Welcome to CS 106S!

Introduction to CS for Social Good, our map for the quarter, and JavaScript!

Ben Yan, Spring 2025 🌸

cs106s.stanford.edu

Stanford | ENGINEERING
Computer Science

Welcome to the First Day of Class!

Hope that you had a wonderful spring break!! 🌱

Hi I'm Ben!

- ❑ **'23 – '26: CS MS/Co-Term Student (AI Track)**, Jerry is my advisor
- ❑ '20 – '24: Just finished my undergrad here! Double-Majored in CS & Math, Minor in Creative Writing—also studied abroad at Oxford my senior winter

- ❑ Prev. SWE @ NVIDIA (CUDA Systems), did CS research for 2ish years
- ❑ **Life rn is more teaching-oriented:** CS106AX (🍁 Head TA), CS107 (TA last winter ☃️ + this spring 🌸!), CS106S (have taught it 4x now and **I love teaching it!**)
- ❑ This'll be my last time teaching CS106S, and so so excited that you're here!

Stanford CS Course Advisor · Follow
May 13, 2022 ·
Benjamin Yan declared CS today! Congratulations, Benjamin!

🎨 Interests, what brings me joy

❑ **Anime & Manga** (my all-time fav *Jujutsu Kaisen*), Spider-Verse, Gravity Falls, been getting into J- and K-dramas very recently

❑ **Poetry & novel writing** – once wrote 2 novels (50k words each) in one month for NaNoWriMo 2023 / English 190E lol

❑ I haven’t written much since :(

❑ **Music:** Charli XCX, Olivia Rodrigo, YOASOBI

❑ Not my Spotify wrapped though – it’s embarrassing

❑ **Books:** Anything by Ocean Vuong

❑ Always looking for recs!

Intros!

- ❑ Name & pronouns if you’re comfortable sharing!
- ❑ What you’re studying / thinking about studying
- ❑ Year
- ❑ Fun fact 😨😱 or **any one of the following!**
    - ❑ What are you looking forward this quarter? 🌲
    - ❑ Something you did over the spring break 🌸
    - ❑ Music / book / show recommendations? 🎶
    - ❑ Anything else you’d like to share! :)

The Map For Today

1 syllabus & logistics

2 getting set up for the class

3 HTML/CSS/javascript basics

4 caesar ciphers!

```
const LOCALE = globalThis.navigator.language

const div = document.body.appendChild(document.createElement('div'))
const list = div.appendChild(document.createElement('ol'))

const dayNames = new Map()

for (let i = 0; i < 7; ++i) {
    const d = Temporal.PlainDate.from({
        year: Temporal.Now.plainDateISO().year,
        month: 1,
        day: i + 1,
    })

    dayNames.set(d.dayOfWeek, d.toLocaleString(LOCAL
}

for (const num of [ ... dayNames.keys()].sort((a, b) =
    list.appendChild(Object.assign(
        document.createElement('li'),
        { textContent: dayNames.get(num) },
    ))
}
```

cs106s.stanford.edu

⛵ Course Mechanics

- ❑ 1 unit, S/NC
- ❑ **Attendance (7/9*)**
  - ❑ Relaxed, workshop-style environment
  - ❑ **Brief check-off forms**
- ❑ Canvas for announcements
- ❑ Questions welcome!

***Please do reach out to us if difficult circumstances arise! We understand life can be very stressful and challenging, and will always create a path for you to pass 106S.**

🌐 Course Website!

cs106s.stanford.edu

📬 Contact Email

bbyan@stanford.edu

🧭 Place & Time

🎨 Lathrop, Room 180

Thursday, 4:30 - 6:20 PM; usually try to keep class to 90 minutes ish

👋 Office Hours

After class, or email/Slack me!

No assignments in 106S, but happy to chat about material, CS, Stanford, anything!

Course Schedule

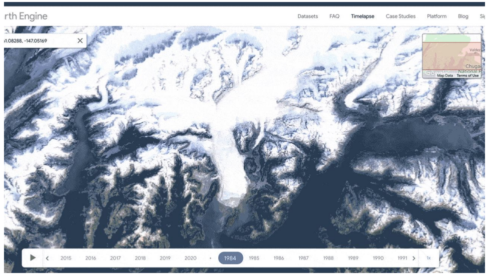
To those fleeing persecution, terror & war, Canadians will welcome you, regardless of your faith. Diversity is our strength #WelcomeToCanada

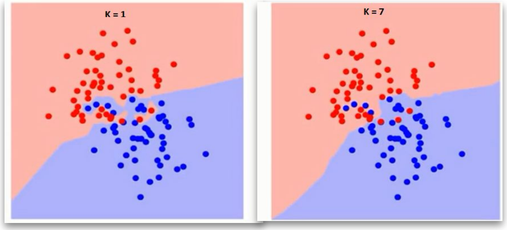
RETWEETS 165,284 LIKES 256,250

12:20 PM - 28 Jan 2017

Week 1	Apr 3	Intro, JavaScript, Ciphers
Week 2	Apr 10	Sentiment Analysis & Refugee Tweets
Week 3	Apr 17	CS for Climate Change
Week 4	Apr 24	Cancer Detection with KNN
Week 5	May 1	Cybersecurity and Ethical Hacking
Week 6	May 8	Open Source & Web Software
Week 7	May 15	Trust & Safety
Week 8	May 22	Mental Health
Week 9	May 29	What's Next – Beyond 106S, End-Term Boba Party 🧋
Week 10	Jun 6	**No class; good luck on your finals!** 🍀

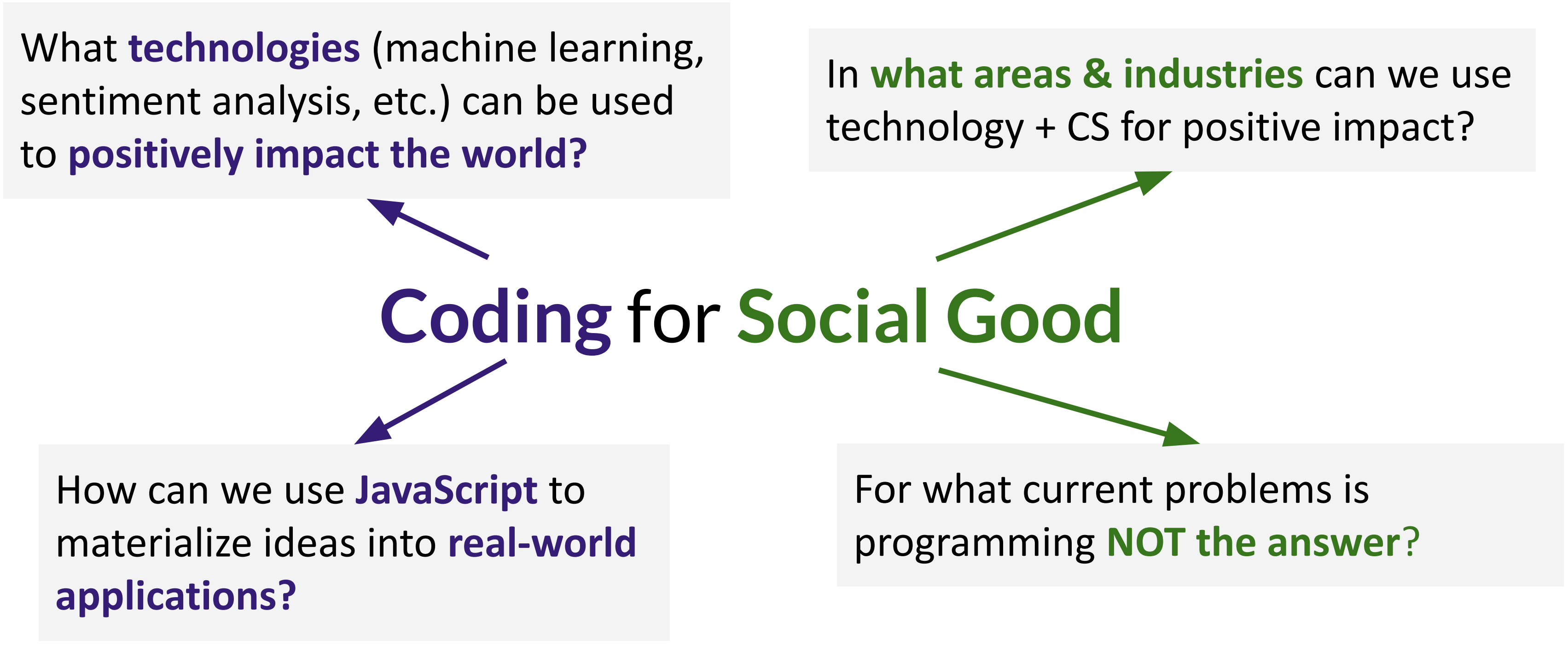
Subject to change – please let me know if you have any feedback or suggestions at any point! Happy to run this class in the way it'd be most helpful to you





```
Welcome to
                 EEEEEE  LL      IIII  ZZZZZZZ   AAAAA
                 EE      LL       II       ZZ   AA   AA
                 EEEEE   LL       II     ZZZ    AAAAAAA
                 EE      LL       II    ZZ      AA   AA
                 EEEEEE  LLLLLL  IIII  ZZZZZZZ  AA   AA

ELIZA is a mock (Rogerian) psychotherapist.
The original program was described by Joseph Weizenbaum in 1966.
This implementation ('elizabot.js') by Norbert Landsteiner 2005.
Graphics and real-time text to speech integration added in 2013.

Note: Try Google Chrome to enjoy the true thrills of speech recognition.


VOICE SETUP
Please choose an accent to be used by ELIZA (speech output):

   [1]   English - US
   [2]   English - EN

>
```

Overview of Classes!



What **technologies** (machine learning, sentiment analysis, etc.) can be used to **positively impact the world?**

In **what areas & industries** can we use technology + CS for positive impact?

Coding for **Social Good**

How can we use **JavaScript** to materialize ideas into **real-world applications?**

For what current problems is programming **NOT the answer**?

Let’s Dive In!

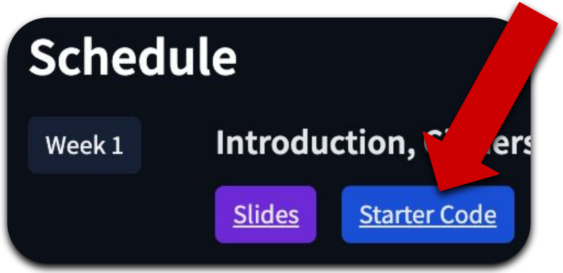
install Chrome

Getting Set Up

install VS Code

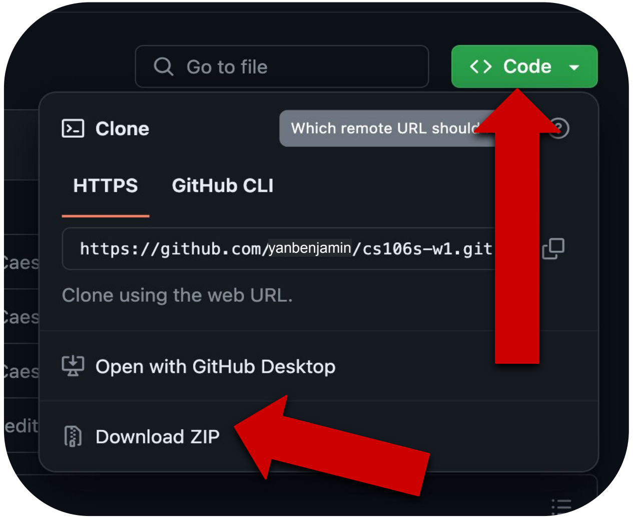
(or an editor of your choice,
Sublime Text is also great)

Opening the Starter Code



1 Navigate to Week 1 of the Schedule section of **cs106s.stanford.edu**

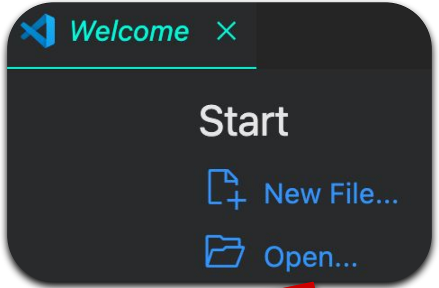
Also, at this link: **https://github.com/yanbenjamin/cs106s-w1**



```
https://github.com/yanbenjamin/cs106s-w1.git
```

Clone using the web URL.

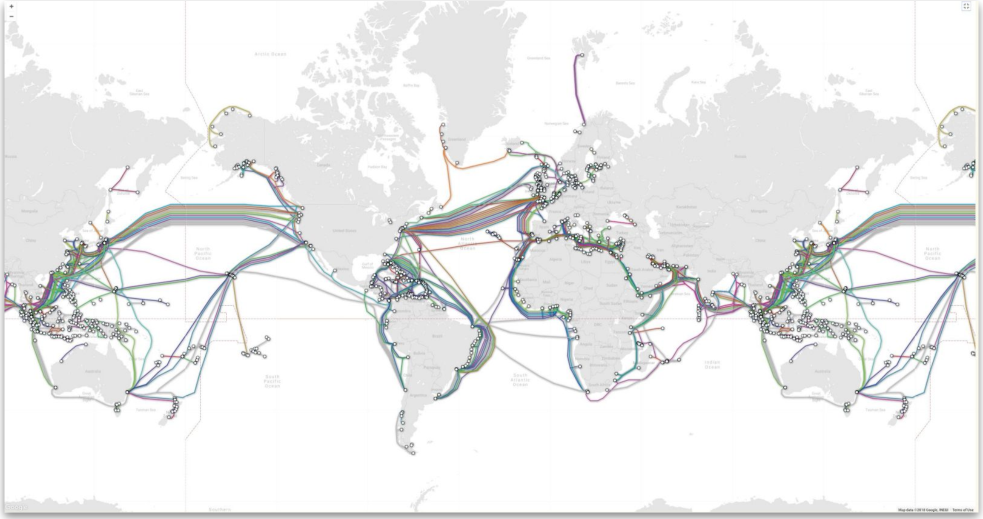
2 Click the bright **"Code"** button, then click "Download ZIP"



In VSCode

3 Unzip the download (clicking .zip file should do the trick) and open the folder / files in your editor

HTML, CSS, JS Overview



HTML, CSS, JS Overview

.html	Hypertext Markup Language
.css	Cascading Style Sheets
.js	JavaScript

- ❑ HTML for defining the **webpage content and basic structure**
- ❑ CSS for **regulating style and formatting**
- ❑ JavaScript for **enabling the HTML/CSS components on the webpage to be interactive**
  - ❑ 🌐 “Language of the Web”
  - ❑ 🔗 99% of websites use JavaScript on the client side, making it essential for building browser applications

HTML Layout

index.html

```
<!doctype html>
<html>
   <head>
       <link rel="stylesheet" href="style.css">
   </head>
   <body>
       <h2>CS 106S Week 1: JavaScript and Cryptography</h2>
       <img src = "hello_cat.jpg" width = 400>
       <p>Open the JavaScript console to continue onward!</p>
   </body>
</html>
```

Note: CS 106S isn’t a dedicated web development course — but I think it’s helpful to at least cover the basics!

hello_cat2.jpg

hello_cat3.jpg

hello_cat4.jpg

hello_cat.jpg

hello_cat5.jpg

hello_cat6.jpg

hello_cat7.jpg

index.html

```
<!doctype html>
<html>
   <head>
       <link rel="stylesheet" href="style.css">
   </head>
   <body>
       <h2>CS 106S Week 1: JavaScript and Cryptography</h2>
       <img src = "hello_cat.jpg" width = 400>
       <p>Open the JavaScript console to continue onward!</p>
   </body>
</html>
```

🌐 **The <html> and </html> tags enclose all the content.**

index.html

```
<!doctype html>
<html>
   <head>
       <link rel="stylesheet" href="style.css">
   </head>
   <body>
       <h2>CS 106S Week 1: JavaScript and Cryptography</h2>
       <img src = "hello_cat.jpg" width = 400>
       <p>Open the JavaScript console to continue onward!</p>
   </body>
</html>
```

🌐 **HEAD contains info not displayed on webpage (e.g., browser title, browser icon, any JavaScript or CSS style files to load)**

index.html

```
<!doctype html>
<html>
   <head>
       <link rel="stylesheet" href="style.css">
   </head>
   <body>
       <h2>CS 106S Week 1: JavaScript and Cryptography</h2>
       <img src = "hello_cat.jpg" width = 400>
       <p>Open the JavaScript console to continue onward!</p>
   </body>
</html>
```

🌐 **BODY contains everything displayed on the webpage (e.g., text, section headings, images, GIFs, etc)**

index.html

```
<!doctype html>
<html>
   <head>
       <link rel="stylesheet" href="style.css">
   </head>
   <body>
       <h2>CS 106S Week 1: JavaScript and Cryptography</h2>
       <img src = "hello_cat.jpg" width = 400>
       <p>Open the JavaScript console to continue onward!</p>
   </body>
</html>
```

🌐 **Tags such as <h2> enclose each of the HTML elements. Typically have end tag (</h2>), but not always (<img>, <br> – line break)** index.html

```
<!doctype html>
<html>
   <head>
       <link rel="stylesheet" href="style.css">
   </head>
   <body>
       <h2>CS 106S Week 1: JavaScript and Cryptography</h2>
       <img src = "hello_cat.jpg" width = 400>
       <p>Open the JavaScript console to continue onward!</p>
   </body>
</html>
```

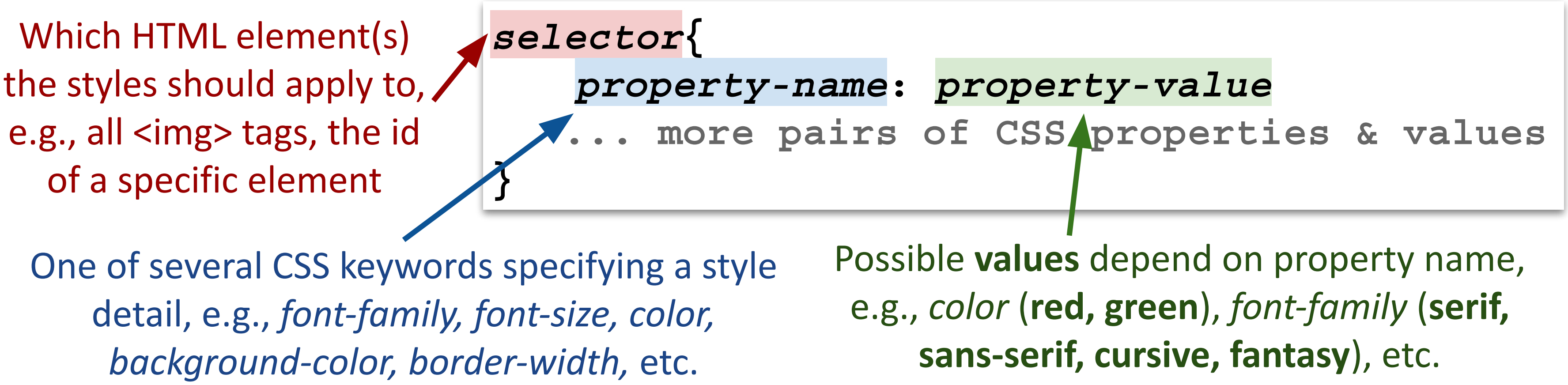
🌐 **Question: How can we stylize each of these webpage elements embedded in tags? (e.g., use different colors, fonts, spacing)** index.html

```
<!doctype html>
<html>
   <head>
       <link rel="stylesheet" href="style.css">
   </head>
   <body>
       <h2>CS 106S Week 1: JavaScript and Cryptography</h2>
       <img src = "hello_cat.jpg" width = 400>
       <p>Open the JavaScript console to continue onward!</p>
   </body>
</html>
```

🌐 **Strategy: We use a separate CSS file to specify stylization, layout, font, colors, etc. (HTML → content, CSS → style)**

```css
style.css
img{
    border-style: dashed;
    border-width: 5px;
} /* adds dashed border with 5 pixel width to images */
h2{
    color: darkblue;
} /* sets section heading (<h2>) to a dark blue color */
```

🌐 In a CSS file, style rules are written in the form below:



```
selector{
    property-name: property-value
   ... more pairs of CSS properties & values
}
```

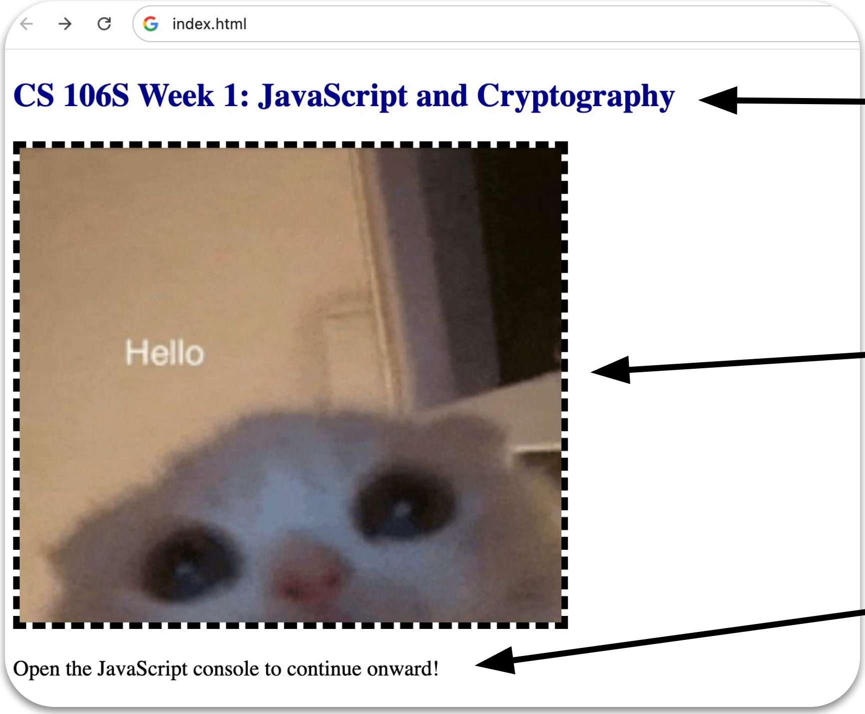
Which HTML element(s) the styles should apply to, e.g., all <img> tags, the id of a specific element

One of several CSS keywords specifying a style detail, e.g., *font-family, font-size, color, background-color, border-width,* etc.

Possible **values** depend on property name, e.g., *color* (**red, green**), *font-family* (**serif, sans-serif, cursive, fantasy**), etc.

HTML/CSS – Browser Rendering

Resulting webpage from **index.html** and **style.css**



CS 106S Week 1: JavaScript and Cryptography

Open the JavaScript console to continue onward!

Heading **<h2>** tag, with dark blue color from CSS

<img> tag, loading in image `hello_cat.jpg`, stylized with dotted 5-pixel-wide frame

Text in paragraph tag **<p>**

What JavaScript does

- JavaScript is a **powerful 'interface' with an HTML webpage,** enabling HTML elements to be programmatically accessed and modified.
  - Empowers **dynamic web apps** that respond to user interactions.
- To get a high-level sense of what JavaScript can do (before we dive into a starter tutorial on its features & syntax), it can:
  - Append, remove, or modify any HTML nodes i.e. tags/elements

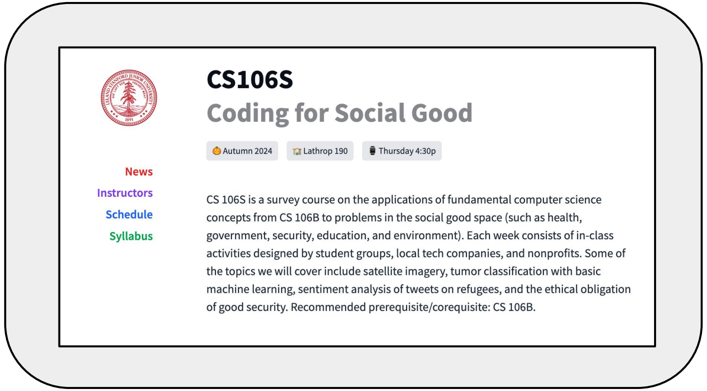
    Open pop-up window | Load new images | Toggle display to night mode
  - React to user events/actions with parts of the webpage, e.g., buttons

    Button clicks | Hovering mouse over image | Keyboard key is pressed
- JS is now quite **general-purpose**! Has evolved beyond its web roots.

Any questions so far?

What is index.html?

- In the starter code, you'll find a file named **index.html**; using Finder or your OS equivalent, **open it in Google Chrome**
- This is the **homepage** of a website.



CS106S

Coding for Social Good

Autumn 2024 | Lathrop 190 | Thursday 4:30p

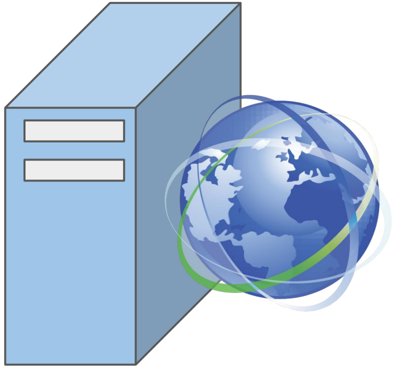
News
Instructors
Schedule
Syllabus

CS 106S is a survey course on the applications of fundamental computer science concepts from CS 106B to problems in the social good space (such as health, government, security, education, and environment). Each week consists of in-class activities designed by student groups, local tech companies, and nonprofits. Some of the topics we will cover include satellite imagery, tumor classification with basic machine learning, sentiment analysis of tweets on refugees, and the ethical obligation of good security. Recommended prerequisite/corequisite: CS 106B.

Client Browser

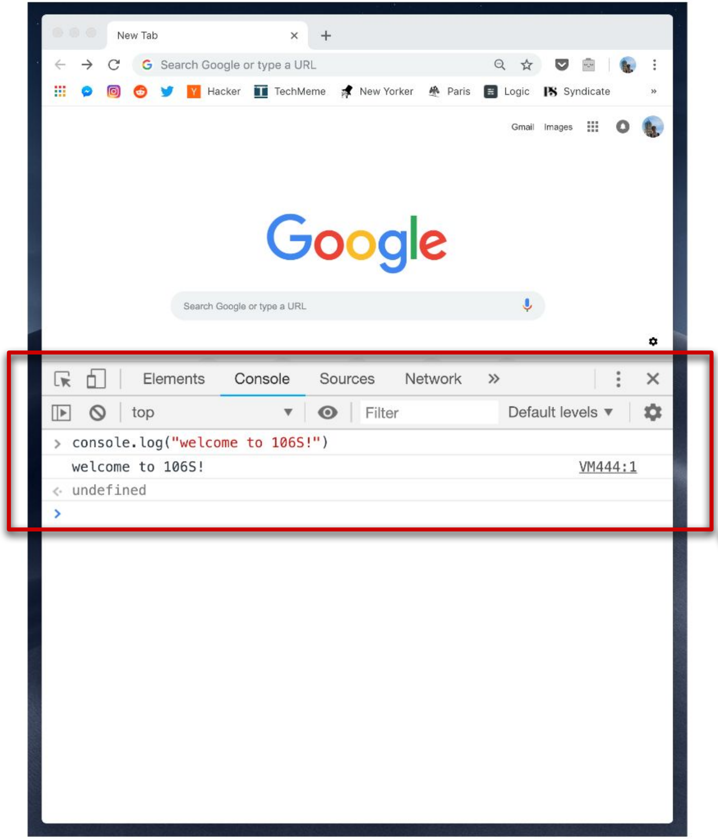
HTTPS Request for **https://cs106s.stanford.edu/** →

← HTTPS Response of **index.html**



Web Server

JavaScript in Chrome



```
> console.log("welcome to 106S!")
  welcome to 106S!                          VM444:1
<· undefined
>
```

1. Open **index.html** in Chrome
2. On Mac: Press **cmd** – **option** – **j**

   On Windows: Press **ctrl** – **shift** – **j**

Don't let go of the previous key while pressing the next!

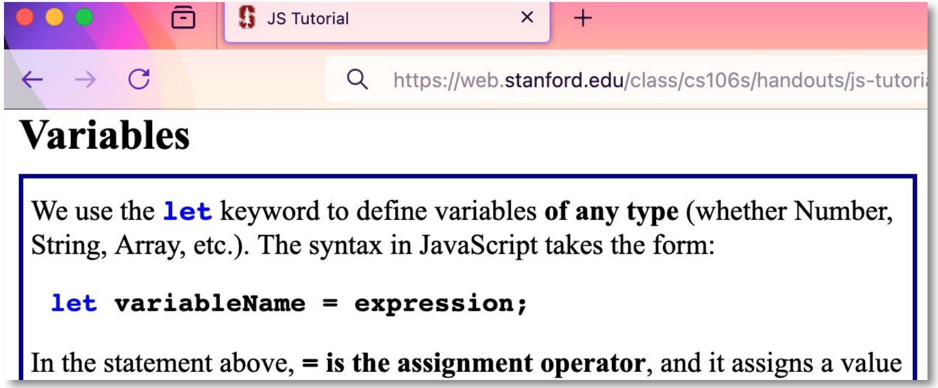
Here, **in the console that pops up**, we can input and run JavaScript code!

Onto the JavaScript Tutorial!

Inspect the file `tutorial.js` in your code editor.

To experiment around, copy & paste, and tinker with / run the JavaScript commands in the Chrome console!

More fleshed-out / lecture-notes-style version at the **link below!**



Variables

We use the **let** keyword to define variables **of any type** (whether Number, String, Array, etc.). The syntax in JavaScript takes the form:

```
let variableName = expression;
```

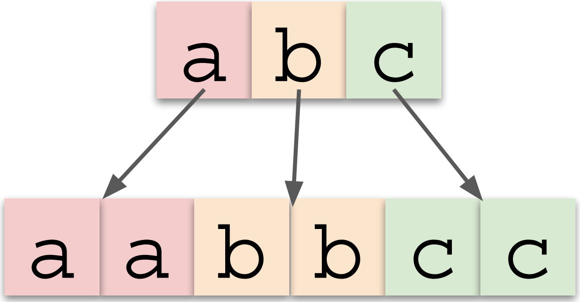
In the statement above, **= is the assignment operator**, and it assigns a value

https://web.stanford.edu/class/cs106s/handouts/js-tutorial.html

Overview: General JavaScript Syntax

Example: Consider a function **twins** which takes an input string and returns a copy of that string with each character repeated once, e.g.,

- "abc"→"aabbcc"
- minnesota"→"mmiinnnneessoottaa".



```
def twins(str): #Python
  result = ""
  for char in str:
      result += char + char
  return result
```

```
function twins(str){ //JavaScript
    let result = "";
    for (let char of str){
        result += char + char;
    }
    return result;
}
```

Overview: General JavaScript Syntax

```
function twins(str){
   let result = "";
   for (let char of str){
       result += char + char;
   }
   return result;
}
```

Overview: General JavaScript Syntax

```
function twins(str){
    let result = "";
    for (let char of str){
        result += char + char;
    }
    return result;
}
```

Unlike Python (blocks of code are defined by indentation), JavaScript uses **curly braces** – similar to C++ / Java.

```
function twins(str){
    let result = "";
    for (let char of str){
        result += char + char;
    }
    return result;
}
```

In Python, having **parentheses** around statements / conditions in `for, while,if` loops is optional.

In JavaScript, it's mandatory. Here's a **for/of** loop that iterates directly over the characters of a string.

Overview: General JavaScript Syntax

```
function twins(str){
    let result = "";
    for (let char of str){
        result += char + char;
    }
    return result;
}
```

In JavaScript, to declare variables, we use the `let` keyword (or `const`).

In Python, we could have simply written

`result = “”`

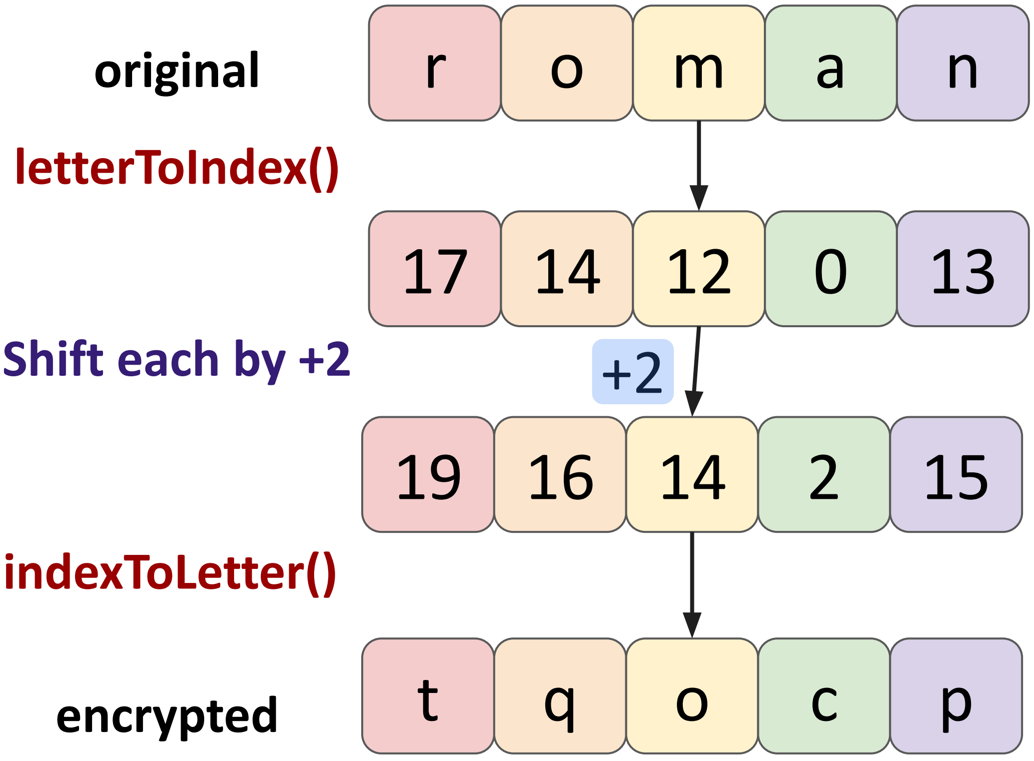
```
function twins(str){
    let result = "";
    for (let char of str){
        result += char + char;
    }
    return result;
}
```

Unlike C++ or Java, **semi-colons (;) are optional** at the end of statements.

But you can include them if you’d like! :) It’s a style decision.

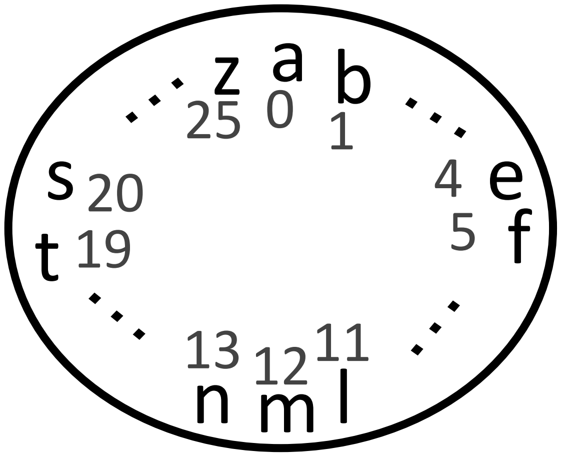
JavaScript is a “vibes”-based language, sort of ✨✨💅

🔐 Today's Coding – Caesar Ciphers



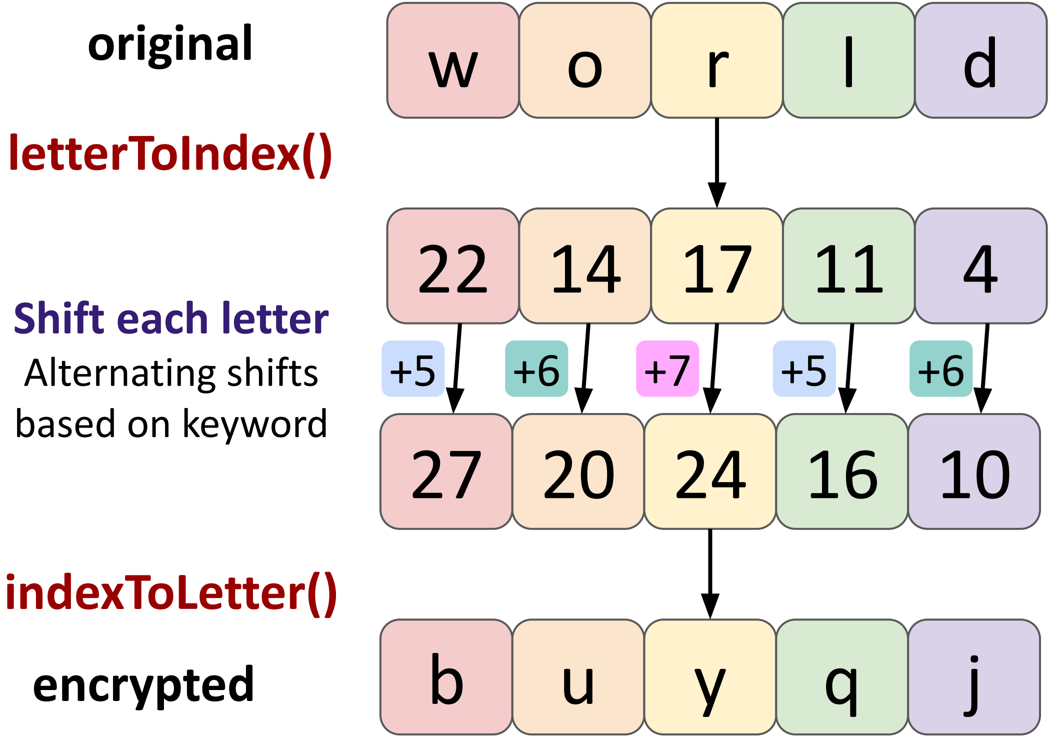
original	r	o	m	a	n
letterToIndex()					
	17	14	12	0	13
Shift each by +2		+2			
	19	16	14	2	15
indexToLetter()					
encrypted	t	q	o	c	p

Shift = +2



(Optional) Vigenère Ciphers

Feel free to try out this extension if you have time!



original	w	o	r	l	d
letterToIndex()					
	22	14	17	11	4
Shift each letter Alternating shifts based on keyword	+5	+6	+7	+5	+6
	27	20	24	16	10
indexToLetter()					
encrypted	b	u	y	q	j

Keyword

f	g	h
+5	+6	+7

Remark: It's like having multiple Caesar ciphers in one encryption! A rotating set of keys.

Checkpoint #1 assignment.js

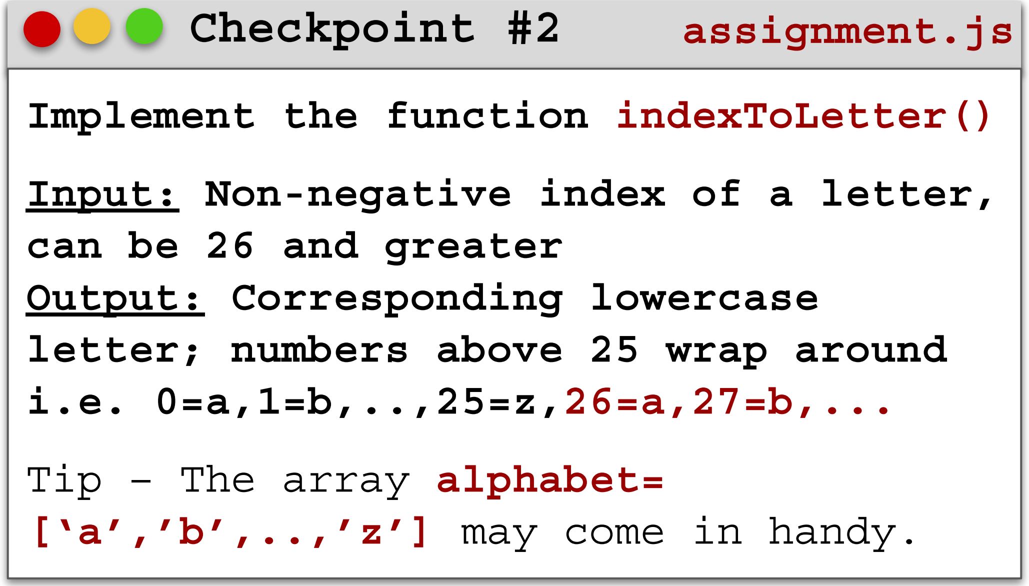
Implement the function letterToIndex()

Input: **A lowercase letter (a-z)**

Output: **Index in alphabet (a=0,b=1,c=2,...,z=25)**

Tip – You may find the key-value object mapping in the file useful.

Note: After editing the JS file, make sure to click **File -> Save in VSCode,** and **refresh the Chrome page**, for the edits to manifest in the console!

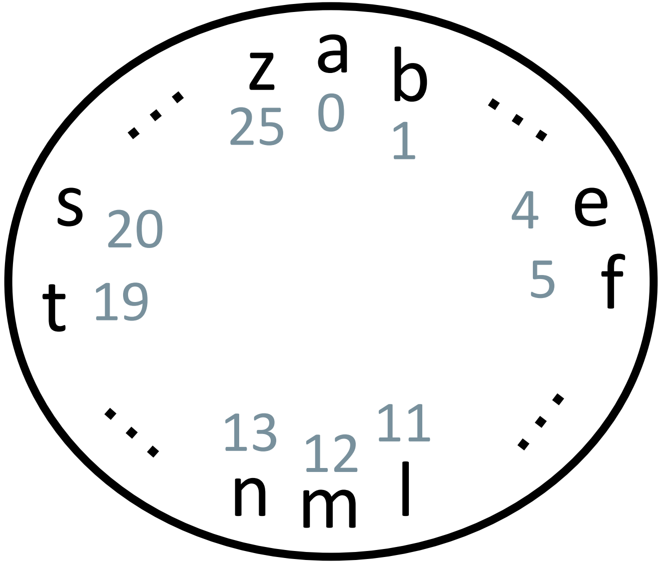


Checkpoint #2 assignment.js

Implement the function indexToLetter()

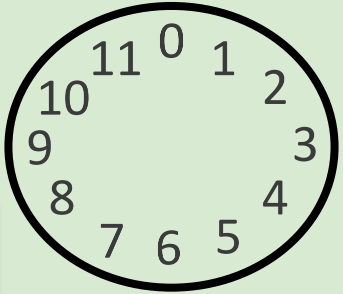
<u>Input:</u> Non-negative index of a letter, can be 26 and greater
<u>Output:</u> Corresponding lowercase letter; numbers above 25 wrap around
i.e. 0=a,1=b,..,25=z,26=a,27=b,...

Tip – The array alphabet=
[‘a’,’b’,..,’z’] may come in handy.



For handling the wrap-around, consider the remainder operator (%) and this clock example. If it’s 8:00 right now, then 7 hours later, it’ll be 3:00, as times *wrap around* the 12-hour clock. This can be computed with the following JavaScript:

```
(8 + 7) % 12 // 8+7 => 15 o'clock 🤔
  // clock only has 12 hours (0-11), so 15%12 => 3:00
```



Checkpoint #3

Implement the function **shiftLetter()**

Inputs: **original** (letter to shift), **shift** (length to transpose letters by)
Output: shifted letter

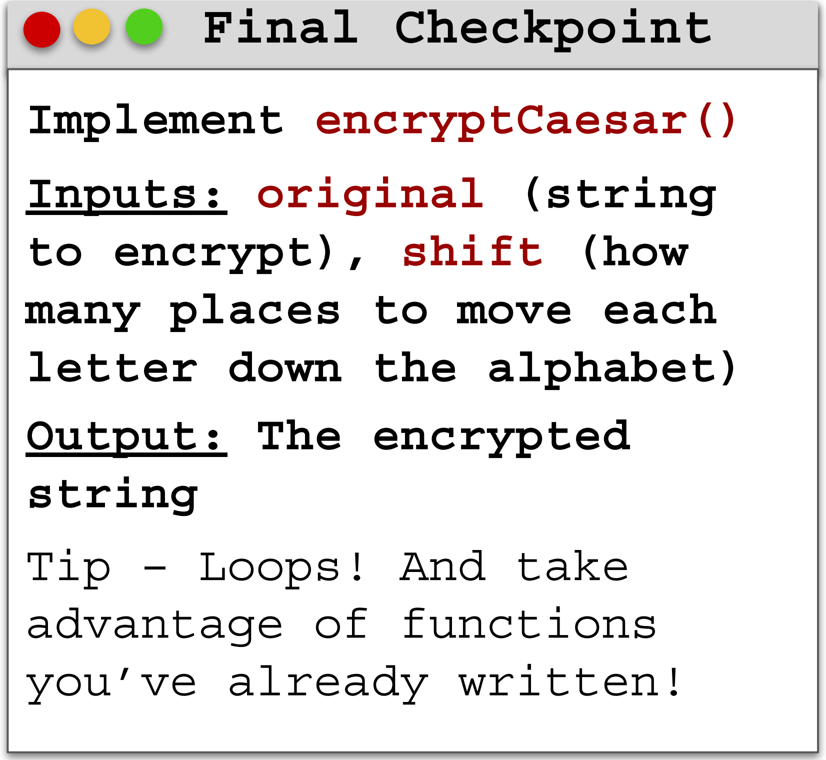
Tip - Use letterToIndex() and indexToLetter()!

Example Functionality

JS Console

```
🎲 shiftLetter('a',1)
'b'

🎲 shiftLetter('a',4)
'e'

🎲 shiftLetter('z',3)
'c'
```

🔐 Full Encryption Pipeline



Final Checkpoint

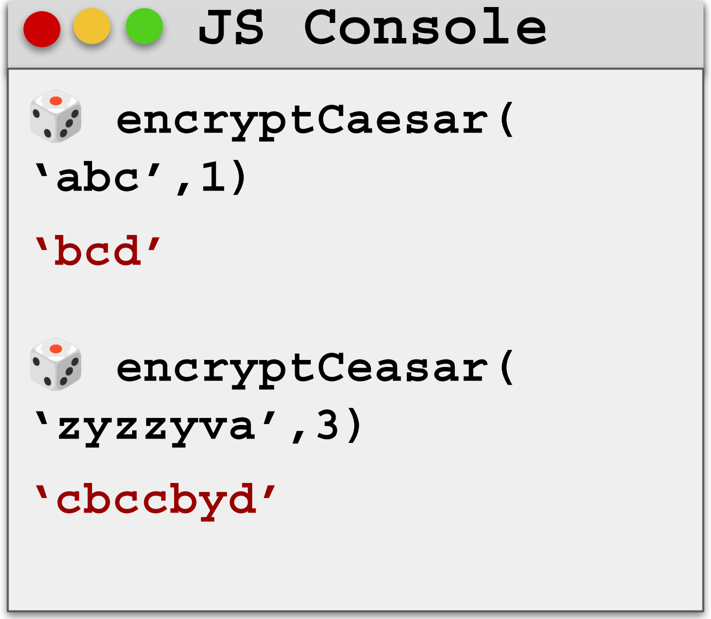
Implement **encryptCaesar()**

Inputs: **original** **(string to encrypt),** **shift** **(how many places to move each letter down the alphabet)**

Output: **The encrypted string**

Tip - Loops! And take advantage of functions you've already written!

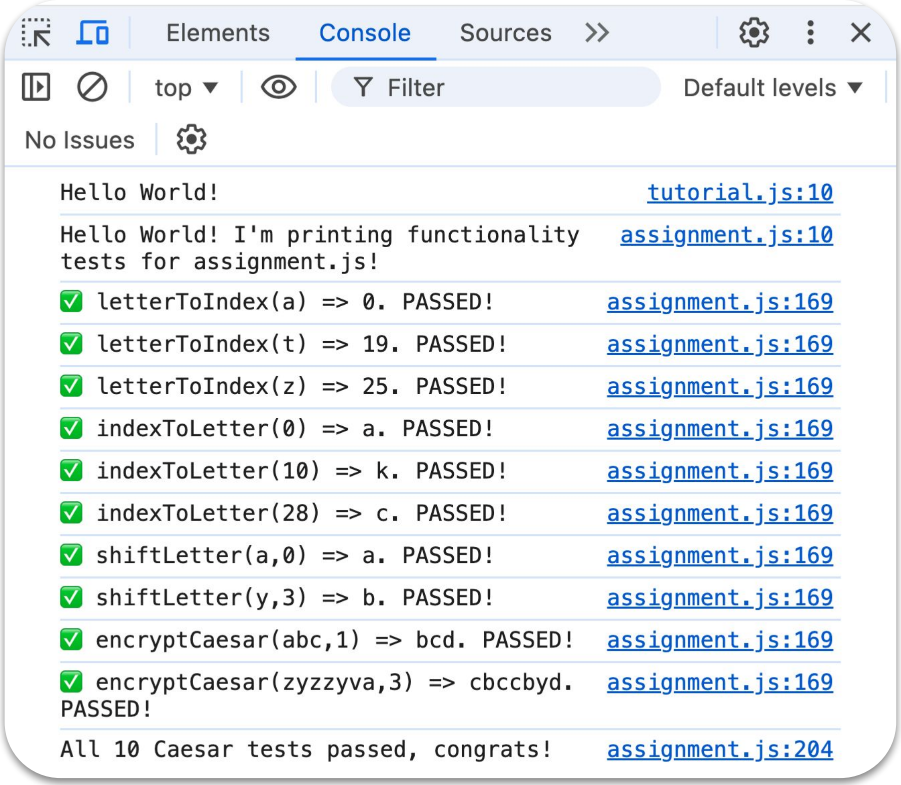
Example Functionality



JS Console

```
🎲 encryptCaesar(
'abc',1)
'bcd'

🎲 encryptCeasar(
'zyzzyva',3)
'cbccbyd'
```

Sanity Testing



```
Hello World!                                          tutorial.js:10
Hello World! I'm printing functionality             assignment.js:10
tests for assignment.js!
✅ letterToIndex(a) => 0. PASSED!                   assignment.js:169
✅ letterToIndex(t) => 19. PASSED!                  assignment.js:169
✅ letterToIndex(z) => 25. PASSED!                  assignment.js:169
✅ indexToLetter(0) => a. PASSED!                   assignment.js:169
✅ indexToLetter(10) => k. PASSED!                  assignment.js:169
✅ indexToLetter(28) => c. PASSED!                  assignment.js:169
✅ shiftLetter(a,0) => a. PASSED!                   assignment.js:169
✅ shiftLetter(y,3) => b. PASSED!                   assignment.js:169
✅ encryptCaesar(abc,1) => bcd. PASSED!             assignment.js:169
✅ encryptCaesar(zyzzyva,3) => cbccbyd.             assignment.js:169
PASSED!
All 10 Caesar tests passed, congrats!               assignment.js:204
```

All tests should pass after **encryptCaesar()** is successfully implemented!

Solution code available on website right after class :)

To test Vigenère (optional!), go to the bottom of the **assignment.js** file, and uncomment the line that reads as:

```
// testVigenere() // uncomment to
test Vigenere cipher (optional)
```

Check-Off Form!

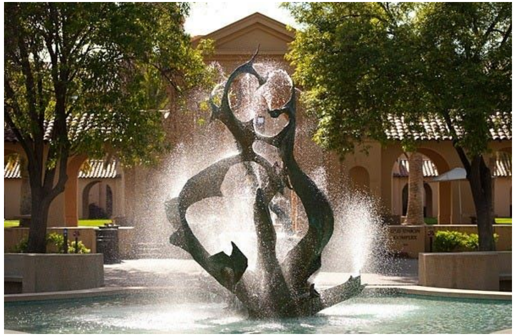
To get attendance credit each class, you'll fill out a **brief check-off form** (~2 – 5 min to complete).

For today, click the "Check-Off Form" link in the Week 1 section of **cs106s.stanford.edu!**

https://tinyurl.com/cs106s-spr25-w1-checkoff (case sensitive!)

Looking Forward to this Spring 🌸

Teaching this 1-unit wonder has been a truly wonderful joy and privilege for me, ~~because of all the free boba over the years~~; thank you for being here to learn with us, and I 👏 hope 👏 this 👏 will 👏 be 👏 fun 👏 for you!!



Have an awesome first week of classes! :)