

Today's schedule

Schedule:

a)

- Inline vs block
- CSS Classes and Ids

Read more: Other selectors

b)

- Box Model
- Mobile Web

HTML and CSS

Quick review

Recall: Course web page

We wrote [some HTML](#) to make the following page:

WPR: Web Pro

Announcements

15/8: Assignment 1 is out! Due in 3 weeks.

16/8: Office hours are now posted.

[View Syllabus](#)

That was weird

- We saw that HTML whitespace collapses into one space...

```
<h1>WPR: Web Pro</h1>  
<strong>Announcements</strong><br/>  
15/8: Assignment 1 is out! Due in 3  
weeks.<br/>
```

- Except weirdly the `<h1>` heading was on a line of its own, and `` was not.

Hmmm... strange...

Oh well, it works! Let's move on!!!

Exercise: Course web page

Let's write some CSS to style our page:

Font face: Helvetica

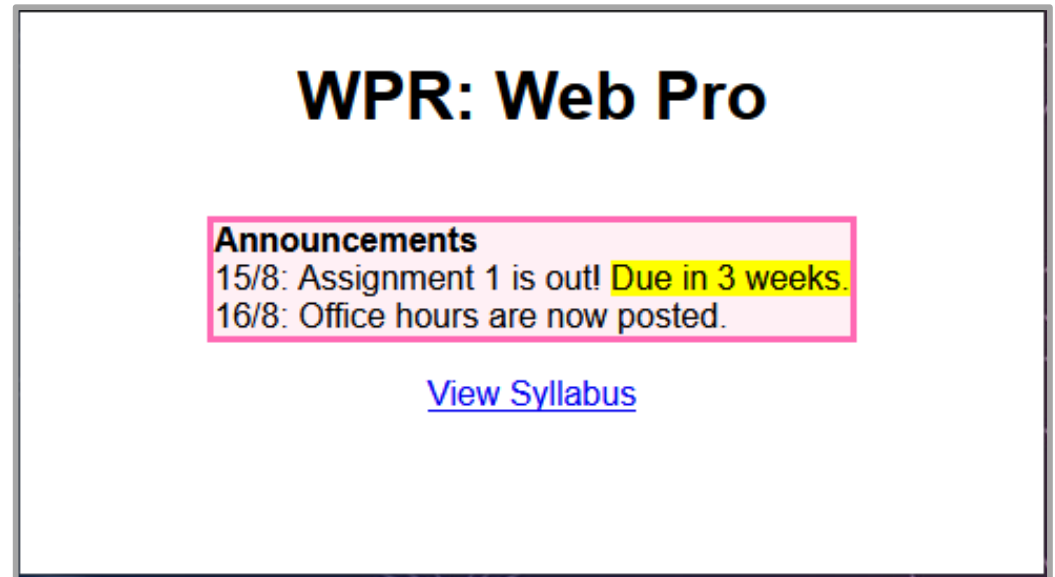
Border: hotpink 3px

Background color:

lavenderblush

Highlight: yellow

- Box is **centered**
- Header and link are **centered**
- Box contents are **left-aligned**



✦✦ [CodePen link: Let's code!](#) ✦✦

Solution?!

```
body {  
  font-family: Helvetica;  
}  
h1 {  
  text-align: center;  
}  
a {  
  text-align: center;  
}  
p {  
  border: 3px solid hotpink;  
  background-color: lavenderblush;  
}
```

Produces:

WPR: Web Pro

Announcements

15/8: Assignment 1 is out! Due in 3 weeks.

16/8: Office hours are now posted.

[View Syllabus](#)

CSS exercise debrief

We used some **key techniques**:

- Add invisible containers in HTML to select groups of elements in CSS.
- Apply styles to parent / ancestor element to style parent and all its children. (Will talk more about this later.)

CSS exercise debrief

But we encountered **more weirdness...**

- `text-align: center;` didn't work on the `<a>` tag
- The box was reaaaaaally wide!
- How to center the box?!
- How do you highlight?!

How do we get from this...

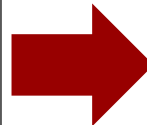
WPR: Web Pro

Announcements

15/8: Assignment 1 is out! Due in 3 weeks.

16/8: Office hours are now posted.

[View Syllabus](#)



... to this?

WPR: Web Pro

Announcements

15/8: Assignment 1 is out! Due in 3 weeks.

16/8: Office hours are now posted.

[View Syllabus](#)

Q: Why is HTML/CSS so bizarre??

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

block vs **inline** display

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>`, ``, ``, `<table>`
2. **inline**: small amount of content, no height or width
`<a>`, ``, ``, `
`
 - a. **inline block**: inline content with height and width
``
3. **metadata**: information about the page, usually not visible
`<title>`, `<meta>`

Block elements

Examples:

`<p>`, `<h1>`, `<blockquote>`, ``, ``, `<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

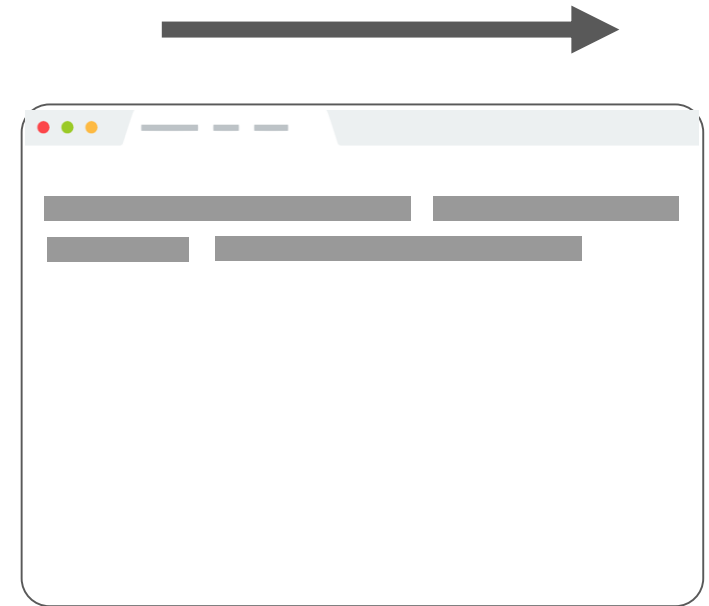
She likes *puppies*

Inline elements

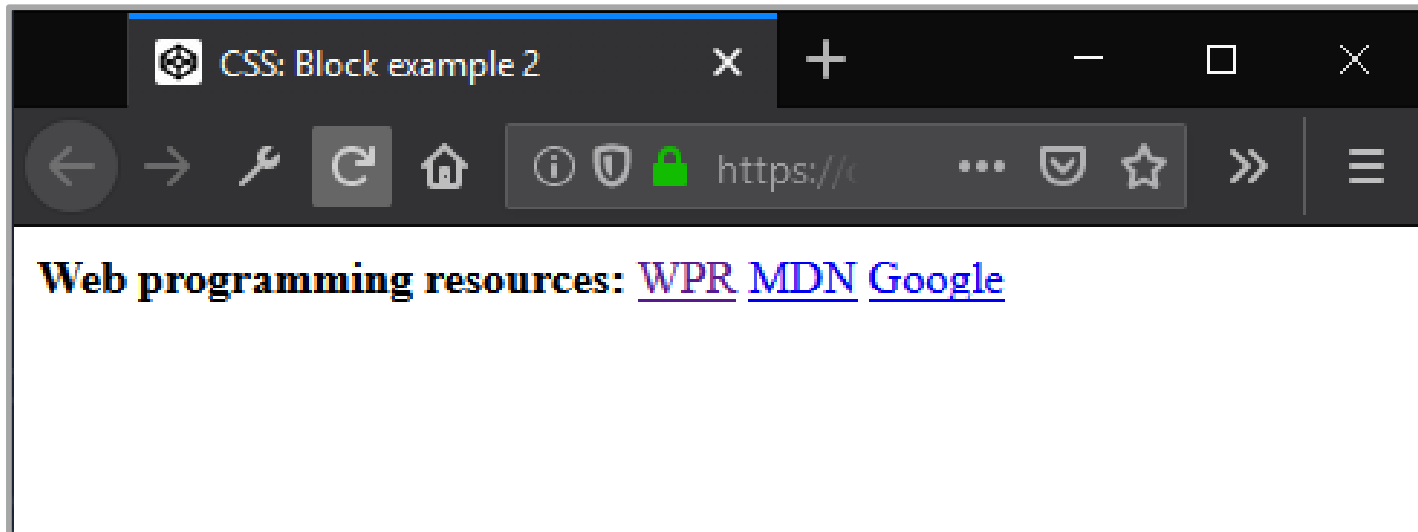
Examples:

`<a>`, ``, ``, `
`

- 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)
 - o Must position **its containing block element** instead



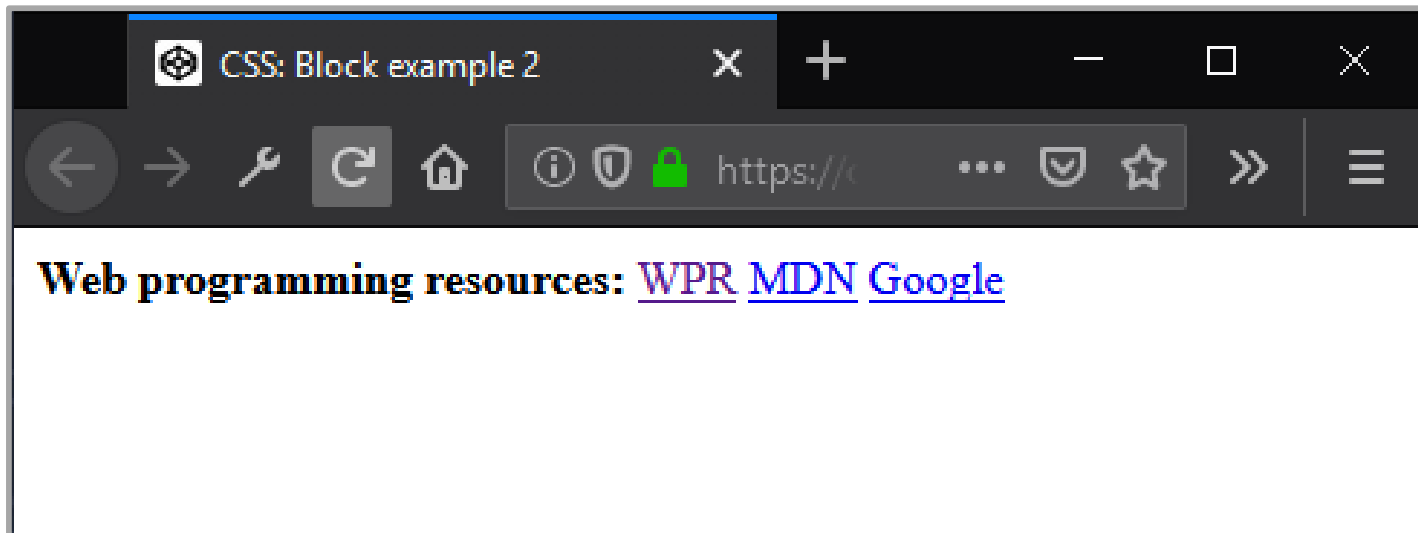
Example: Inline



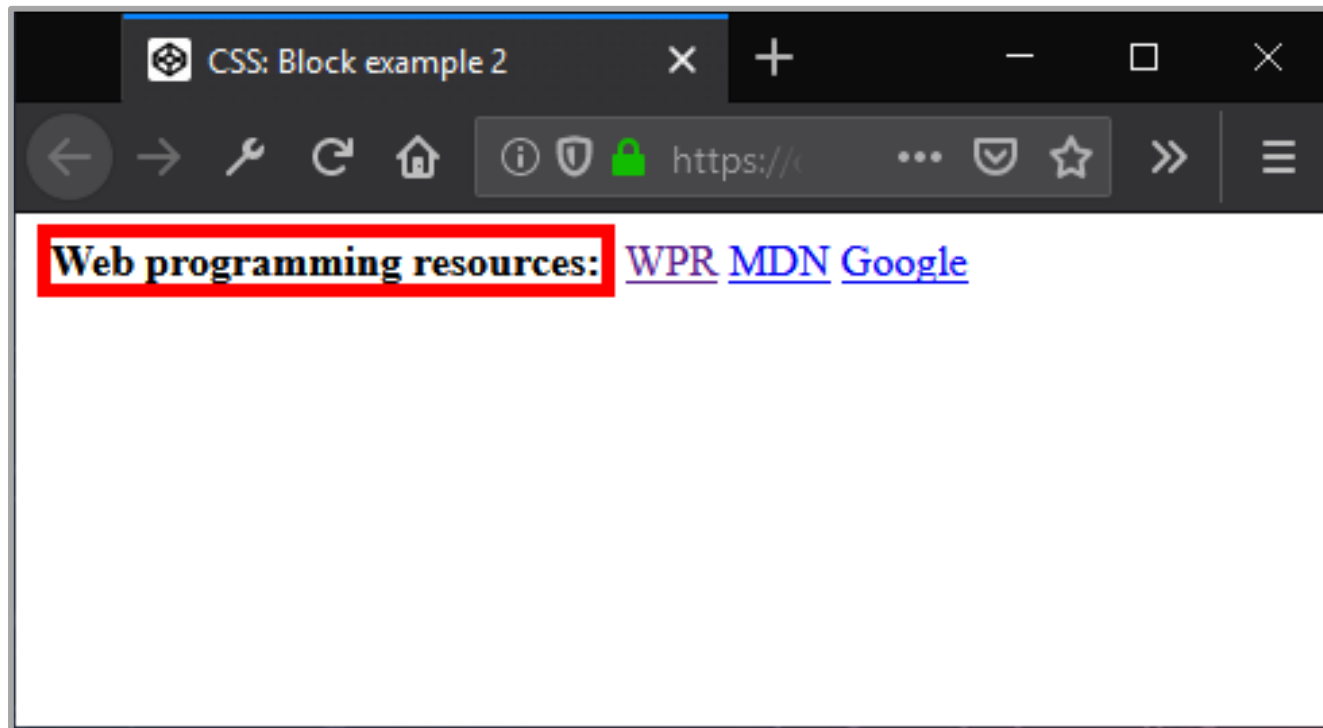
```
<strong>Web programming resources:</strong>  
<a href="http://fit.hanu.vn/">WPR</a>  
<a href="https://developer.mozilla.org/en-US/">MDN</a>  
<a href="http://google.com">Google</a>
```

**Q: What does this
look like in the
browser?**

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



```
<strong>Web programming resources:</strong>  
<a href="http://fit.hanu.vn/">WPR</a>  
<a href="https://developer.mozilla.org/en-US/">MDN</a>  
<a href="http://google.com">Google</a>
```

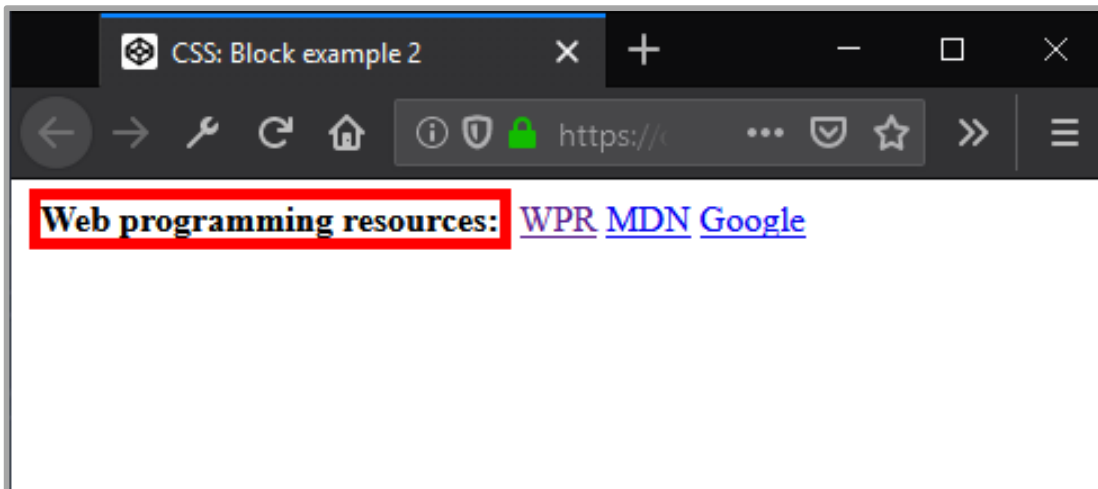


([Codepen](#))

Inline elements ignore width

width cannot be modified

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

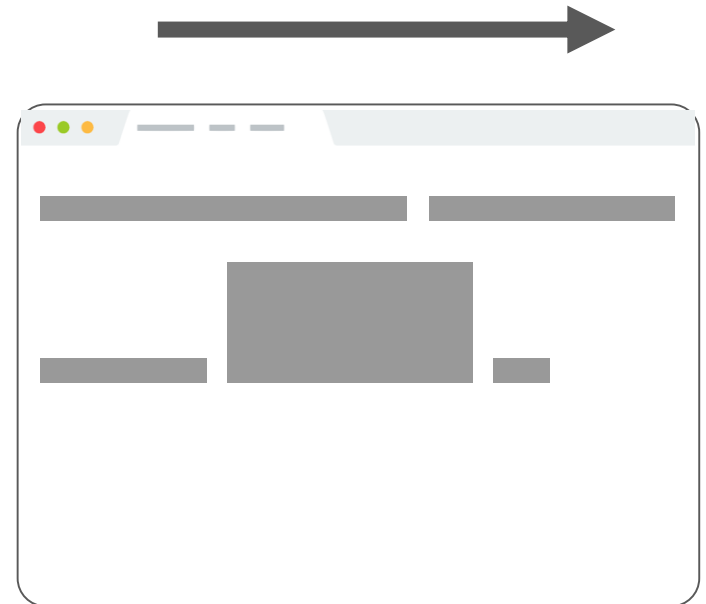


Cannot set **width** on inline element, so it is ignored ([Codepen](#))

inline-block

Examples: ``, 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;  
}
```

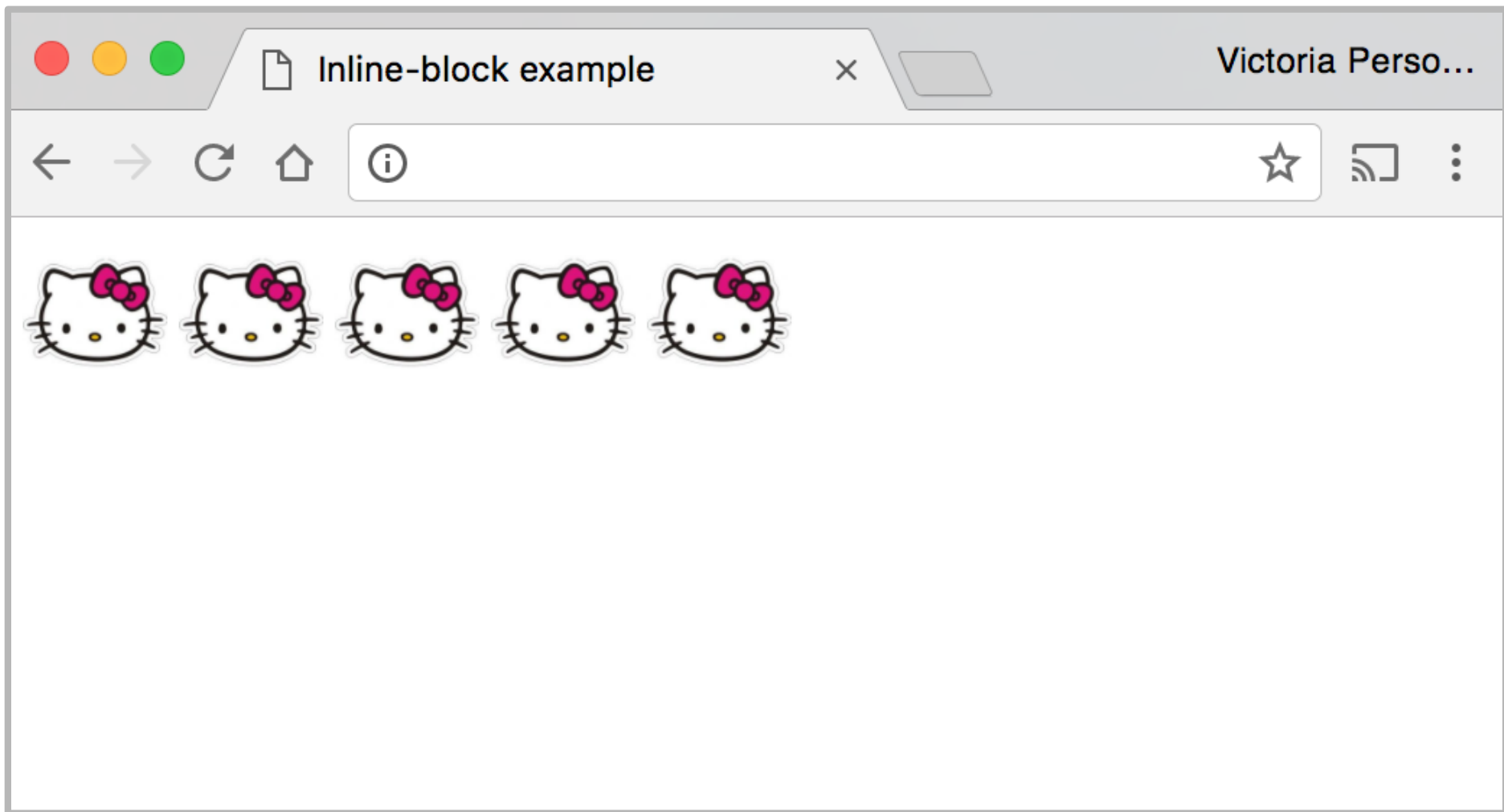
Q: What does this look like in the browser?

```
  
  
  
  

```

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





([Codepen](#))

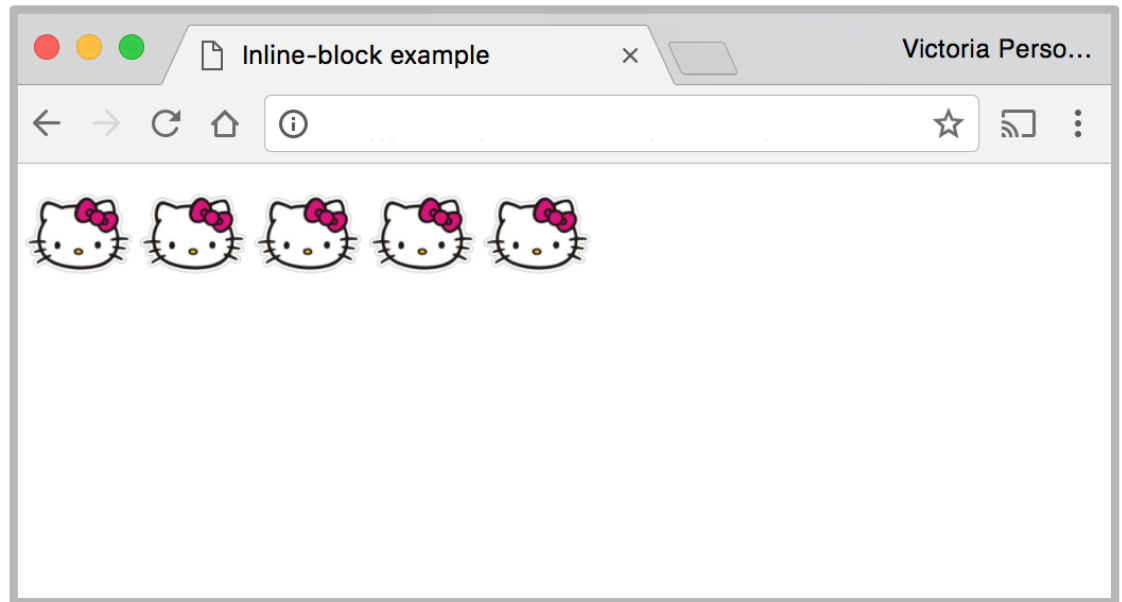
Inline-block

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;  
}
```

```
  
  
  
  

```

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>`, ``, ``, `<table>`
2. **inline**: flows **left-to-right**; **does not have** height and width
`<a>`, ``, ``, `
`
 - a. **inline block**: flows **left-to-right**; **has** height and width
equal to size of the content
``

Questions?

Moral of the story:

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

h1 vs strong mystery

Recall: Weirdly the `<h1>` heading was on a line of its own, and `` was not. -- **Why?**

```
<h1>WPR: Web Pro</h1>
<strong>Announcements</strong>
15/8: Assignment 1 is out!
Due in 3 weeks.<br/>
```

WPR: Web Pro

Announcements 15/8: Assignment 1 is out! Due in 3 weeks.

```
<h1>WPR: Web Pro</h1>
<strong>Announcements</strong><br/>
15/8: Assignment 1 is out!
Due in 3 weeks.<br/>
```

WPR: Web Pro

Announcements
15/8: Assignment 1 is out! Due in 3 weeks.

h1 vs strong mystery

Recall: Weirdly the `<h1>` heading was on a line of its own, and `` was not. -- **Why?**

```
<h1>WPR: Web Pro</h1>
<strong>Announcements</strong>
15/8: Assignment 1 is out!
Due in 3 weeks.<br/>
```

WPR: Web Pro

Announcements 15/8: Assignment 1 is out! Due in 3 weeks.

```
<h1>WPR: Web Pro</h1>
<strong>Announcements</strong><br/>
15/8: Assignment 1 is out!
Due in 3 weeks.<br/>
```

WPR: Web Pro

Announcements
15/8: Assignment 1 is out! Due in 3 weeks.

**Because h1 is a block-level element,
and strong is an inline-level element**

text-align mystery

Recall: We couldn't set `text-align: center;` on the `<a>` tag directly, but we could center `<h1>`. **Why?**

```
h1 { /* works! */  
  text-align: center;  
}  
  
a { /* fails :( */  
  text-align: center;  
}
```

WPR: Web Pro

Announcements

15/8: Assignment 1 is out! Due in 3 weeks.

16/8: Office hours are now posted.

[View Syllabus](#)

Let's try looking at the [MDN description of text-align](#)...

text-align mystery

Summary

The **text-align** CSS property describes how inline content like text is aligned in its parent block element. **text-align** does not control the alignment of block elements, only their inline content.

Initial value

start, or a nameless value that acts as left if **direction** is ltr, right if **direction** is rtl if start is not supported by the browser.

Applies to

block containers

([source](#))

text-align demystified!

Why? From the [spec](#), **can't apply text-align to an inline element**; must apply text-align to its block container, or set **a { display : block; }**

The screenshot shows a web browser window with a dark theme. On the left, there are two panels: 'HTML' and 'CSS'. The 'HTML' panel shows the following code:

```
<p>
  <a href="http://cs193x.stanford.ed
    View Syllabus
  </a>
</p>
```

The 'CSS' panel shows the following code:

```
p { /* works! */
  text-align: center;
}
```

On the right, the rendered page is shown. It has a title 'WPR: Web Pro' in a large, bold, black serif font. Below the title is a section 'Announcements' in a smaller, bold, black serif font. Under 'Announcements', there are two lines of text: '15/8: Assignment 1 is out! Due in 3 weeks.' and '16/8: Office hours are now posted.' Below this text is a link 'View Syllabus' in a blue, underlined serif font. The text and link are centered horizontally on the page.

Box size mystery

Recall: The pink box we put around the announcements extended the entirety of the page.

```
p {  
  border: 3px solid hotpink;  
  background-color: lavenderblush;  
}
```

WPR: Web Pro

Announcements

15/8: Assignment 1 is out! Due in 3 weeks.

16/8: Office hours are now posted.

[View Syllabus](#)

Why?

How do we fix this?

Box size mystery

Recall: The pink box we put around the announcements extended the entirety of the page.

```
p {  
  border: 3px solid hotpink;  
  background-color: lavenderblush;  
}
```

WPR: Web Pro

Announcements

15/8: Assignment 1 is out! Due in 3 weeks.

16/8: Office hours are now posted.

[View Syllabus](#)

Why? Because `p` is block-level, so `width == width of the page`

How do we fix this?

Box size mystery: demystified!

Recall: The pink box we put around the announcements extended the entirety of the page.

```
p {  
  border: 3px solid hotpink;  
  background-color: lavenderblush;  
  display: inline-block;  
}
```

WPR: Web Pro

Announcements

15/8: Assignment 1 is out! Due in 3 weeks.
16/8: Office hours are now posted.

[View Syllabus](#)

Why? Because `p` is block-level, so `width == width of the page`

How do we fix this? Change `display` to `inline-block` (though now the space above the box has increased... will address later!)

Centering the box

We can also center the box by centering the body tag, since p is now inline-block.

```
body {  
  text-align: center;  
}  
  
p {  
  border: 3px solid hotpink;  
  background-color: lavenderblush;  
  display: inline-block;  
}
```

WPR: Web Pro

Announcements

15/8: Assignment 1 is out! Due in 3 weeks.
16/8: Office hours are now posted.

[View Syllabus](#)

Highlight mystery

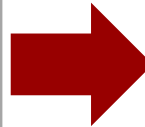
Recall: We didn't know how to select a random snippet of text to change its background.

WPR: Web Pro

Announcements

15/8: Assignment 1 is out! Due in 3 weeks.
16/8: Office hours are now posted.

[View Syllabus](#)



WPR: Web Pro

Announcements

15/8: Assignment 1 is out! Due in 3 weeks.
16/8: Office hours are now posted.

[View Syllabus](#)

How do we fix this?

Highlight: demystified!

We can select a random segment of text by wrapping it in an **inline element**:



The image shows a web browser's developer tools interface. On the left, the 'HTML' and 'CSS' panels are visible. The 'HTML' panel shows the following code:

```
<strong>Announcements</strong><br/>
4/3: Homework 0 is out!
<em>Due Friday.</em><br/>
4/3: Office hours are now posted.
```

The 'CSS' panel shows the following code:

```
em {
  font-style: normal; /* undoes italics */
  background-color: yellow;
}
```

On the right, the rendered page is shown. It has a title 'WPR: Web Pro'. Below the title, there is a section titled 'Announcements' which contains two paragraphs: '15/8: Assignment 1 is out! Due in 3 weeks.' and '16/8: Office hours are now posted.'. The second paragraph is highlighted in yellow. Below the announcements, there is a link 'View Syllabus'.

Hmmm... but wouldn't it be better to have a "highlight" element?

Highlight: demystified!

We can select a random segment of text by wrapping it in an **inline element**:

The screenshot shows a web editor with two panels on the left: HTML and CSS, and a preview panel on the right.

HTML Panel: The code is as follows:

```
<strong>Announcements</strong><br/>
4/3: Homework 0 is out!
<em>Due Friday.</em><br/>
4/3: Office hours are now posted.
```

The `Due Friday.` segment is circled in red.

CSS Panel: The code is as follows:

```
em {
  font-style: normal; /* undoes italics */
  background-color: yellow;
}
```

Preview Panel: The rendered output shows the text "WPR: Web Pro" at the top. Below it, the word "Announcements" is bolded. The text "15/8: Assignment 1 is out! Due in 3 weeks." is highlighted in yellow, and "16/8: Office hours are now posted." is below it. A blue link "View Syllabus" is at the bottom.

Hmmm... but wouldn't it be better to have a "highlight" element?
How do we make a generic HTML element?

Have you heard of `<div>` and ``?

What are they?

<div> and

Two generic tags with no intended purpose or style:

- <div>: a generic **block** element
- : a generic **inline** element

 in action

We can use as a generic inline HTML container:

HTML

```
<strong>Announcements</strong><br/>
4/3: Homework 0 is out!
<span>Due Friday.</span><br/>
4/3: Office hours are now posted.
```

CSS

```
span {
  background-color: yellow;
}
```

WPR: Web Pro

Announcements
15/8: Assignment 1 is out! Due in 3 weeks.
16/8: Office hours are now posted.

[View Syllabus](#)

Multiple generic containers?

But won't we often want multiple generic containers?

How do we distinguish two generic containers?

In other words, how do we select a subset of elements instead of **all** elements on the page?

WPR: Web Pro

Announcements

15/8: Assignment 1 is out! Due in 3 weeks.

16/8: Office hours are now posted.

[View Syllabus](#)

CSS Selectors: Classes and Ids

Classes and ids

There are 3 basic types of CSS selectors:

Element selector (this is the one we've been using)	p	All <p> elements
✦✦ID selector✦✦	#abc	element with id="abc"
✦✦Class selector✦✦	.abc	elements with class="abc"

```
<h1 id="title">Homework</h1>
<em class="hw">HW0</em> is due Friday.<br/>
<em class="hw">HW1</em> goes out Monday.<br/>
<em>All homework due at 11:59pm.</em>
```

Classes and ids

```
<h1 id="title">Homework</h1>
<em class="hw">HW0</em> is due Friday.<br/>
<em class="hw">HW1</em> goes out Monday.<br/>
<em>All homework due at 11:59pm.</em>
```

```
.hw {
  color: hotpink;
}

#title {
  color: purple;
}
```

Homework

HW0 is due Friday.

HW1 goes out Monday.

All homework due at 11:59pm.

More on `class` and `id`

- **`class`** and **`id`** are special HTML attributes that can be used on any HTML element
 - **`class`**: Used on 1 or more elements; identifies a **collection** of elements
 - **`id`**: Used on exactly 1 element per page; identifies **one unique** element
- Can apply multiple classes by space-separating them:
`HW1`
- Often used with `span` and `div` to create generic elements: e.g. `` is like creating a "highlight" element

Selector summary

Example	Description
p	All <p> elements
.abc	All elements with the abc class , i.e. class="abc"
#abc	Element with the abc id , i.e. id="abc"
p.abc	<p> elements with abc class
p#abc	<p> element with abc id (p is redundant)
div strong	 elements that are descendants of a <div>
h2, div	<h2> elements and <div> s

Before we move on:
A few style notes

Why not `<div>` everywhere?

Technically, you can define your entire web page using `<div>` and the `class` attribute.

- Is this a good idea?
- Why does HTML have `ids` when you have `classes`?
- Why does HTML have `<p>`, `<h1>`, ``, etc. when you have `<div>`, ``, `class`, and `id`?

Next!

Box Model &
Mobile Web