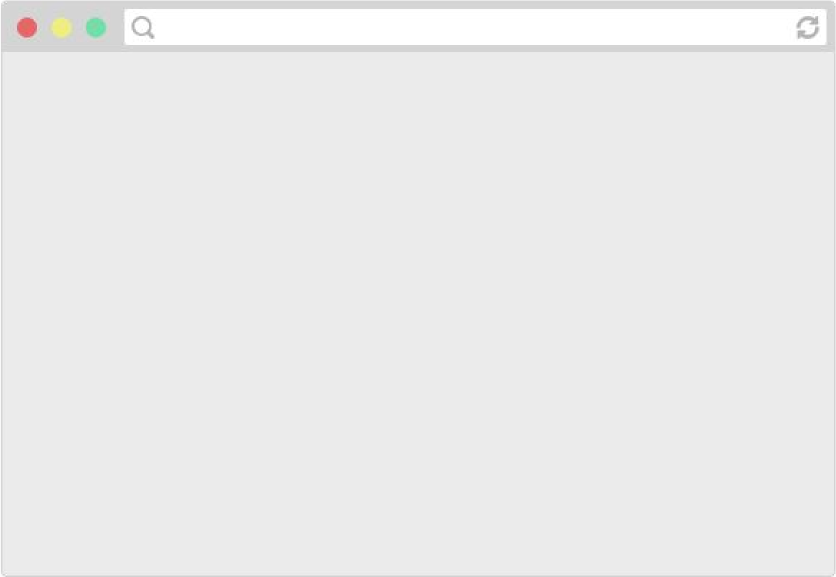
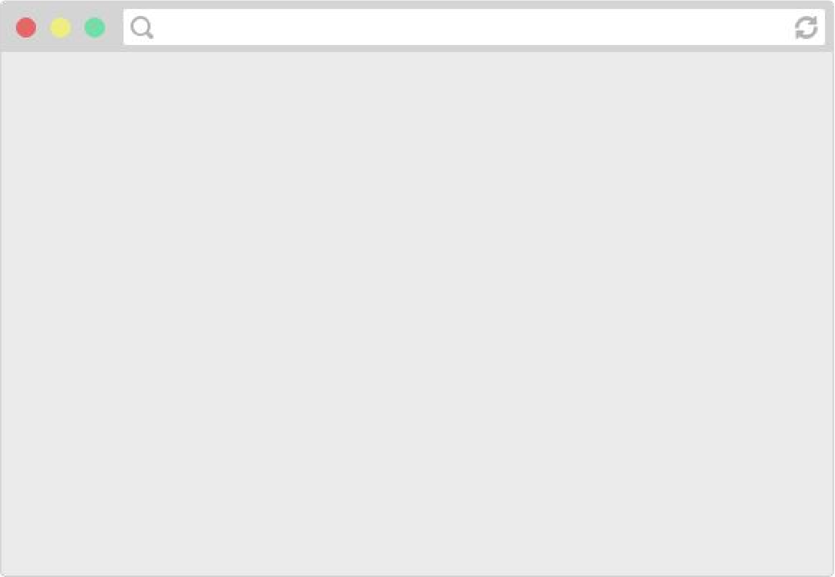
Browsers!
The Internet!
The web!

# How do web pages work?



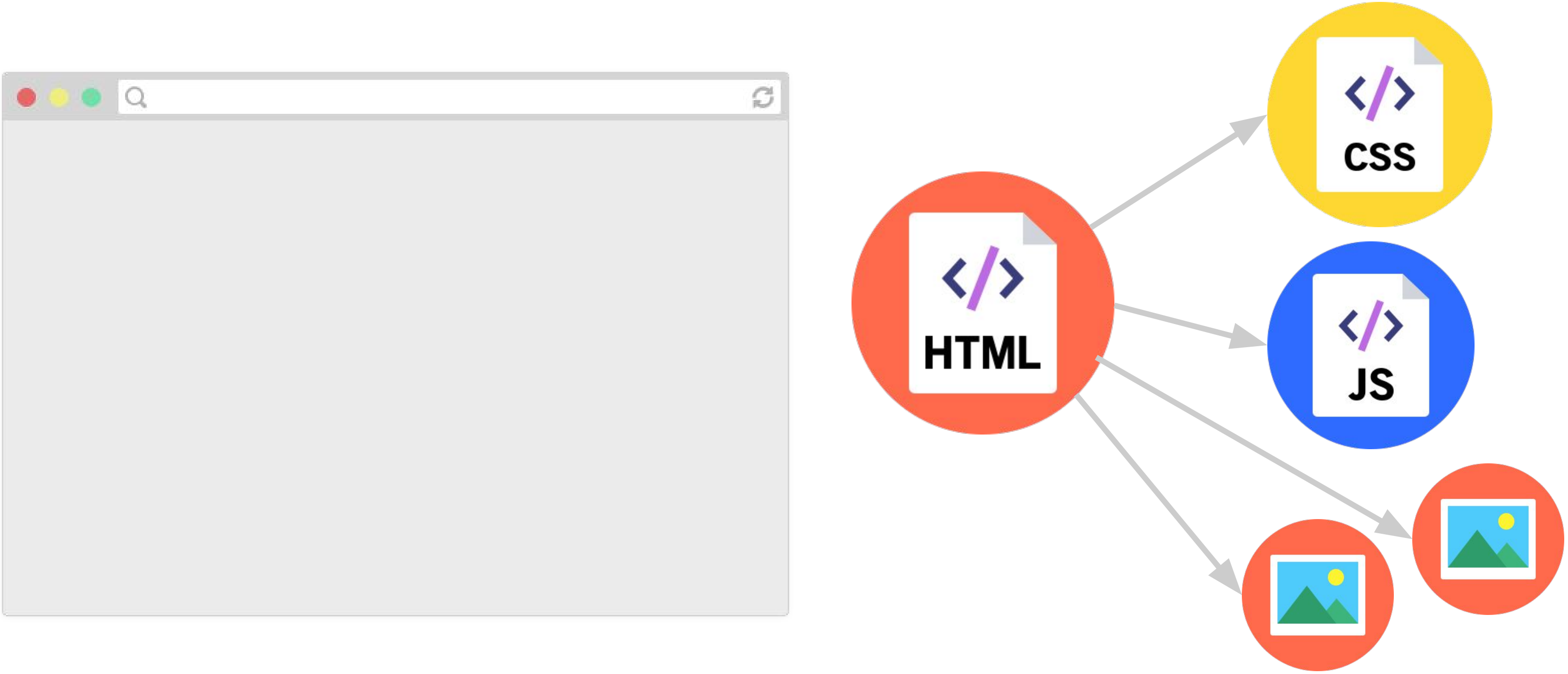
Browsers are applications that can display web pages.
E.g. Chrome, Firefox, Safari, Internet Explorer, Edge, etc.

# How do web pages work?



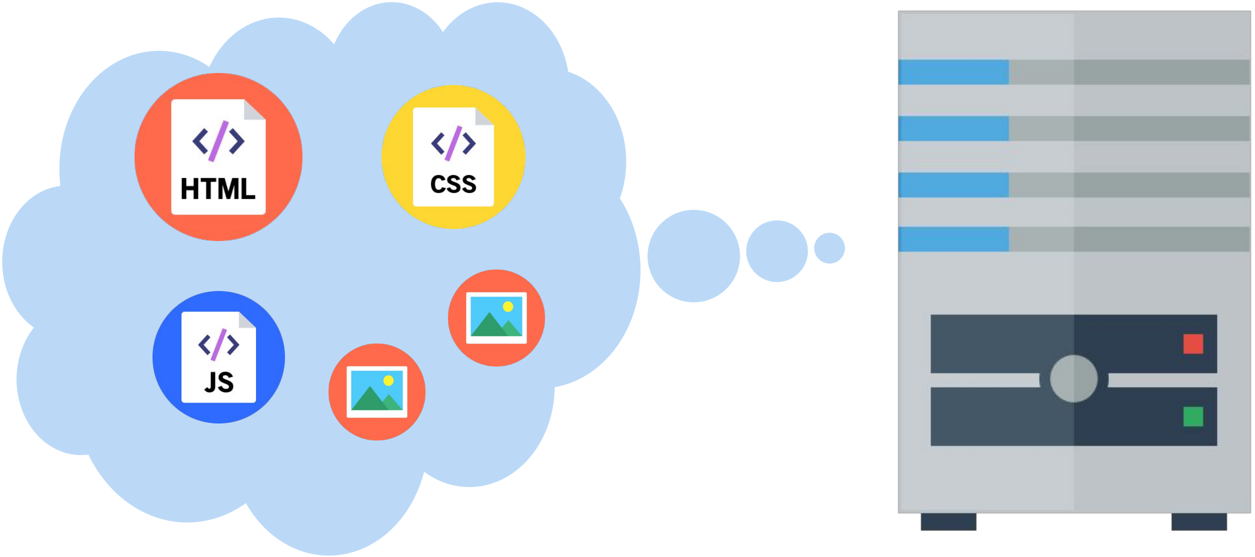
Web pages are written in a markup language called **HTML**, so browsers display a web page by reading and interpreting its HTML.

# How do web pages work?



The HTML file might link to other resources, like images, videos, as well as **JavaScript** and **CSS** (stylesheet) files, which the browser then also loads.

# How do web pages work?

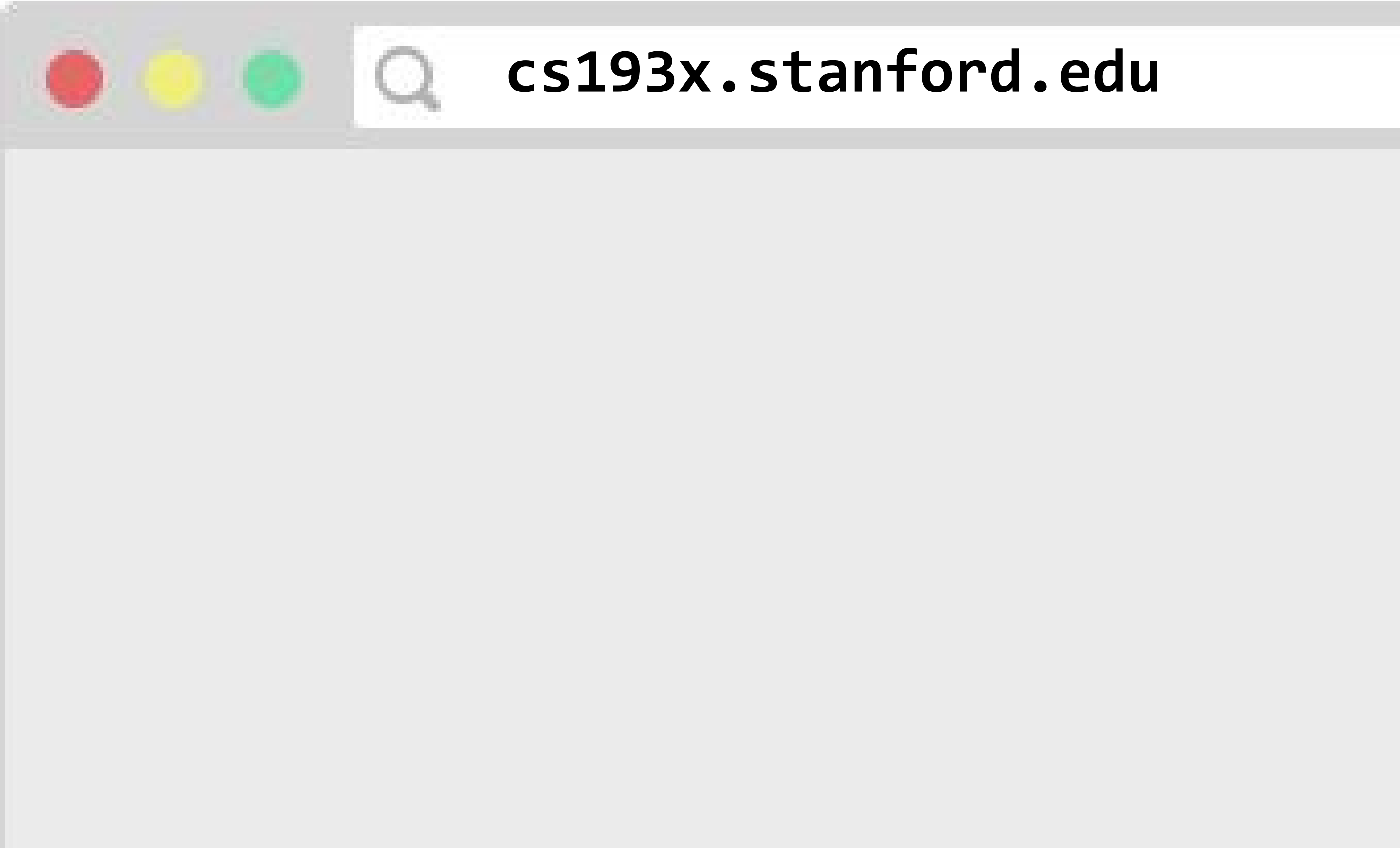


A **web server** is a program running on a computer that delivers web pages in response to requests.

It either stores or generates the web page returned.

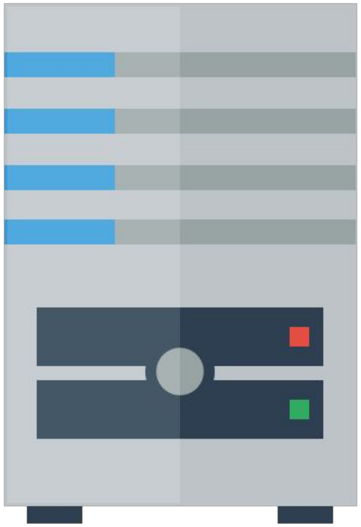
# How do web pages work?

1. You type in a URL, which is the address of the HTML file on the internet.

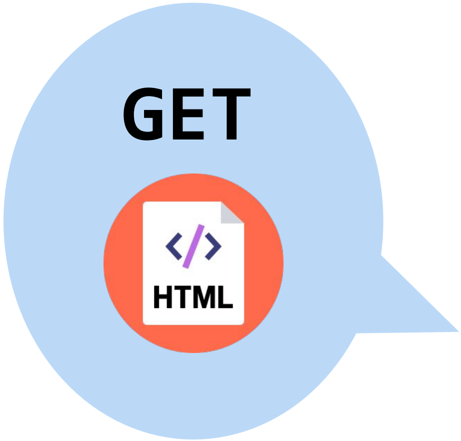


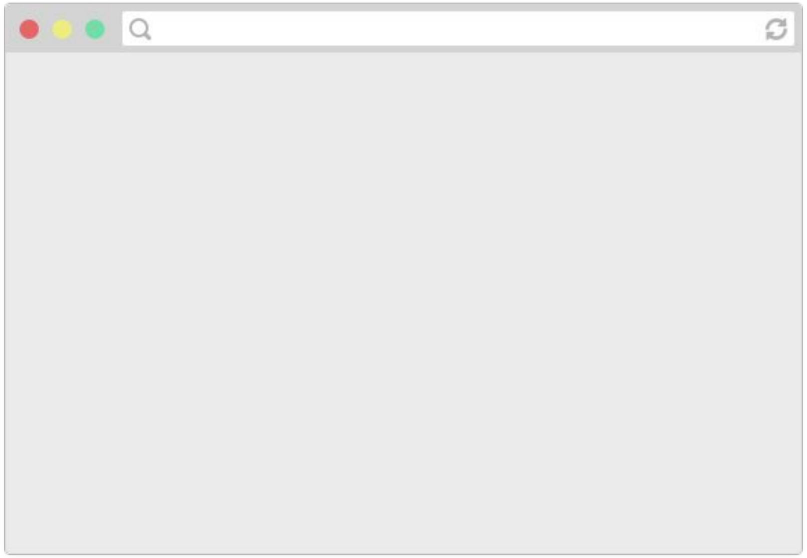
cs193x.stanford.edu

# How do web pages work?

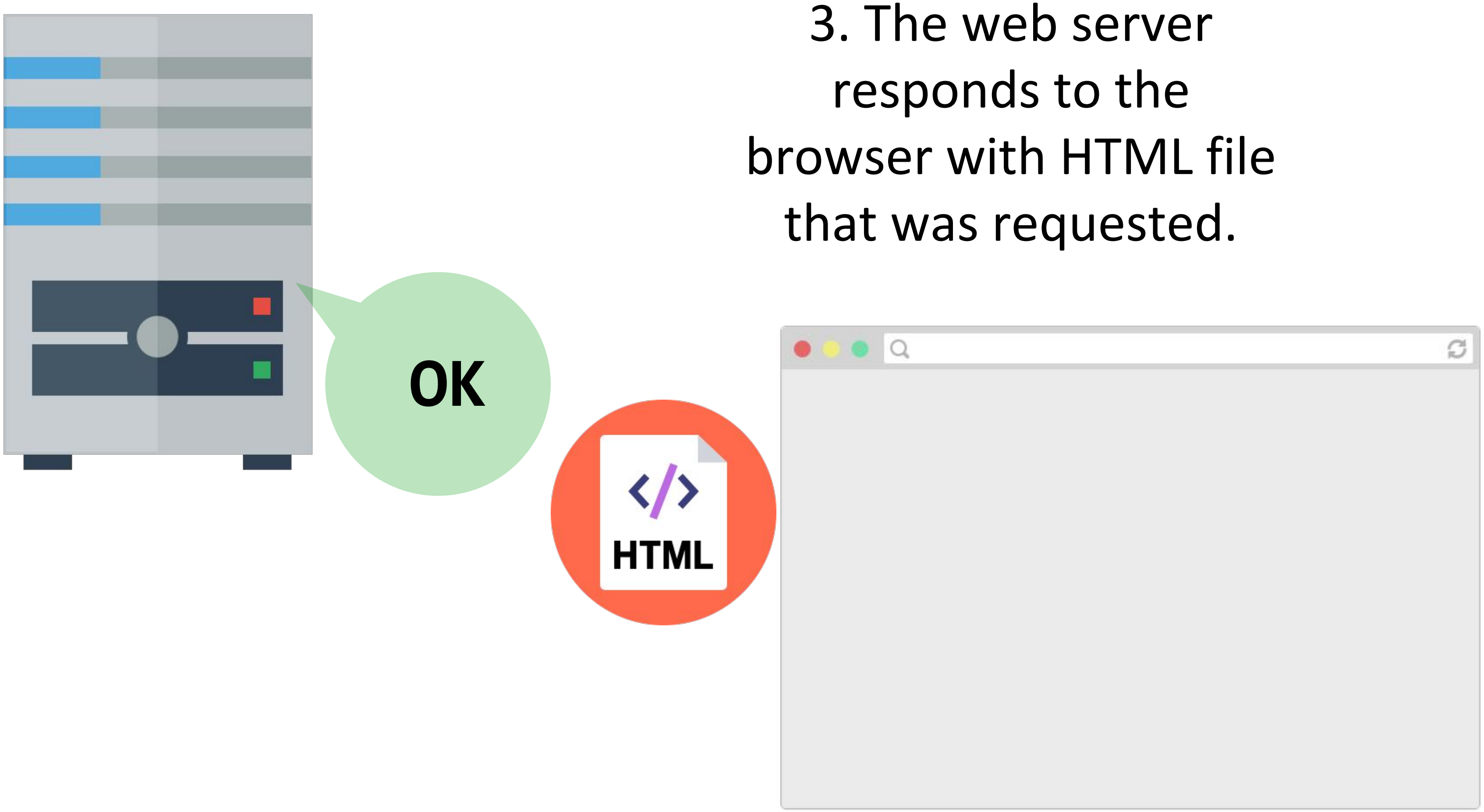


2. The browser asks the web server that hosts the document to send that document.





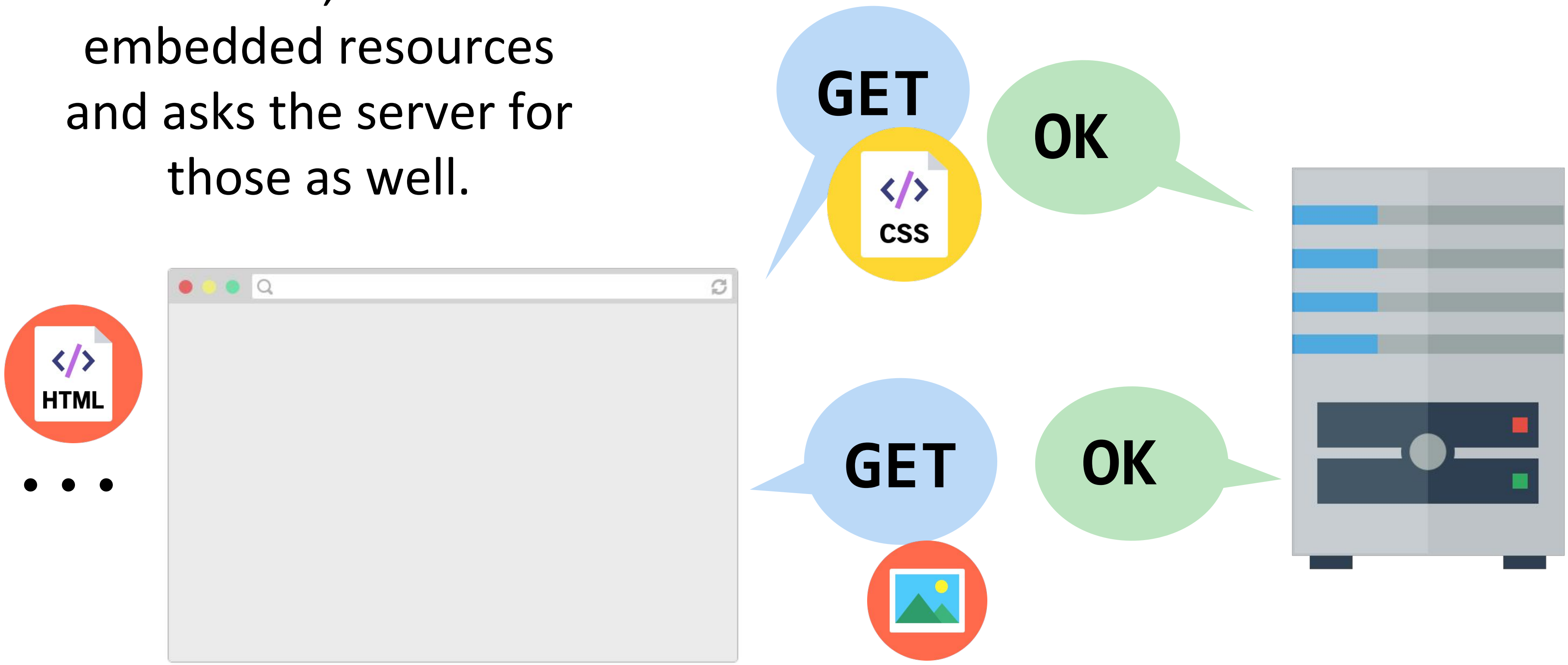
# How do web pages work?



3. The web server responds to the browser with HTML file that was requested.

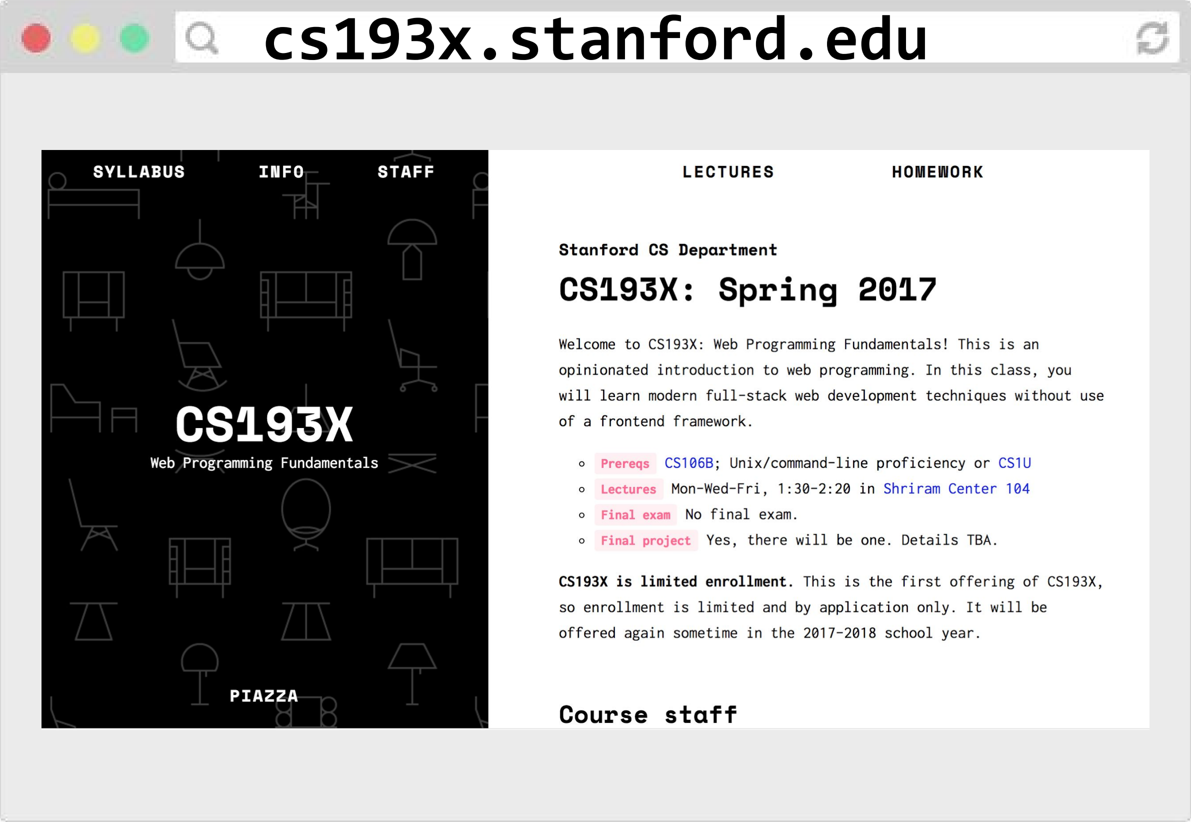
# How do web pages work?

4. The browser reads the HTML, sees the embedded resources and asks the server for those as well.



# How do web pages work?

5. The web page is loaded when all the resources are fetched and displayed.



Stanford CS Department

## CS193X: Spring 2017

Welcome to CS193X: Web Programming Fundamentals! This is an opinionated introduction to web programming. In this class, you will learn modern full-stack web development techniques without use of a frontend framework.

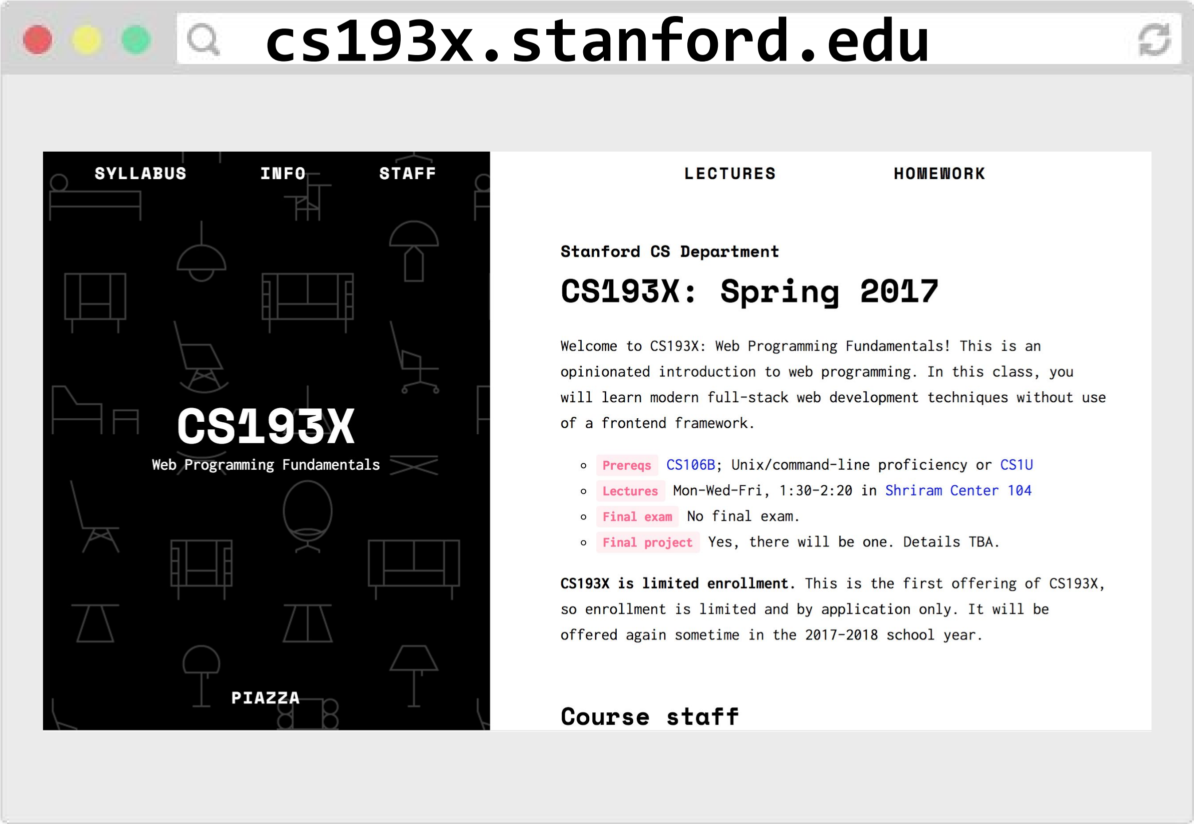
- Prereqs CS106B; Unix/command-line proficiency or CS1U
- Lectures Mon-Wed-Fri, 1:30-2:20 in Shriram Center 104
- Final exam No final exam.
- Final project Yes, there will be one. Details TBA.

**CS193X is limited enrollment.** This is the first offering of CS193X, so enrollment is limited and by application only. It will be offered again sometime in the 2017-2018 school year.

## Course staff

# P.S.

(That was obviously very hand-wavy. We'll get more detailed when we talk about servers later in the quarter.)



Stanford CS Department

## CS193X: Spring 2017

Welcome to CS193X: Web Programming Fundamentals! This is an opinionated introduction to web programming. In this class, you will learn modern full-stack web development techniques without use of a frontend framework.

- Prereqs CS106B; Unix/command-line proficiency or CS1U
- Lectures Mon-Wed-Fri, 1:30-2:20 in Shriram Center 104
- Final exam No final exam.
- Final project Yes, there will be one. Details TBA.

**CS193X is limited enrollment.** This is the first offering of CS193X, so enrollment is limited and by application only. It will be offered again sometime in the 2017-2018 school year.

## Course staff

# HTML and CSS

# HTML and CSS strategy

**Assumption:** Most people have cursory familiarity with HTML and CSS. Therefore we will:

- **Speed through** the obvious stuff
- **Skip** self-explanatory syntax
- **Skip** the parts you can look up easily through Google

✦ **Therefore, be aggressive with questions!** ✦

# What is HTML?

**HTML** (**H**yper**t**ext **M**arkup **L**anguage)

- Describes the **content** and **structure** of a web page; not a programming language.
- Made up of building blocks called **elements**.

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

# Basic HTML page structure

(i.e. copy/paste boilerplate)

```
<!DOCTYPE html>
<html>
  <head>
   <title>CS 193X</title>
  </head>

  <body>
   ... contents of the page...
  </body>
</html>
```

Saved in a *filename*.**html** file.

# Basic HTML page structure

(i.e. copy/paste boilerplate)

Metadata that doesn't appear in the viewport of the browser

```
<!DOCTYPE html>
<html>
  <head>
    <title>CS 193X</title>
  </head>

  <body>
    ... contents of the page...
  </body>
</html>
```

E.g. **`<title>`** shows up as the name of the tab

Contents that render in the viewport of the browser

# HTML elements

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

- An element usually has start and ending tags (**`<p>`** and **`</p>`**)
  - **content**: stuff in between start and end tags
- An element can be self-closing (**`img`**)
- An element can have attributes (**`src="puppy.jpg"`**)
- Elements can contain other elements (**`p`** contains **`em`** and **`img`**)

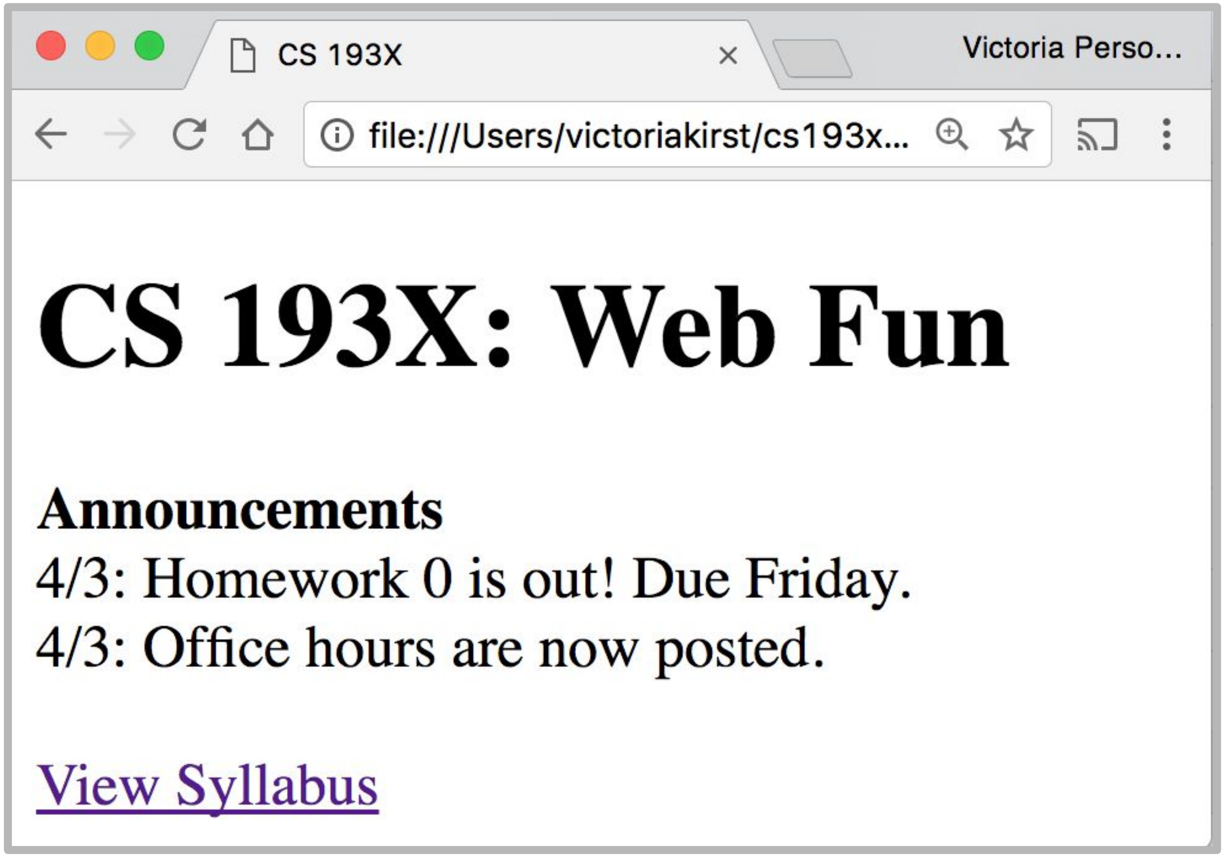
# Some HTML elements

(to place within `<body>`)

| | |
|---|---|
| Top-level heading<br>**`h1, h2, … h6`** | `<h1>Moby Dick</h1>` |
| Paragraph | `<p>Call me Ishmael.</p>` |
| Line break | `since feeling is first<br/>`<br>`who pays any attention` |
| Image | `<img src="cover.png" />` |
| Link | `<a href="google.com">click here!</a>` |
| Strong (bold) | `<strong>Be BOLD</strong>` |
| Emphasis (italic) | `He's my <em>brother</em> and all` |

# Exercise: Course web page

Let's write some HTML to make the following page:



# CS 193X: Web Fun

**Announcements**
4/3: Homework 0 is out! Due Friday.
4/3: Office hours are now posted.

View Syllabus

# Exercise: Course web page

**HTML boilerplate**

```
<!DOCTYPE html>
<html>
  <head>
    <title>CS 193X</title>
  </head>

  <body>
   ...
  </body>
</html>
```

**Plaintext contents of the page**

```
CS 193X: Web Fun

Announcements
4/3: Homework 0 is out! Due
Friday.
4/3: Office hours are now
posted.

View Syllabus
```

JSBin

# Solution

```
<!DOCTYPE html>
<html>
  <head>
    <title>CS 193X</title>
  </head>
  <body>
    <h1>CS 193X: Web Fun</h1>
    <strong>Announcements</strong><br/>
    4/3: Homework 0 is out!<br/>
    4/3: Office hours are now posted.<br/>
    <br/>
    <a href="http://cs193x.stanford.edu/syllabus">
      View Syllabus
    </a>
  </body>
</html>
```

# That was weird

- We saw that HTML whitespace collapses into one space…

```
<h1>CS 193X: Web Fun</h1>
<strong>Announcements</strong><br/>
4/3: Homework 0 is out!<br/>
```

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

Hmmm… strange…
Oh well, it works! Let's move on!!!

# CSS

# CSS

**CSS**: **C**ascading **S**tyle Sheets

- Describes the **appearance** and **layout** of a web page
- Composed of CSS **rules**, which define sets of styles

```
selector {
  property: value;
}
```

# CSS

A CSS file is composed of **style rules**:

```
selector {
  property: value;
}
```

*selector*: Specifies the HTML element(s) to style.
*property*: The name of the CSS style.
*value*: The value for the CSS style.

Saved in a *filename*`.css` file.

# CSS

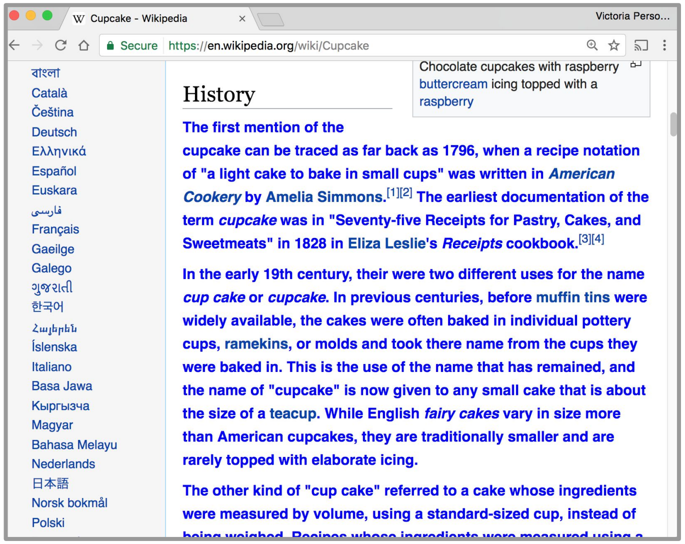
```
// NOT REAL CSS
fork {
  color: gold;
}
```

"All forks on the table should be gold"

# CSS

```
p {
  color: blue;
  font-weight: bold;
}
```

"All `<p>` elements on the page should be blue and bold"



Chocolate cupcakes with raspberry buttercream icing topped with a raspberry

## History

The first mention of the cupcake can be traced as far back as 1796, when a recipe notation of "a light cake to bake in small cups" was written in *American Cookery* by Amelia Simmons.[1][2] The earliest documentation of the term *cupcake* was in "Seventy-five Receipts for Pastry, Cakes, and Sweetmeats" in 1828 in Eliza Leslie's *Receipts* cookbook.[3][4]

In the early 19th century, their were two different uses for the name *cup cake* or *cupcake*. In previous centuries, before muffin tins were widely available, the cakes were often baked in individual pottery cups, ramekins, or molds and took there name from the cups they were baked in. This is the use of the name that has remained, and the name of "cupcake" is now given to any small cake that is about the size of a teacup. While English *fairy cakes* vary in size more than American cupcakes, they are traditionally smaller and are rarely topped with elaborate icing.

The other kind of "cup cake" referred to a cake whose ingredients were measured by volume, using a standard-sized cup, instead of being weighed. Recipes whose ingredients were measured using a

# Linking CSS in HTML

(i.e. copy/paste boilerplate)

```
<!DOCTYPE html>
<html>
  <head>
    <title>CS 193X</title>
    <link rel="stylesheet" href="filename.css" />
  </head>

  <body>
  ... contents of the page...
  </body>
</html>
```

# Some CSS properties

There are over 500 CSS properties! Here are a few:

| | |
|---|---|
| Font face (mdn) | `font-family: Helvetica;` |
| Font color (mdn) | `color: gray;` |
| Background color (mdn) | `background-color: red;` |
| Border (mdn) | `border: 3px solid green;` |
| Text alignment (mdn) | `text-align: center;` |

Aside: Mozilla Developer Network (MDN) is the best reference for HTML elements and CSS properties

- The actual W3 spec is very hard to read (meant for browser developers, not web developers)

# Main ways to define CSS colors:

**140 predefined names (list)**

```
color: black;
```

**rgb() and rgba()**

```
color: rgb(34, 12, 64);
color: rgba(0, 0, 0, 0.5);
```

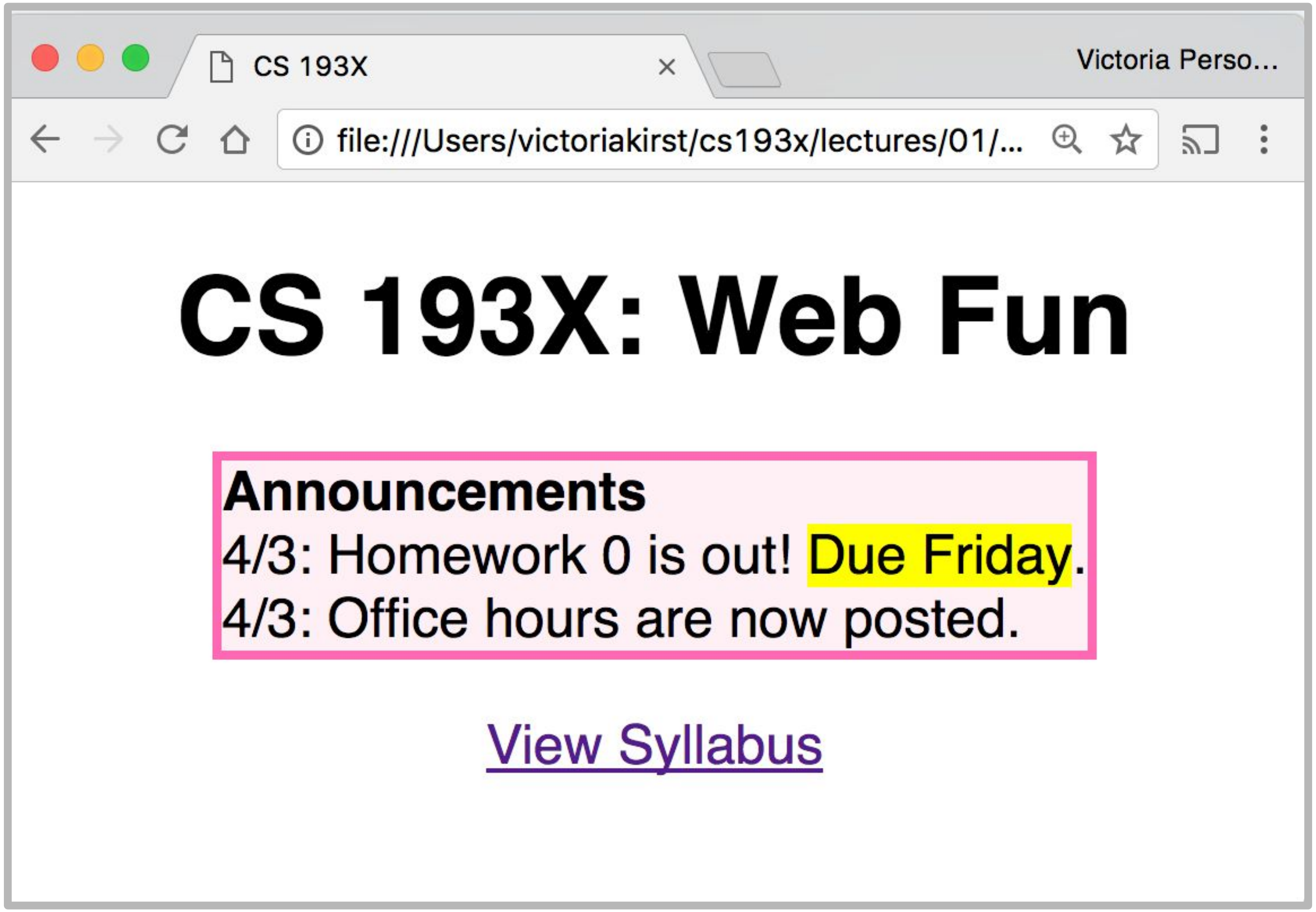
**Hex values**

```
color: #00ff00;
color: #0f0;
color: #00ff0080;
```

- The "a" stands for **alpha channel** and is a **transparency** value
- Generally prefer more descriptive over less:
  1. Predefined name
  2. rgb / rgba
  3. Hex

# Exercise: Course web page

Let's write some CSS to style our page:



CS 193X: Web Fun

**Announcements**
4/3: Homework 0 is out! Due Friday.
4/3: Office hours are now posted.

View Syllabus

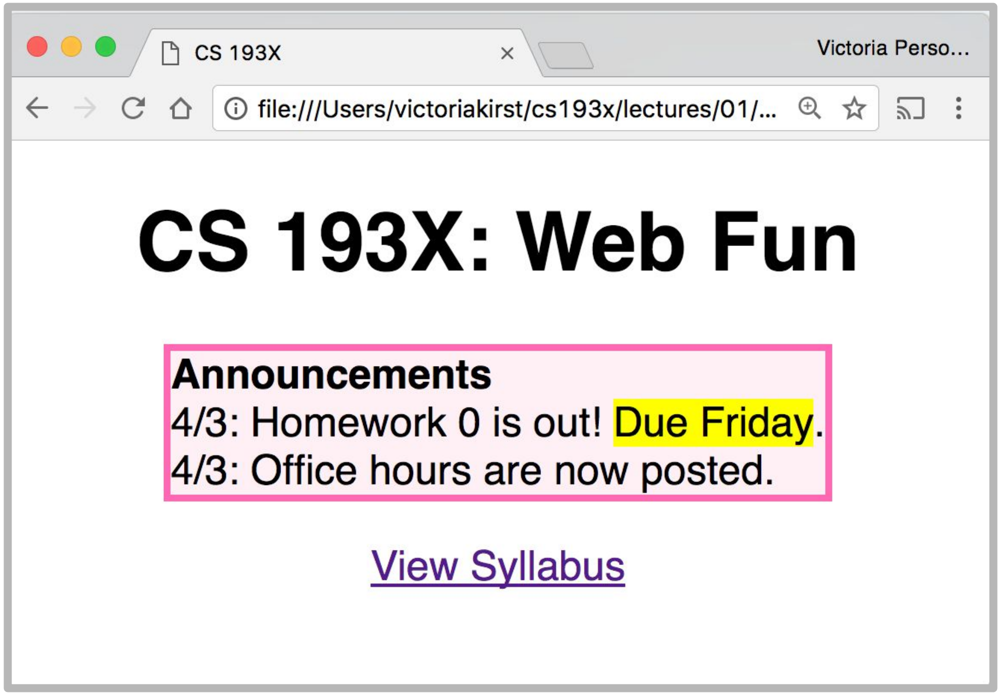
# Exercise: Course web page

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**



CS 193X: Web Fun

**Announcements**
4/3: Homework 0 is out! Due Friday.
4/3: Office hours are now posted.

View Syllabus

JSBin

# CSS exercise debrief

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.)

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

- Couldn't set `text-align: center;` to the `<a>` or `<strong>` tags directly, but could center `<p>` and `<h1>`
- Had to set a `width` on the box to make it hug the text … any other way to do this?
- How to center the box?! How do you highlight?!

# Q: Why is HTML/CSS so bizarre??

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

**block** vs **inline** display