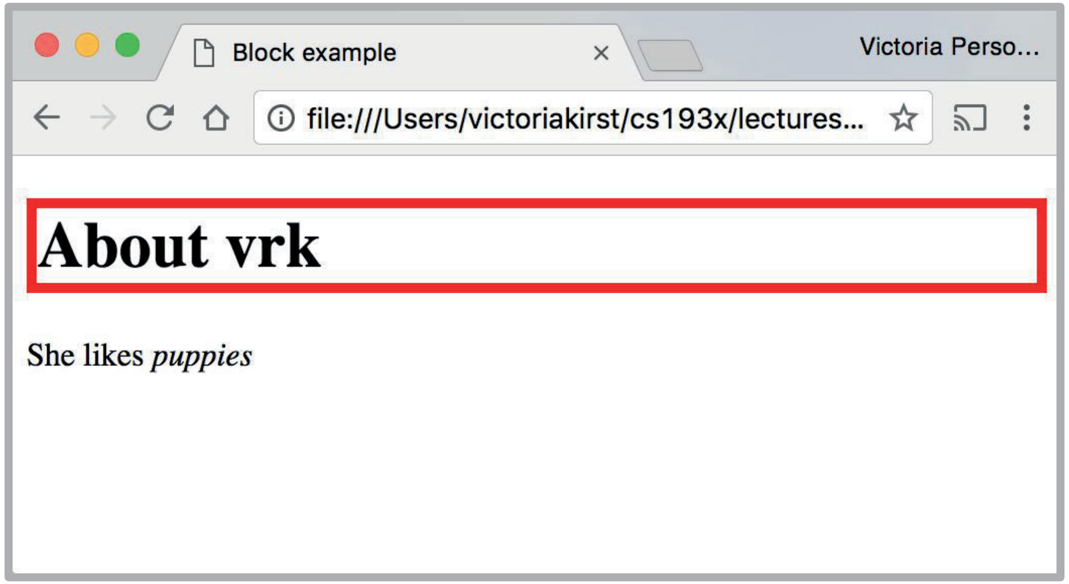
```
h1 {
  border: 5px solid red;
}
```

```
<h1>About vrk</h1>
<p>
  She likes <em>puppies</em>
</p>
```

<h1> is block-level, so it extends the full width of the page by default

Note how block-level elements (**h1**, **p**) flow top to bottom

See: Codepen

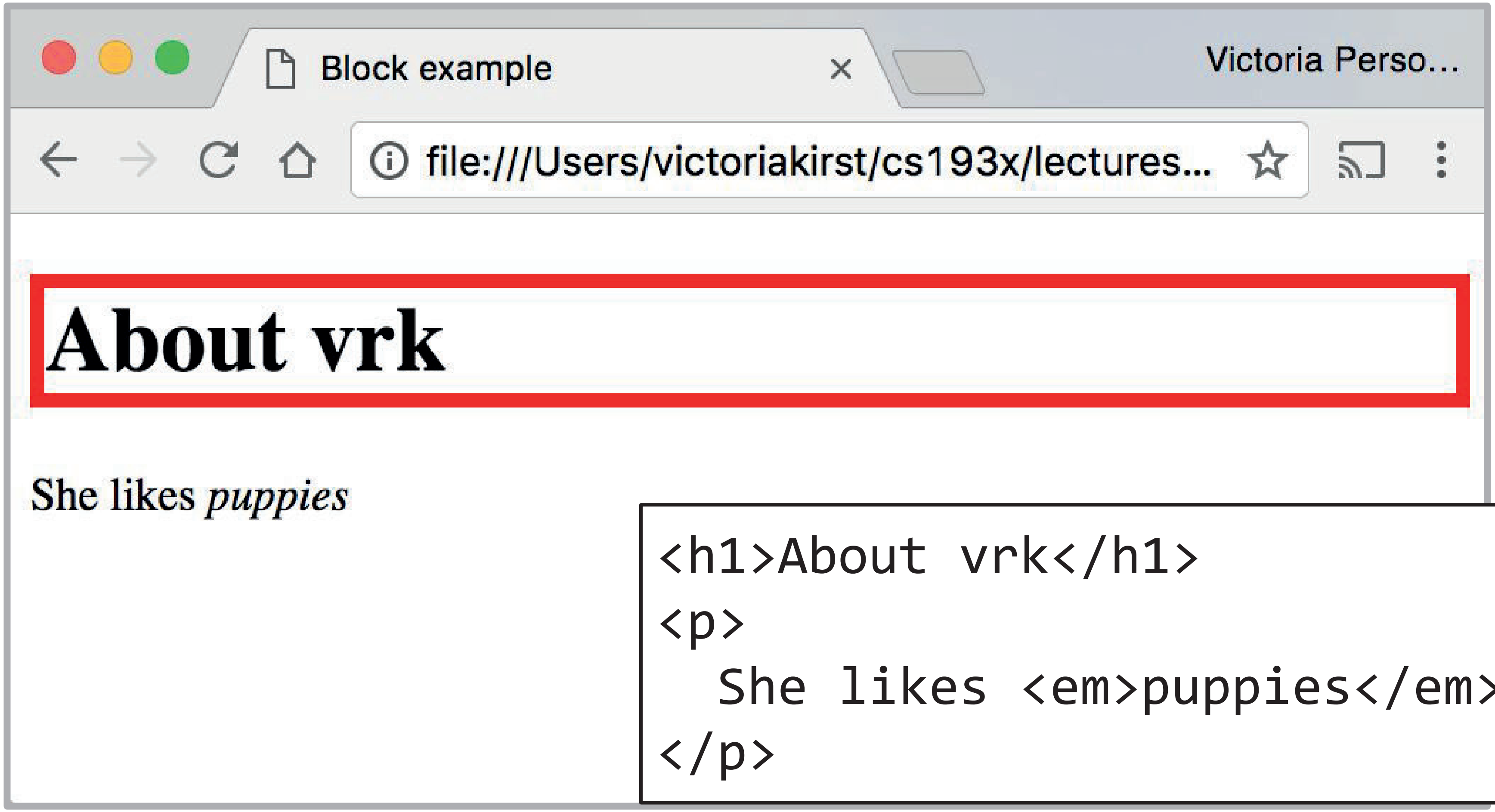


About vrk

She likes *puppies*

Q: What does this look like in the browser?

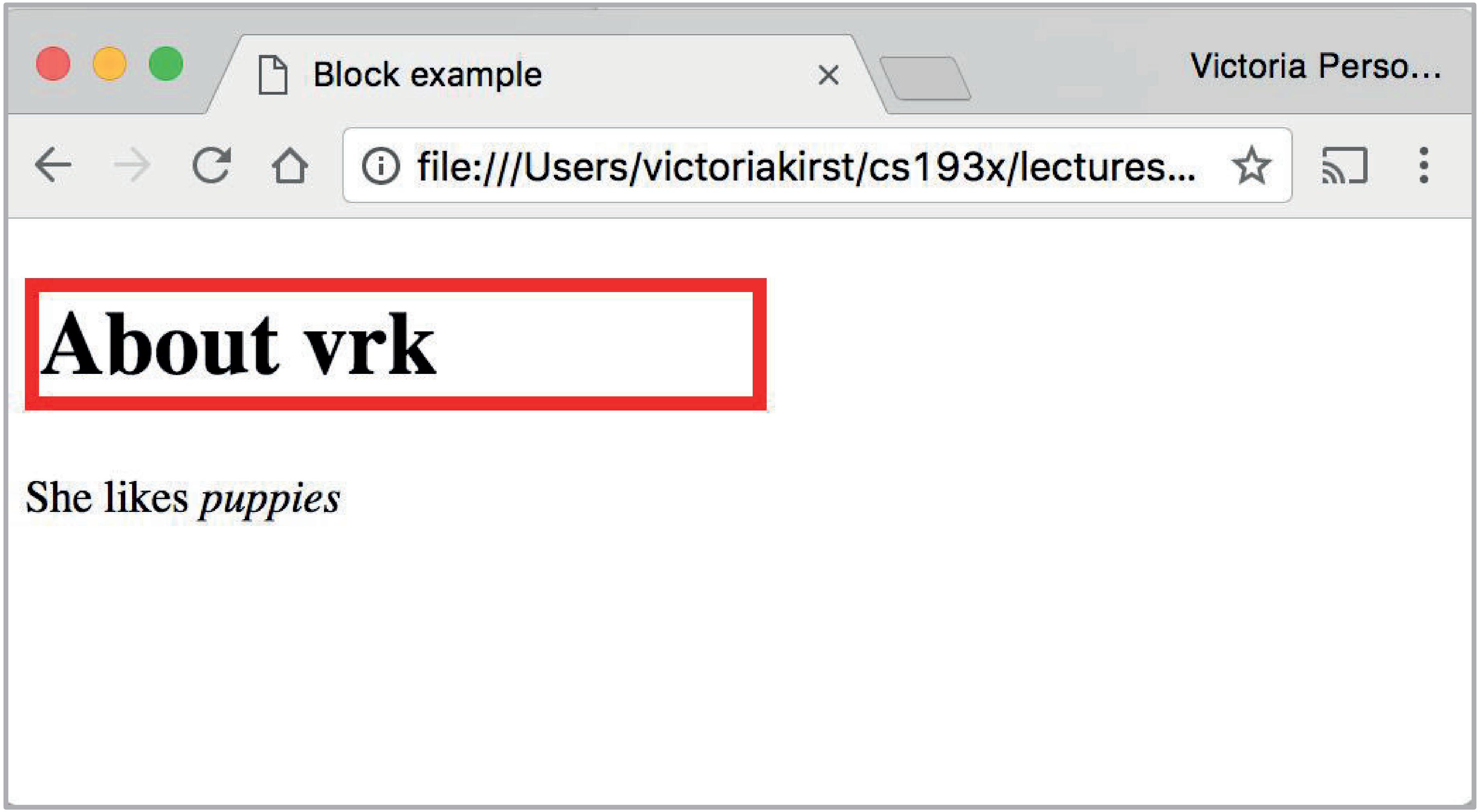
```
h1 {
  border: 5px solid red;
  width: 50%;
}
```



About vrk

She likes *puppies*

```
<h1>About vrk</h1>
<p>
  She likes <em>puppies</em>
</p>
```



About vrk

She likes *puppies*

(Codepen)

Block-level

width can be modified

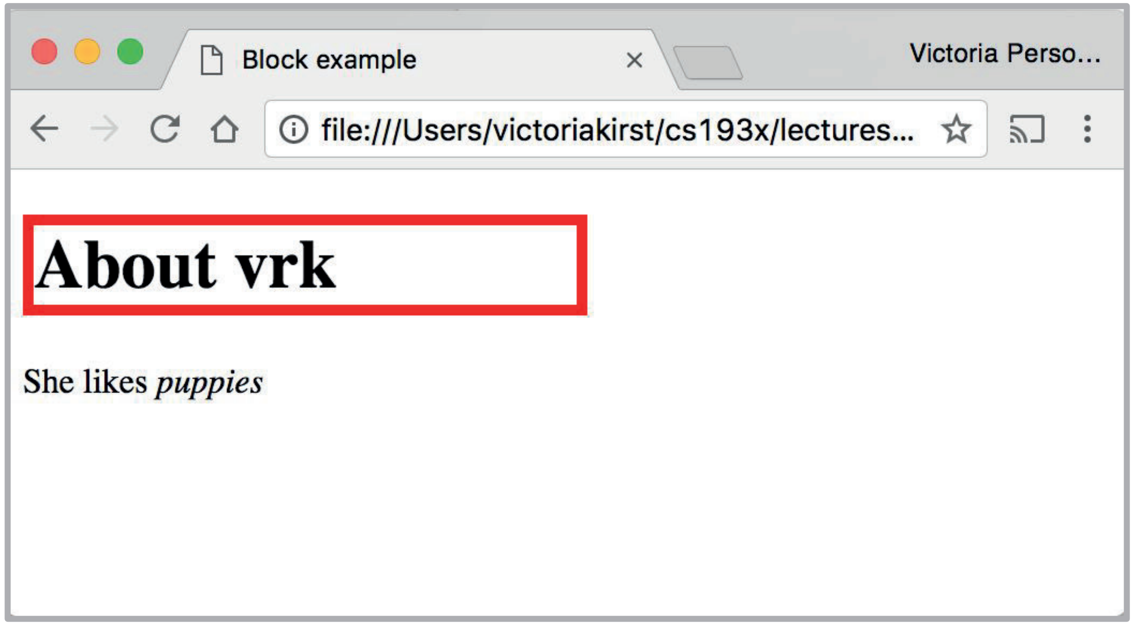
```
h1 {
  border: 5px solid red;
  width: 50%;
}
```

```
<h1>About vrk</h1>
<p>
  She likes <em>puppies</em>
</p>
```

<h1> is block-level, so its **width can** be modified

Block-level elements still flow top to bottom

See: Codepen



About vrk

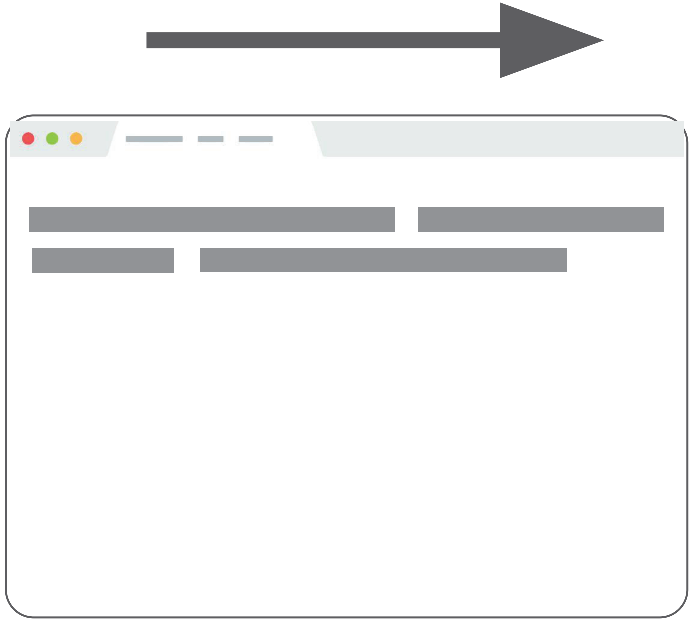
She likes *puppies*

Inline elements

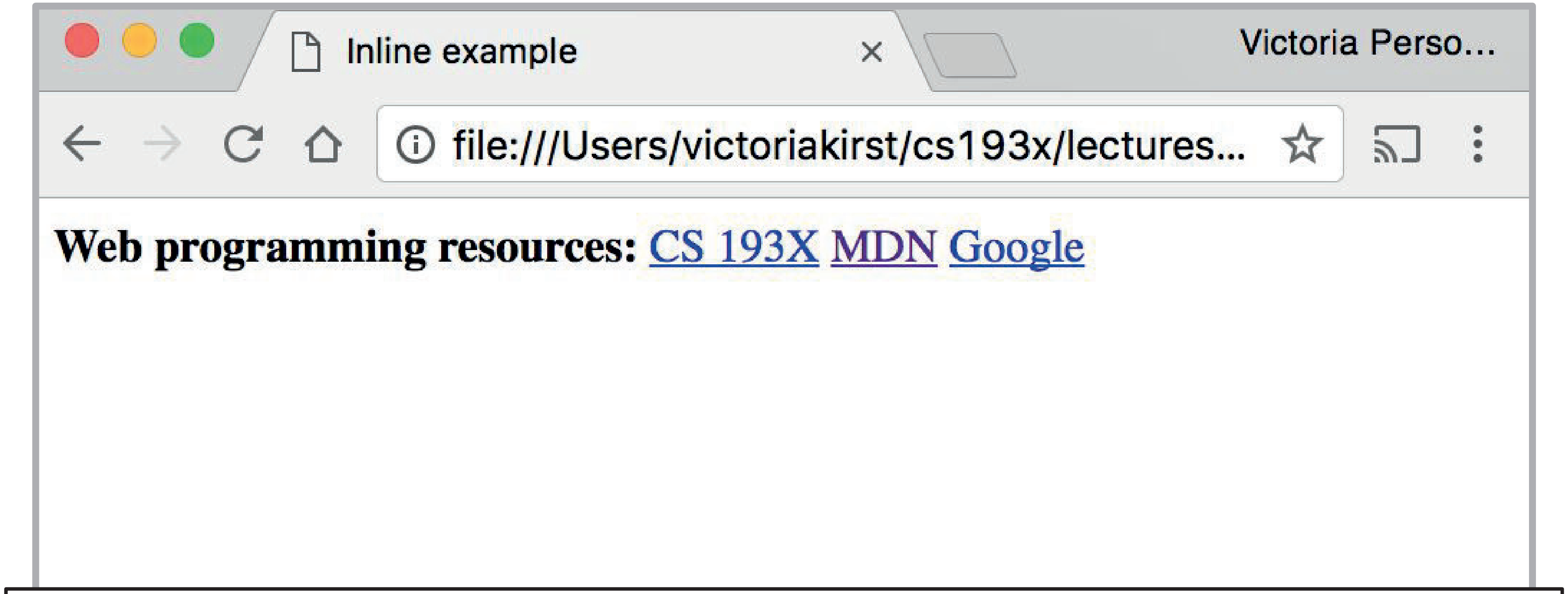
Examples:

`<a>, <em>, <strong>, <br>`

- Take up only as much width as needed (flows left to right)
- **Cannot** have `height` and `width`
- **Cannot** have a block element child
- **Cannot** be positioned (i.e. CSS properties like `float` and `position` do not apply to inline elements)
    - ○ Must position **its containing block element** instead



Example: Inline

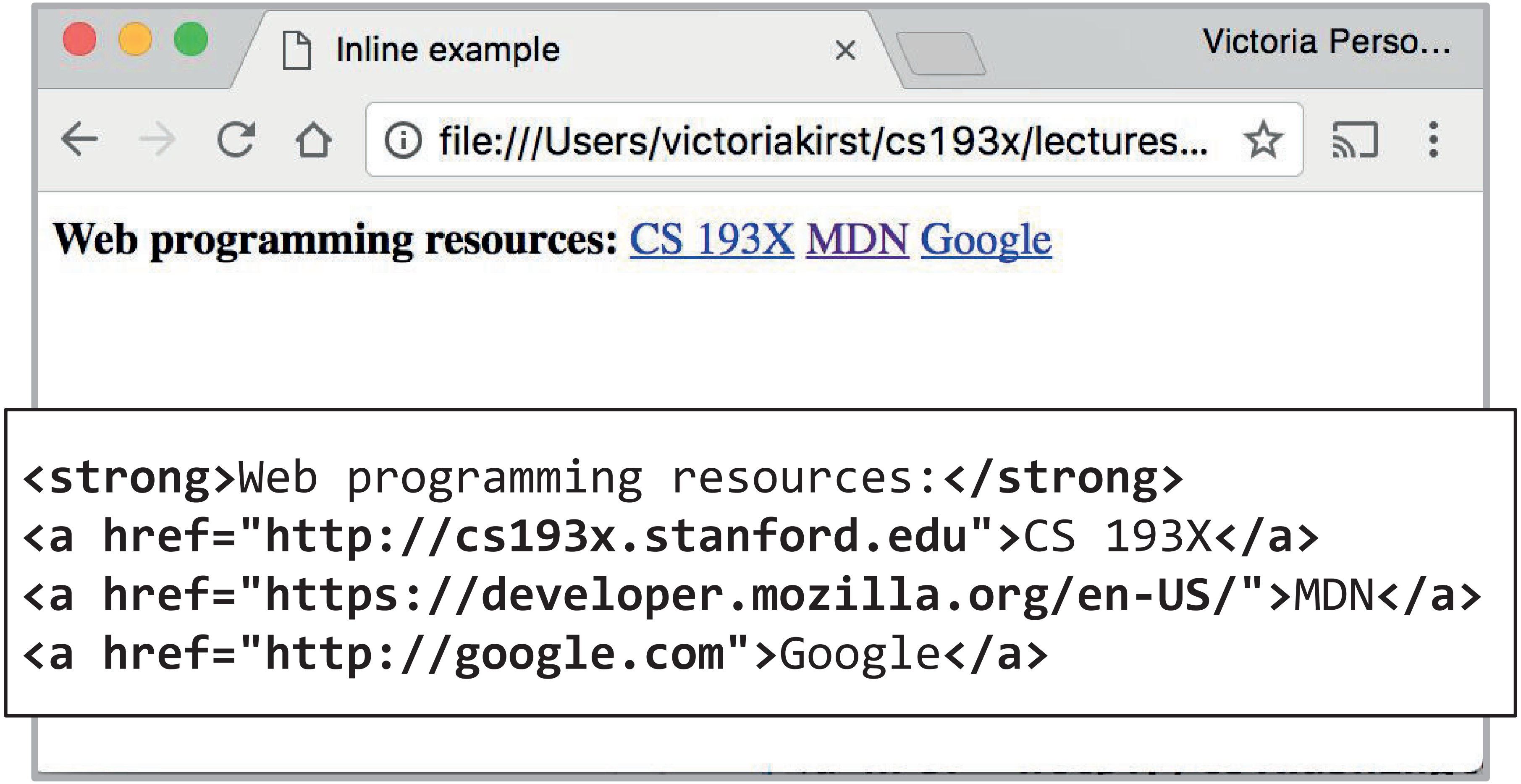


Web programming resources: CS 193X MDN Google

```
<strong>Web programming resources:</strong>
<a href="http://cs193x.stanford.edu">CS 193X</a>
<a href="https://developer.mozilla.org/en-US/">MDN</a>
<a href="http://google.com">Google</a>
```

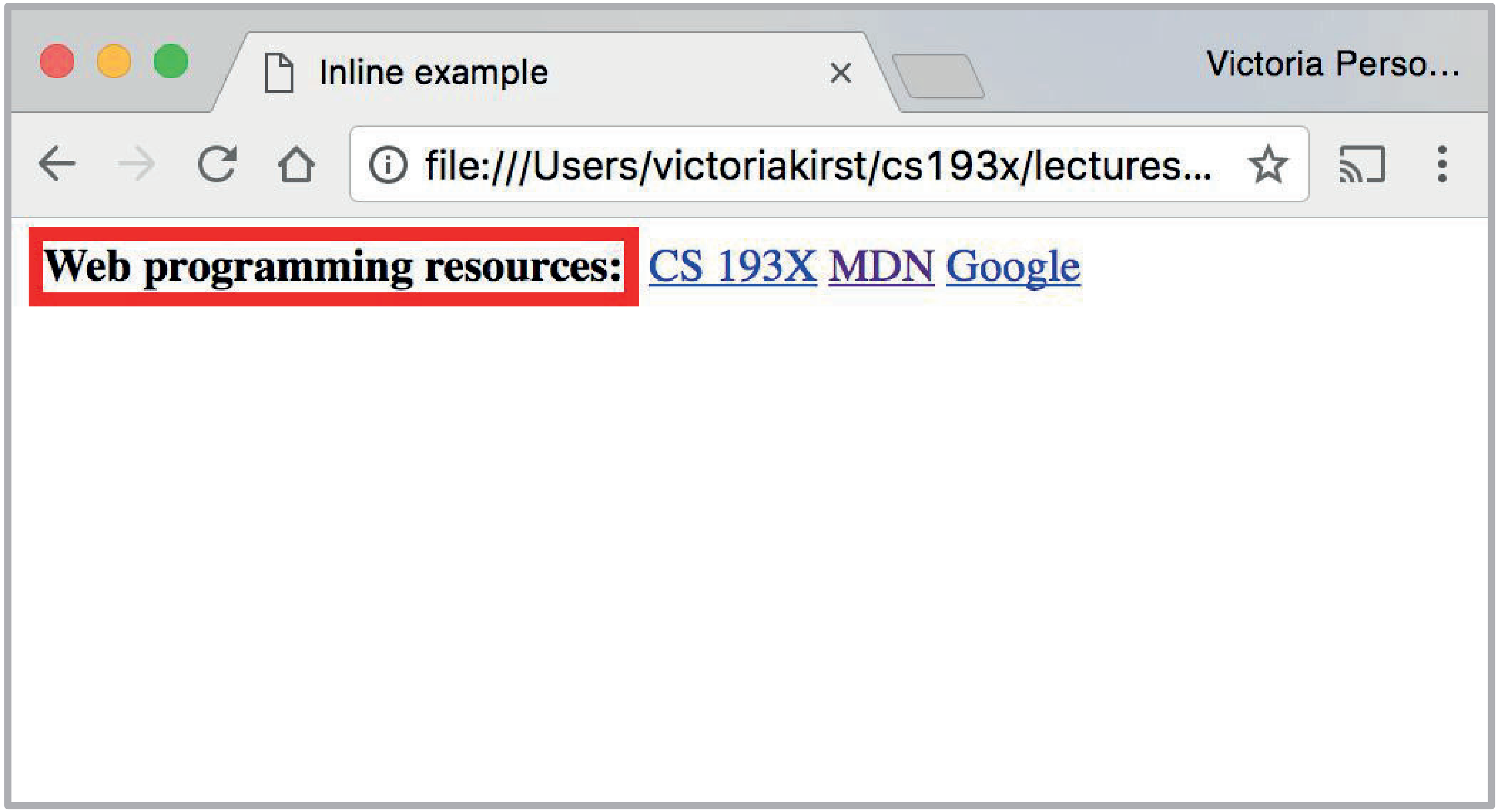
Q: What does this look like in the browser?

```
strong {
  border: 5px solid red;
  width: 1000px;
}
```



Web programming resources: CS 193X MDN Google

```
<strong>Web programming resources:</strong>
<a href="http://cs193x.stanford.edu">CS 193X</a>
<a href="https://developer.mozilla.org/en-US/">MDN</a>
<a href="http://google.com">Google</a>
```



Web programming resources: CS 193X MDN Google

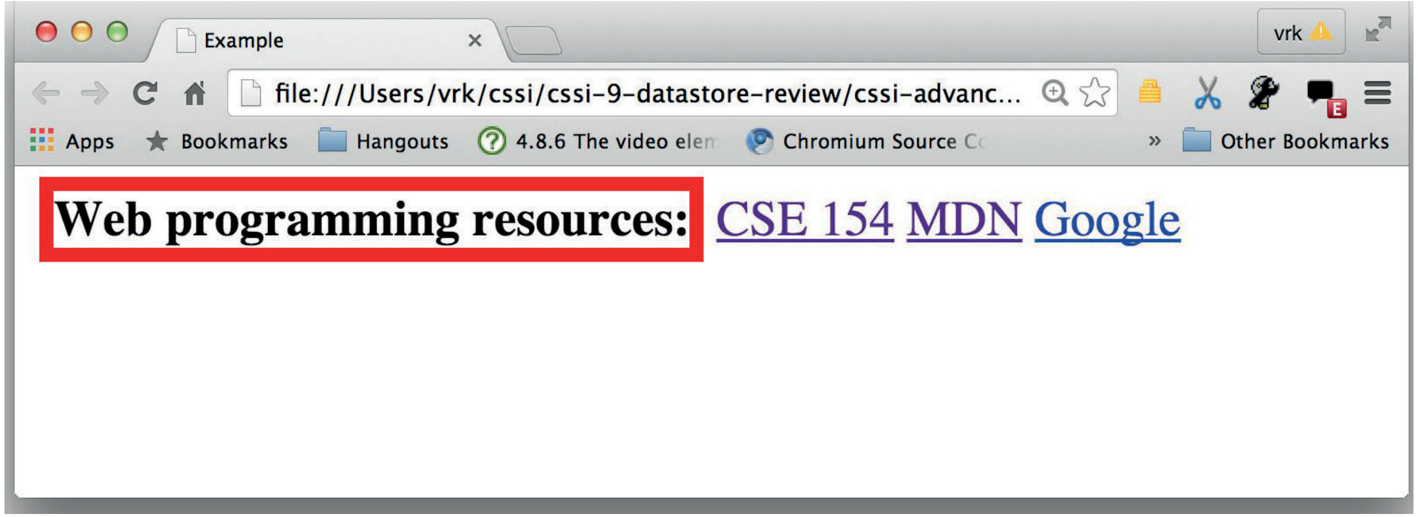
(Codepen)

Inline elements ignore width

width cannot be modified

```
strong {
  border: 5px solid red;
  width: 1000px;
  /* Will not work; strong is
     inline! */
}
```

```
<strong>Web programming reso
<a href="http://cs193x.stanf
<a href="https://developer.m
<a href="http://google.com">
```



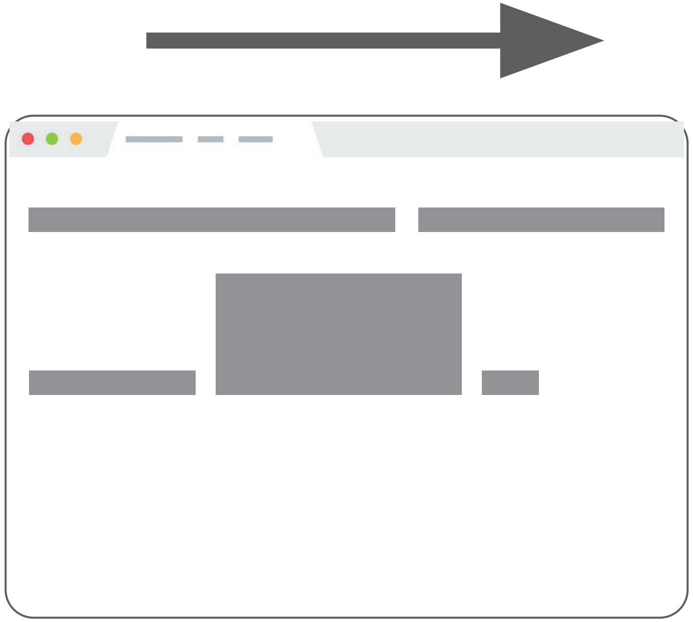
Web programming resources: CSE 154 MDN Google

Cannot set **`width`** on inline element, so it is ignored (Codepen)

inline-block

Examples: `<img>`, any element with `display: inline-block;`

- Width is the size of the content, i.e. it takes only as much space as needed (flows left to right)
- **Can** have `height` and `width`
- **Can** have a block element as a child
- **Can** be positioned (i.e. CSS properties like `float` and `position` apply)



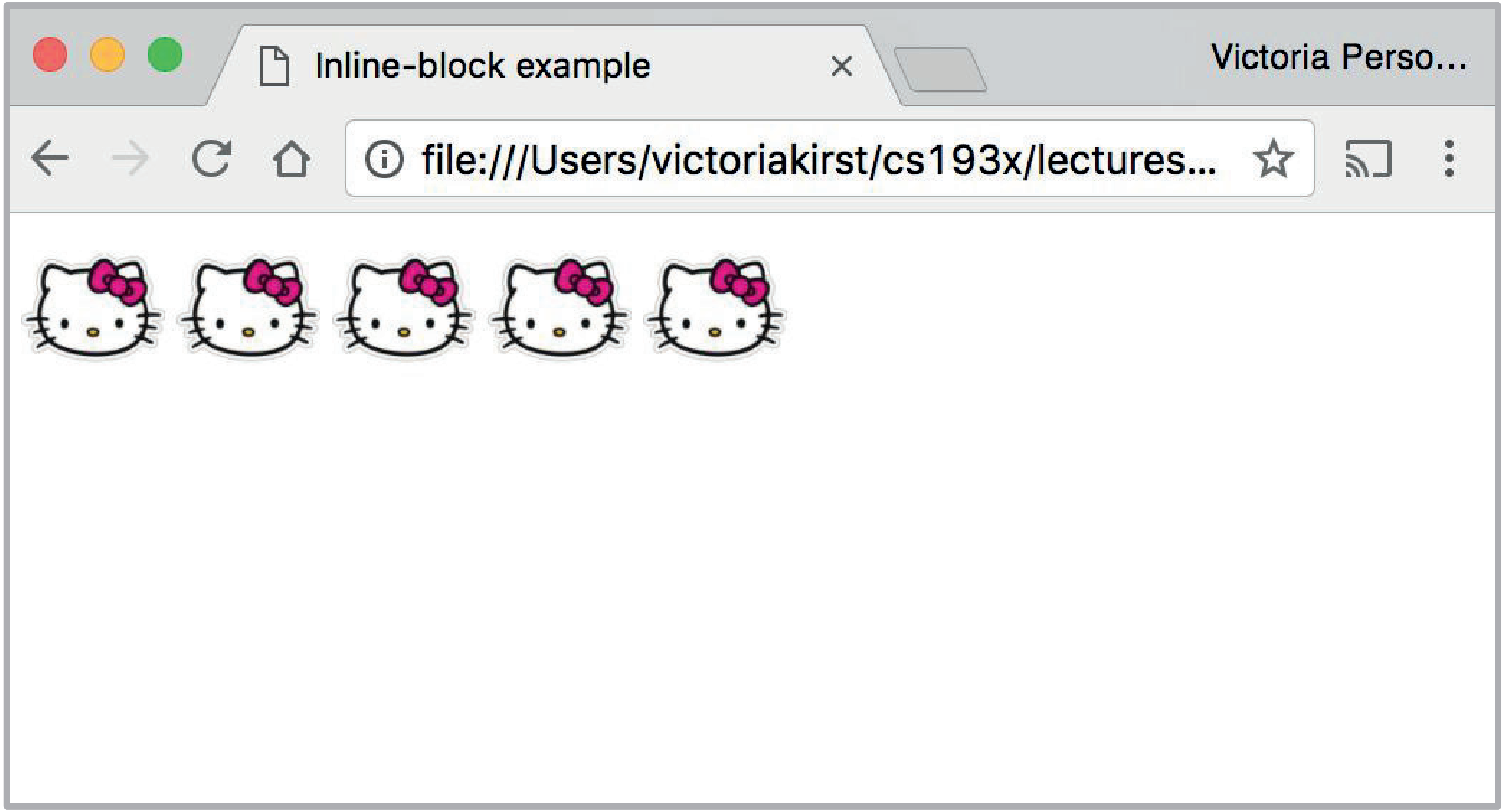
Example: Inline-block

```
img {
  width: 50px;
}
```

Q: What does this look like in the browser?

```
<img src="http://i.imgur.com/WJToVGv.jpg" />
<img src="http://i.imgur.com/WJToVGv.jpg" />
<img src="http://i.imgur.com/WJToVGv.jpg" />
<img src="http://i.imgur.com/WJToVGv.jpg" />
<img src="http://i.imgur.com/WJToVGv.jpg" />
```

http://i.imgur.com/WJToVGv.jpg =



Inline-block example
Victoria Perso...
file:///Users/victoriakirst/cs193x/lectures...

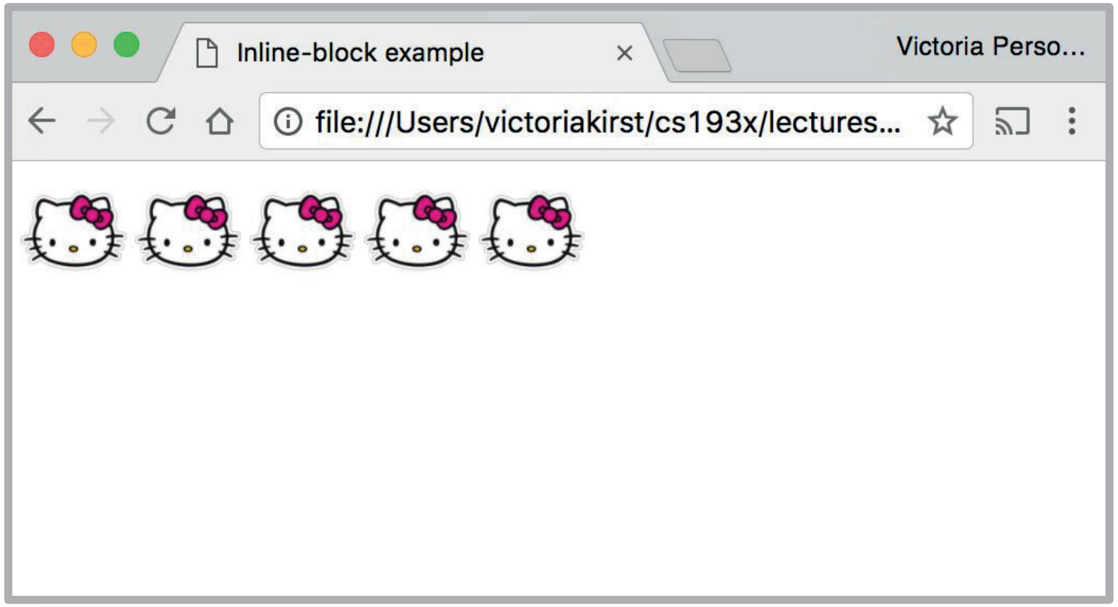
(Codepen)

Inline-block

Has width and height; flows left to right

Can set **`width`** on inline-block element, so image width is set to 50px. (Codepen)

`inline-block` flows left to right, so images are right next to each other.



```
img {
  width: 50px;
}
```

```
<img src="http://i.imgur.com/WJToVGv.jpg" />
<img src="http://i.imgur.com/WJToVGv.jpg" />
<img src="http://i.imgur.com/WJToVGv.jpg" />
<img src="http://i.imgur.com/WJToVGv.jpg" />
<img src="http://i.imgur.com/WJToVGv.jpg" />
```

The display CSS property

You can change an element's default rendering type by changing the **display** property. Examples:

```
p {
 display: inline;
}
```

```
a {
 display: block;
}
```

Possible values for display:

- block
- inline
- inline-block
- some others: link

Review

1. **block:** flows **top-to-bottom**; **has** **`height`** and **`width`**
   `<p>, <h1>, <blockquote>, <ol>, <ul>, <table>`
2. **inline:** flows **left-to-right**; **does not have** **`height`** and **`width`**
   `<a>, <em>, <strong>,<br>`
   a. **inline block:** flows **left-to-right**; **has** **`height`** and **`width`** equal to size of the content
      `<img>`

Questions?

Moral of the story:

If your CSS isn't working, see if you're trying to apply block-level properties to inline elements