

Welcome to CS193X

Michael Chang
Spring 2023

Plan for today

Course overview, logistics, syllabus

Context: the Internet

Servers, web browsers, accessing a web page

Introduction to HTML

File structure, tags

Elements, attributes

Introduction to JavaScript

Our Approach

Web dev is a huge space

Constantly changing, often a bit slow to converge

Many ways to solve the same problem

We will be "opinionated"

Show you one way to do things

Won't necessarily be best, but it will be one we've seen a lot, based on reliable sources

No frontend framework

By the end, we hope you can pick up new frameworks and tech quickly

CS142 vs. CS193X

CS142

Focus on systems principles

Uses web dev as tool to talk about larger systems

Uses frontend framework (React) to build larger apps

Well-established class, stable materials

Prereq: 107

Counts toward undergrad major

CS142 vs. CS193X

CS193X

"Practical skills" class

Focus on core concepts in web dev, smaller apps

No frontend framework, modern standardized tech

Newer course, materials in flux

Prereqs: 106B

Does not count toward undergrad program sheet

Course website

<https://cs193x.stanford.edu>

Lectures

Lots of live coding

Materials

Will post slides/notes and code before lecture

Will post final version of code after lecture

Getting Help

Ed forum

Office hours

Will post calendar tonight, start tomorrow or Thursday

Email

For personal/private issues

Coursework

Assignments

~5 assignments + setup/survey (assignment 0)

Roughly 1-1.5 weeks apart

Individual

Project

Your chance to be creative, make something cool

Proposal around week 6, milestone check-in week 8

Final submission week 10/finals week

Individual

Grading

Bucket scale, like 106s

Functionality and style, equal weight

Buckets

+: met all requirements, great work

ok: solid work, some opportunities for improvement

-: more significant issue, multiple smaller issues

--: incomplete or multiple major issues

0: no submission

Grading

Late policy

Most assignments and milestones have a “grace period”

- Submissions by the due date receive a small “on time bonus”

- Submissions within the grace period are not penalized

- No submissions after the grace period

Intended to function like an automatically granted extension

Some exceptions (e.g. assign0, end of quarter)

Honor Code and Collaboration

Please read the **collaboration policy** fully

Summary

Write your own code

Do your own design and debugging

But please do solicit and offer tips

Seek clarification, discuss course topics

Look up background info, references, documentation

Cite your sources of inspiration, ideas, fixes

Topic outline

Client-side JavaScript (3 weeks)

Syntax, data structures, page interactions, events

Web page layout: HTML and CSS (1.5 weeks)

APIs, dynamic content

APIs and server-side JavaScript (2.5 weeks)

fetch, async/await

Creating an API

Data storage with MongoDB

Full stack topics (2 weeks)

Mobile and accessible design

User authentication

Plan for today

Course overview, logistics, syllabus

Context: the Internet

Servers, web browsers, accessing a web page

Introduction to HTML

File structure, tags

Elements, attributes

Introduction to JavaScript

The Internet

When you type an address (URL) in a browser

Find the server that has the web page

Ask the server for the page

Display the page

Servers and Ports

Each server (computer) can offer many things

Web, email, remote login (SSH)

How to know which program a request belongs to

Each program listens on a "port" (a number)

Client says which port it wants to talk to

Standard ports for web, mail, etc.

Can use custom ports for more unusual services, internal use only, etc.

Protocols

How do the client and server "talk?"

Protocol: set of "messages" written in a standard

Any program that "speaks" a protocol can communicate with any other

E.g. different browsers, different web servers

Example (demo): HTTP

Why talk about this now

We could technically just double-click web pages to open them

But we'll run into many problems later

Instead, we will run a web server on localhost

Provided in the starter code

localhost: always refers to the local machine

Others can't access your server (for security)

Custom port

Plan for today

Course overview, logistics, syllabus

Context: the Internet

Servers, web browsers, accessing a web page

Introduction to HTML

File structure, tags

Elements, attributes

Introduction to JavaScript

Demo: Intro HTML

See website for link to demo code

Takeaways: HTML

HTML is a markup language

Not a program, doesn't define logic

Structured as a tree

element: a node in the tree

Has a parent, most can have children

Can contain text

tag: marks the start or end of a node

E.g. `<p> ... </p>`

Takeaways: JavaScript

DevTools / JavaScript console

Should basically always have this open while working

JavaScript syntax

C/C++-like (braces, semicolons)

Variables with `let`, `if`, `while`, `for` (`let x of y`)

Functions: `const foo = (arg1, arg2) => { ... }`

Summary

For next time

Assign0 will go up tonight (due next Tue)

Join the Ed forum

See website for materials, OH calendar, etc.

Next time: more JavaScript

Arrays, objects (maps)

Classes