

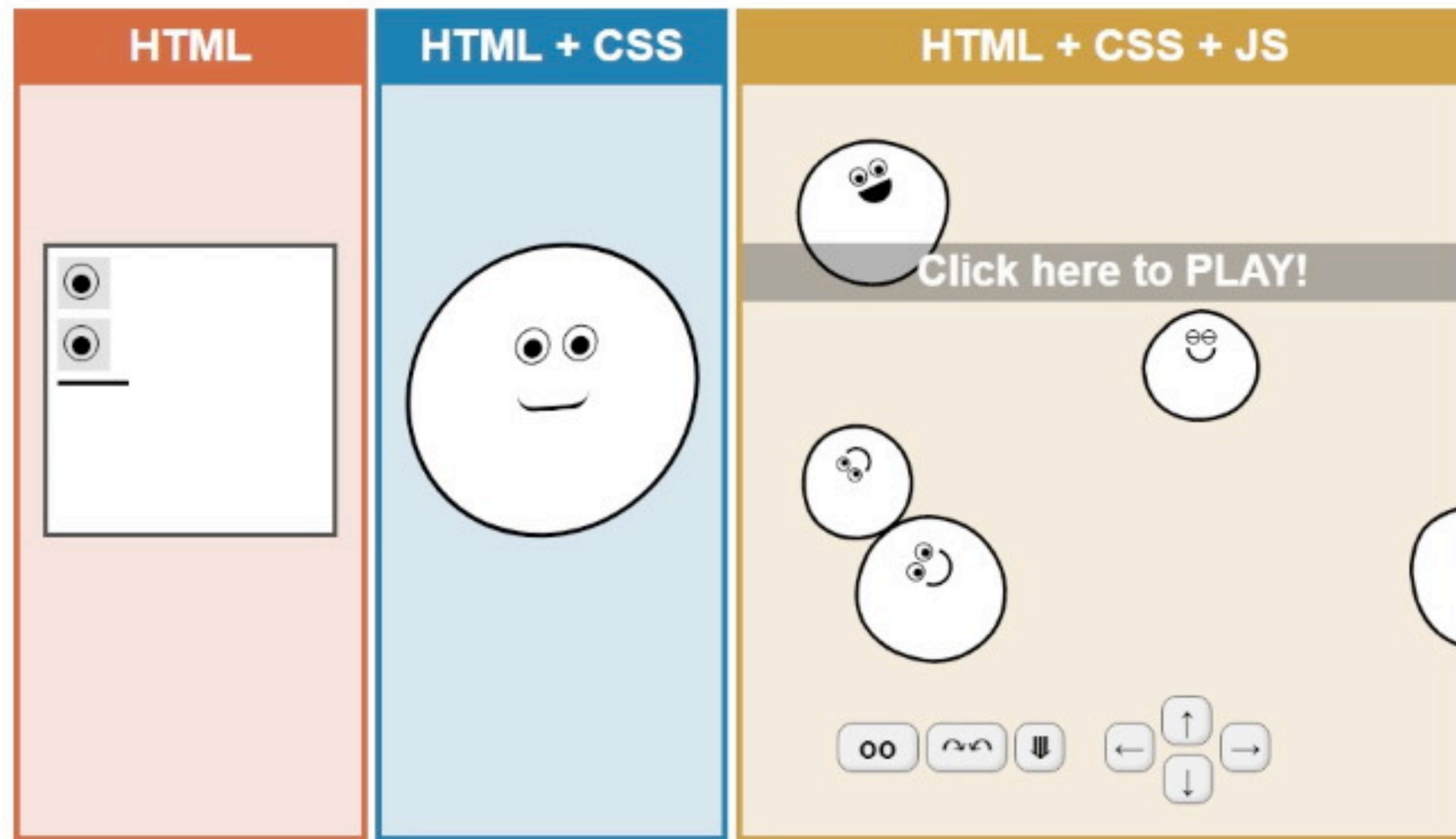
CS181DT Class 20: (new) web dev 101 & project OH



Class 20 agenda

- 5C Hack announcement & zipcrit
- Web dev 101
- Project OHs

Web Dev 101



HTML creates the content

JS makes things interactive

CSS stylizes the content

Static website DOM example

Jingyi Li

About

Research

Teaching

Projects

Personal

Hi, I'm Jingyi!

I'm a computer science educator and human-computer interaction researcher



I'm an Assistant Professor of Computer Science at Pomona College, where I direct the Doodle Lab. My work focuses on understanding, designing, and critiquing creativity support tools. I believe in the liberatory power of every day making and investigate how computation can augment these practices without displacing existing manual skills.

My PhD is from Stanford University where I was co-advised by Sean Follmer and Maneesh Agrawala. I did my undergrad at UC Berkeley, where I worked with Björn Hartmann.

p.about-text | 396 x 308 | Flex Item

Research



Beyond the artifact: power as a lens for creativity support tools

Jingyi Li, Eric Rawn, Jacob Ritchie, Jasper Tran O'Leary, Sean Follmer

When we see a creativity support tool (CST), we should see a power relationship. From interviews with 11 creative practitioners and tool designers, we build a preliminary theory of how power relationships can manifest in CSTs, and what we could do about it.

Pronouns: **They/them**
✉ firstname.lastname@pomona.edu
🐦 [@jingyitweets](https://twitter.com/jingyitweets)
🎓 [Google Scholar](#) / [CV](#)

The screenshot shows the browser's developer tools interface. The DOM Inspector on the left shows the HTML structure, with the selected element being a paragraph with class 'about-text'. The Styles pane in the middle shows the CSS rules for this element, including margin, padding-left, and font settings. The Layout pane on the right shows a visual representation of the box model for the element, with dimensions of 396x308 pixels and a flex item status.

Static website DOM example

- `console.log()` to print in Javascript
- console can inspect DOM elements
- use JS to select and manipulate DOM elements (by `#class` or by `.id`)
- use CSS to style DOM elements

More complicated example: next.js drawing tool with React and TypeScript

The image shows a screenshot of a web browser displaying the GitHub repository for 'flatdraw' by Diogo Capela. The repository page includes a list of files such as .env.mrc, .prettiignore, .prettierrc, LICENSE, README.md, next.config.js, package-lock.json, package.json, and tsconfig.json. The README section describes 'Flatdraw' as a simple canvas drawing web application with a responsive UI, built using TypeScript, React, and Next.js. A live demo link is provided: flatdraw.com. Below the text is a screenshot of the Flatdraw application interface, which features a toolbar on the left with various drawing tools (pen, eraser, selection, etc.), a color picker, and a central canvas area with a red bounding box around the text 'Add text'.

On the right side of the screenshot, a code editor displays the source code for a file named '_app.tsx'. The code is a Next.js page component that imports various libraries and hooks, including Mantine, Next.js, and React. It defines a RouterTransition function and uses useEffect to handle route changes. The code is as follows:

```
1 import { MantineProvider, ColorSchemeProvider, type ColorScheme } from '@mantine/core';
2 import { ModalsProvider } from '@mantine/modals';
3 import { Notifications } from '@mantine/notifications';
4 import { NavigationProgress, nprogress } from '@mantine/nprogress';
5 import type { AppProps } from 'next/app';
6 import NextHead from 'next/head';
7 import { useRouter } from 'next/router';
8 import { GoogleAnalytics } from 'nextjs-google-analytics';
9 import React, { useEffect, useState } from 'react';
10
11 import '~/theme/styles/global.css';
12 import LoadingOverlay from '~/components/LoadingOverlay';
13 import metadata from '~/config/metadata';
14 import { DEFAULT_COLOR_SCHEME } from '~/config/settings';
15 import { CanvasContextProvider } from '~/context/useCanvasContext/useCanvasContext';
16 import { ColorSchemeContextProvider } from '~/context/useColorSchemeContext/useColorSchemeContext';
17 import { ModalContextProvider } from '~/context/useModalContext/useModalContext';
18 import useCookies from '~/hooks/useCookies';
19 import useAvailableFonts from '~/store/useAvailableFonts';
20 import theme from '~/theme';
21 import colors from '~/theme/colors';
22 import { GlobalStyle } from '~/theme/styles/global';
23 import getAvailableFonts from '~/utils/getAvailableFonts';
24
25 function RouterTransition() {
26   const router = useRouter();
27
28   useEffect(() => {
29     const handleStart = (url: string) => url !== router.asPath && nprogress.start();
30     const handleComplete = () => nprogress.complete();
31
32     router.events.on('routeChangeStart', handleStart);
33     router.events.on('routeChangeComplete', handleComplete);
34     router.events.on('routeChangeError', handleComplete);
35
36     return () => {
37       router.events.off('routeChangeStart', handleStart);
38       router.events.off('routeChangeComplete', handleComplete);
39       router.events.off('routeChangeError', handleComplete);
40     };
41     // eslint-disable-next-line react-hooks/exhaustive-deps
42   }, [router.asPath]);
43 }
```

<https://github.com/diogocapela/flatdraw?tab=readme-ov-file>

Workflow for more complex libraries

- Don't reinvent the wheel!
- 1. Search around for existing Github repos you can use as examples, or tutorials you can follow to get close to what you want
- 2. Search around for existing libraries/modules you can integrate to expand this example
- I won't teach next.js or node.js specifics in class (and I'm not an expert either), but you can come to OH for help. And you're not required to use them
- It's OK to use ChatGPT to ask what existing code does or to help you write code
 - Don't forget about StackOverFlow!

Class 20 recap

- Exit ticket: <http://tiny.cc/cs181dt-week11>
- TODO
 - Next Tuesday: System diagram
 - Zipcrits from Abrar (Tues) & Larry (Thurs)
 - Four seminars next week; 2 sketchnotes
 - Tues: Shuyan & Keya; Vivian & Abrar
 - Thurs: Kovit & Cassidy; Vitor & Rediet



Project OH

- I'll come by and we can (1) test your WoZ (if we haven't yet already) and (2) talk about implementation suggestions or woes
- Group order
 - Shape composition
 - Story nodes
 - Nail design
 - Music visualizer
 - Music map