

WELCOME TO CSCI E-33A WEB PROGRAMMING W/ PYTHON AND JAVASCRIPT

TF FOR THIS SECTION: GLENN LANGDON

LANGDON@cs50.HARVARD.EDU



ABOUT ME

- CONDUCTOR
 - BROADWAY
 - BALLET
 - OPERA
- PIANIST
- PROGRAMMER
- FARMER
 - FAMILY FARM SINCE 1840





THOUGH A
MUSICIAN BY
TRADE, I'VE ALWAYS
BEEN INTERESTED IN
COMPUTERS AND
TECHNOLOGY

MY FIRST COMPUTER: 1987 ZENITH Z-286
(512K RAM)



FORMAL COMPUTER SCIENCE EDUCATION

- STANFORD UNIVERSITY, DISTINGUISHED CAREERS INSTITUTE:
FOUNDATIONAL PROGRAMMING COURSES
- HARVARD ALM IN DGMD: CERTIFICATES IN FRONT &
BACKEND PROGRAMMING
- ASSOCIATED WITH CSCI E-33A SINCE 2020

ENOUGH OF ME! LET'S TALK ABOUT YOU AND THIS COURSE



THE PLAN FOR TONIGHT



Overview

Overview of course requirements



Review

Review of text editor environments



Review

Review the assignment



Build

Build a simple responsive web page with CSS

- Navbar

COURSE OVERVIEW & HOW THE SEMESTER WILL UNFOLD

Description

Prerequisites

Expectations

Grades

Lectures

Quizzes

Sections/OH

Projects

Lateness/Extension Policy

Academic honesty



HOW MY SECTIONS ARE STRUCTURED

- DON'T DO A VERBATIM REVIEW OF WHAT BRIAN HAS ALREADY PRESENTED
- ASSUME THAT EVERYONE HAS VIEWED BRIAN'S PRESENTATION BEFORE THE SECTION
- BEGIN EACH SECTION READY TO ANSWER ANY QUESTIONS PRESENTED IN THE LECTURE VIDEOS OR ABOUT THE MATERIAL
- GO A BIT DEEPER INTO RELATED TOPICS WHICH ARE APPLICABLE TO THE FULFILLMENT OF THE ASSIGNMENTS
- ALWAYS READY TO EXPLAIN OR ADDRESS ANY CONCERNSTHAT STUDENTS HAVE IN FULFILLING THE ASSIGNMENT DURING THE SECTIONS
- ASK QUESTIONS AT ANY TIME!

A WORD ON GRADING FOR THE 6 PROJECTS

Correctness: We grade only on the fulfillment of the specifications (nothing more, nothing less)

Design: Grading is additive, not subtractive (more on that next week)

A WORD ON QUIZ GRADING

For discussion questions, a score of 4 is reserved for only answers that are “exceptional answers that demonstrate a full understanding of the material in a manner that exceeds the course’s expectations”.

A score of 3 should be considered a benchmark for success.

TIPS FOR A SUCCESSFUL SEMESTER



START ASSIGNMENTS
EARLY!



FOLLOW THE
SPECIFICATIONS IN
THE ASSIGNMENTS
(THIS IS NOT A DESIGN
CLASS)



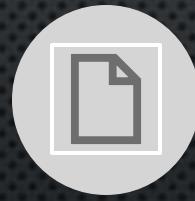
NEW LECTURES ARE
RELEASED ON
MONDAY APPROX.
EVERY 2 WEEKS
(EVERY WEEK DURING
THE SUMMER)



CONSULT ED TO POST
QUESTIONS



THERE ARE NO
TEXTBOOKS, YOU WILL
BE GIVEN
EVERYTHING YOU
NEED TO COMPLETE
THE PROJECTS



ATTEND SECTIONS
AND OFFICE HOURS
(AS MANY AS YOU
LIKE)



ASK QUESTIONS



DID I SAY START THE
ASSIGNMENTS EARLY?

TIPS ON USING CS50.AI



BE SPECIFIC



USE FOLLOW UP
QUESTIONS TO CLARIFY



THE ELEPHANT IN THE ROOM

- CHATGPT, OPENAI, ETC.
- PLEASE NOTE THE CLASS GUIDELINES FOR OUTSIDE HELP AND WHAT IS CONSIDERED "ACCEPTABLE" AND "NOT ACCEPTABLE".



HOW DO I GET HELP WITH THE PROJECTS?

- POST ON ED
 - IF POSTING CODE, MAKE THE POST PRIVATE, OTHERWISE KEEP THE POST PUBLIC
- USE CS50 AI
- COME TO OFFICE HOURS
- SEND ME AN EMAIL
 - IF NECESSARY, I'LL SET UP A ZOOM APPT

WHAT TEXT EDITOR SHOULD I
USE FOR THIS CLASS?

SETTING UP YOUR TEXT EDITOR

You can use any editor you wish

VSCODE is free with deep selection of packages and personalization for users

Download at

<https://code.visualstudio.com>

Add VSCODE executable to \$PATH var:
Command + Shift + P (Mac)

Open VSCODE from any directory by typing code .

You're set!

ASSIGNMENT 0 (SEARCH)

Read the project Background (self-explanatory)

Download the project

Fulfill the specifications

When we give you a hint, it's a hint that will be useful in the fulfillment of the specs...

Zip your implementation in the folder in which your project is located (no exotic compression algorithms, please. Just a regular .zip file)

Upload to GradeScope (don't wait until the last minute)

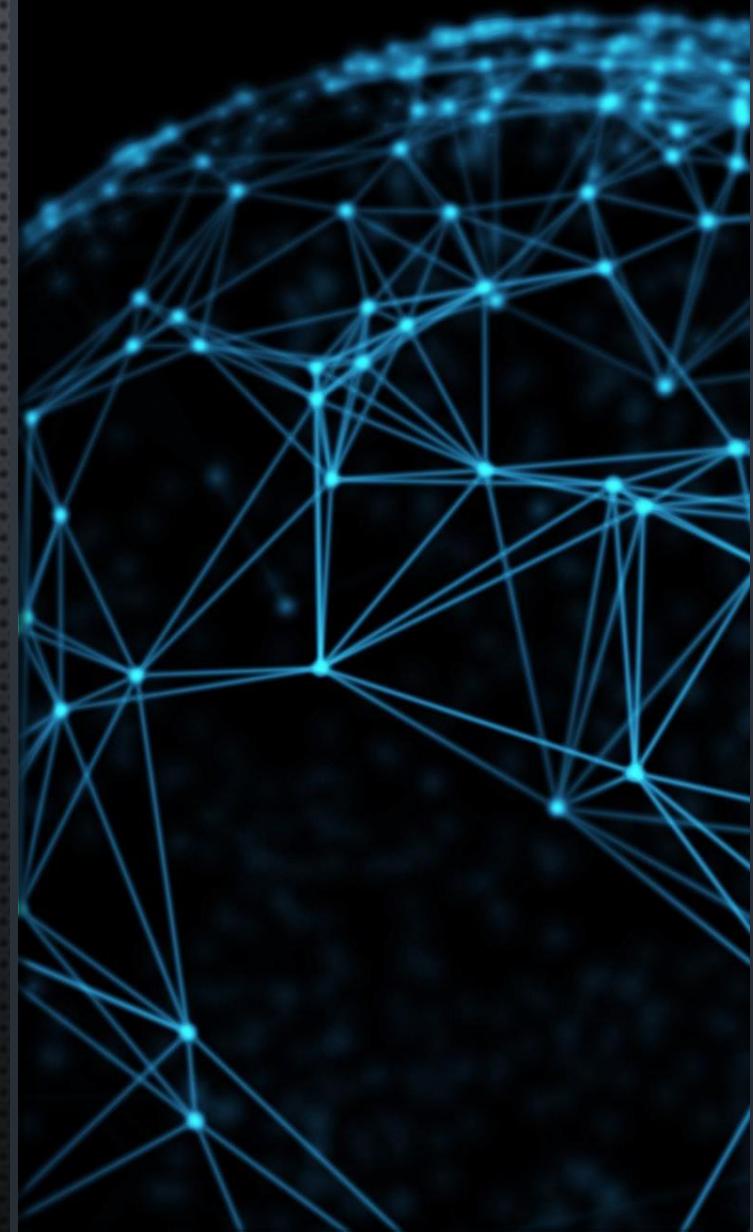


ANY QUESTIONS BEFORE WE GET INTO
THE MATERIAL FOR THIS WEEK?

PARTS OF A WEB ADDRESS

HTTP://WWW.DJANGO-IS-FUN.COM:8000/ABOUT

- PROTOCOL
- DOMAIN (HOSTNAME OR WEBSITE NAME)
 - BEST TO READ RIGHT TO LEFT
 - DOMAIN TYPE (TLD)
 - HOST NAME IS A SUBDOMAIN OF .COM (A PARTICULAR SERVER)
 - WWW. IS THE WORLD WIDE WEB SUBDOMAIN OF DJANGO-IS-FUN.COM
- PORT
- RESOURCE (PATH)



WHAT IS HTTP?

HYPertext Transfer
Protocol is the
Protocol used for
communication on
the World Wide
Web.

WHAT THIS ADDRESS MEANS

<http://www.djangoproject.com:8000/about>

Make an *HTTP* request to the server at subdomain www of *djangoisfun.com* on port 8000 and ask for the /about page

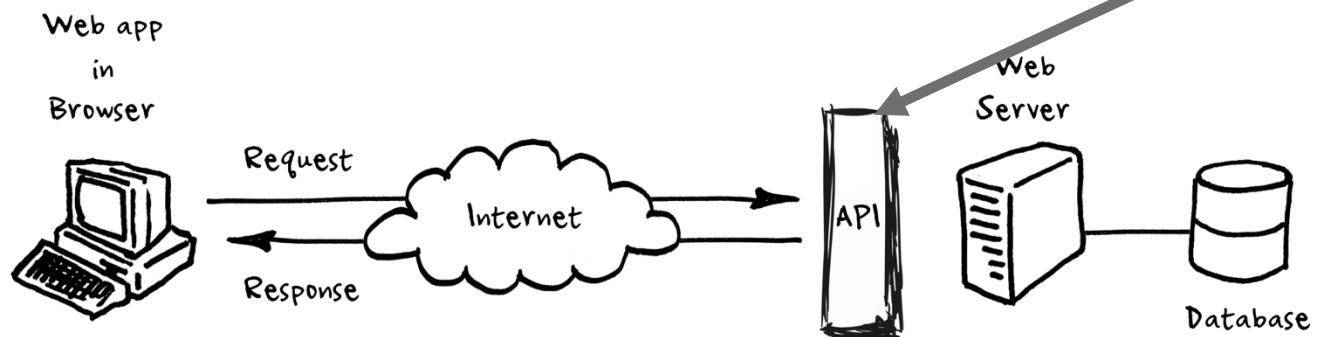
HTTP REQUEST & RESPONSE

- AN UNDERSTANDING OF THE LIFECYCLE OF THE HTTP REQUEST/RESPONSE IS IMPORTANT FOR THIS COURSE
- THIS IS THE BASIS FOR ALL WEB APPLICATION COMMUNICATION
- HTTP REQUEST: THE CALL THE BROWSER/CLIENT MAKES TO A WEB SERVER (GOOGLE SEARCH FOR EXAMPLE)
- HTTP RESPONSE: THE RESPONSE A SERVER SENDS BACK TO THE BROWSER/CLIENT AS A RESULT OF THE REQUEST

```
mirror_mod = modifier_ob
# set mirror object to mirror
mirror_mod.mirror_object = ob
if operation == "MIRROR_X":
    mirror_mod.use_x = True
    mirror_mod.use_y = False
    mirror_mod.use_z = False
elif operation == "MIRROR_Y":
    mirror_mod.use_x = False
    mirror_mod.use_y = True
    mirror_mod.use_z = False
elif operation == "MIRROR_Z":
    mirror_mod.use_x = False
    mirror_mod.use_y = False
    mirror_mod.use_z = True

# selection at the end - add
modifier_ob.select= 1
modifier.select=1
context.scene.objects.active = modifier
("Selected" + str(modifier))
modifier.select = 0
bpy.context.selected_objects.append(modifier)
data.objects[one.name].select = 1
print("please select exactly one object")
print("operator selected")

- OPERATOR CLASSES -
# types.Operator:
# X mirror to the selected object.mirror_mirror_x"
# for X"
# context):
# context.active_object is not None
```



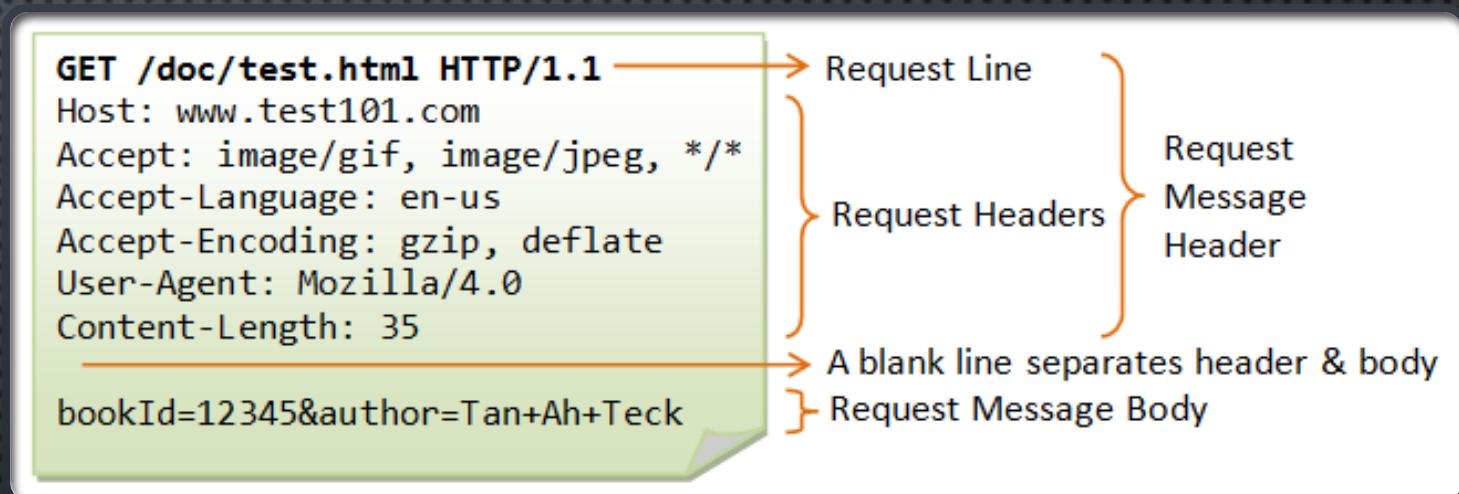
API- Application programming interface is a software interface which allows two or more computers to communicate with each other.

Computers talking to computers as opposed to human computer interaction.

HTTP REQUEST

A MESSAGE SENT BY A BROWSER (CLIENT) TO A SERVER

- REQUEST LINE
 - METHOD
 - GET, POST, PUT DELETE
 - URI OF THE RESOURCE (UNIFORM RESOURCE IDENTIFIER)
 - HTTP VERSION
- REQUEST HEADERS
 - HOST, USER-AGENT, CONTENT-LENGTH, ETC.
- BLANK LINE
- REQUEST BODY
 - CONTAINS FORM DATA BEING SENT TO SERVER



HTTP RESPONSE

WHAT THE SERVER SENDS BACK

- STATUS LINE
 - HTTP VERSION
 - STATUS CODE
 - REASON PHRASE
- RESPONSE HEADERS
(INFORMATION ABOUT THE RESPONSE)
 - DATE
 - CONTENT-TYPE
- BLANK LINE
- RESPONSE BODY

HTTP response message

status line
(protocol
status code
status phrase)

header lines

data, e.g.,
requested
HTML file

```
HTTP/1.1 200 OK
Connection close
Date: Thu, 06 Aug 1998 12:00:15 GMT
Server: Apache/1.3.0 (Unix)
Last-Modified: Mon, 22 Jun 1998 .....
Content-Length: 6821
Content-Type: text/html

data data data data data ...
```

HERE'S A REQUEST-RESPONSE IN ACTION

CURL (SHORT FOR CLIENT URL)

- GET
 - CURL -V HTTPS://HTTPBIN.ORG/GET
- POST
 - CURL -V -X POST HTTPS://HTTPBIN.ORG/POST -D "NAME=JOHN&AGE=30"

REST CLIENT FOR VS CODE

- [HTTPS://GITHUB.COM/HUACHAO/VSCODE-RESTCLIENT](https://github.com/huachao/vscode-restclient)

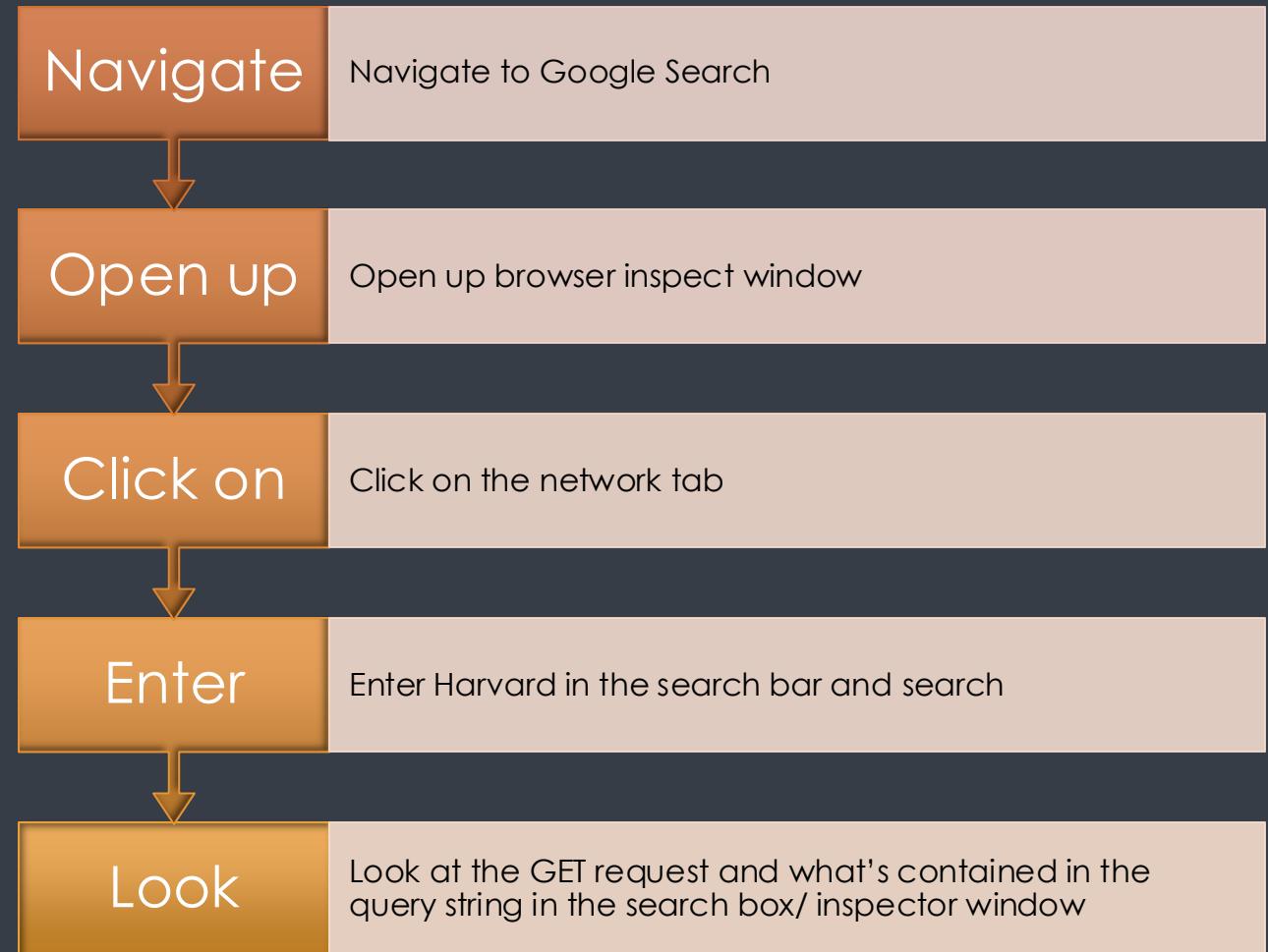


REQUEST/RESPONSE QUESTION

- AFTER SEEING GET AND POST REQUESTS IN ACTION, WHY WOULD A POST REQUEST BE CONSIDERED MORE SECURE THAN A GET REQUEST?



HOW CAN I SEE THESE REQUESTS AND RESPONSES IN MY BROWSER?





A GOOGLE SEARCH
PAGE IS NOTHING
BUT A FORM IN
WHICH YOU SUBMIT
INFORMATION TO A
SERVER

FORMS

HOW DATA IS SUBMITTED TO A SERVER

<form>

- Container for different types of input elements

<input>

- Types: text, radio, checkbox, submit, button

<label>

- Defines a label and is useful for screen-readers

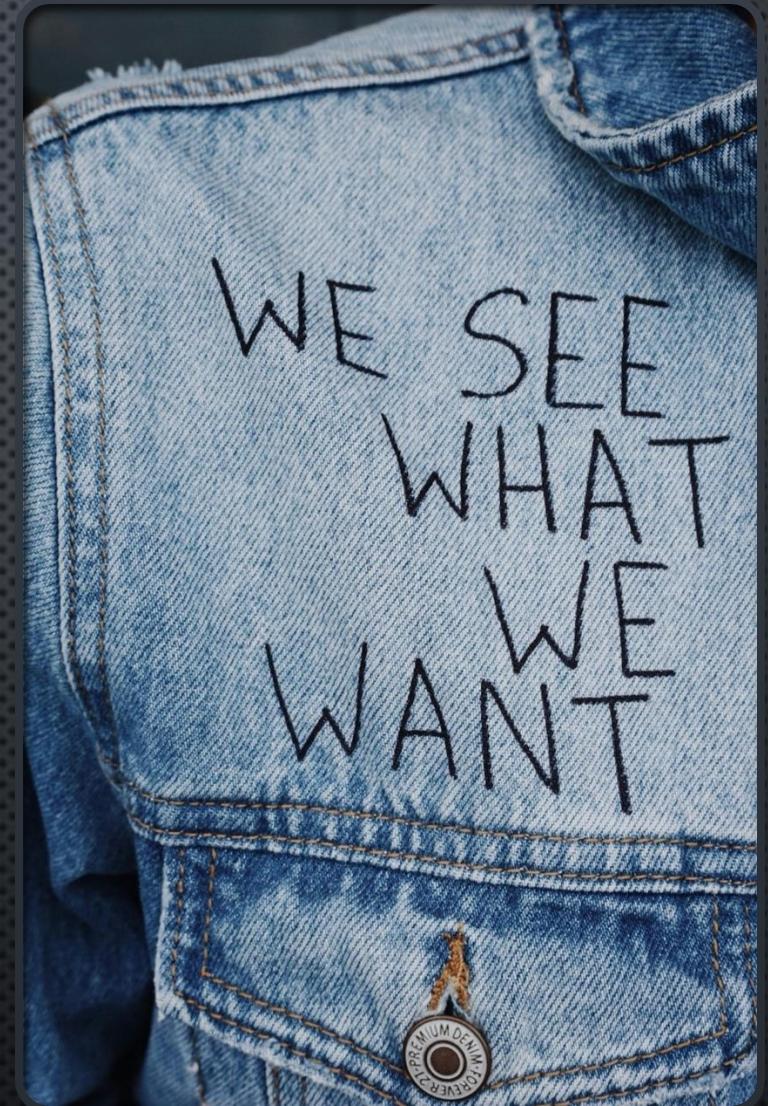
Name attribute for
<input>

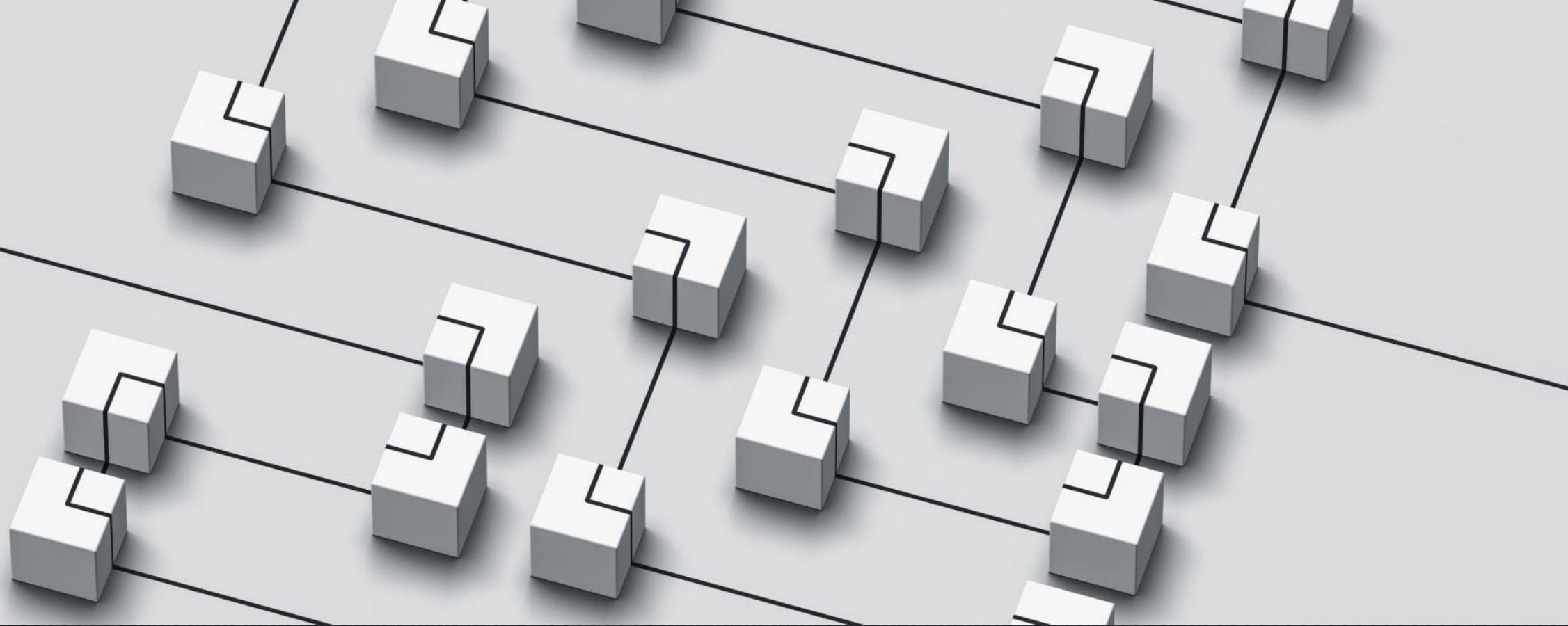
- Each input field must have a name attribute, or the value of the input field will not be sent

LET'S LOOK AT A FORM IN
ACTION

STYLING YOUR WEB PAGES

- ALTHOUGH THIS ISN'T A STYLING COURSE, YOU SHOULD BE COMFORTABLE ENOUGH WITH A CSS LIBRARY OR MODULE WHICH WILL ALLOW YOU TO IMPLEMENT YOUR ASSIGNMENT WITHOUT GETTING CAUGHT UP WITH STYLING CHALLENGES
- *BOOTSTRAP* IS FINE IF YOU'RE FAMILIAR WITH IT. THERE'S A WEALTH OF ONLINE DOCUMENTATION.
- *FLEXBOX* AND *GRID* ARE 2 OTHER GREAT OPTIONS. I'LL LOOK AT FLEXBOX NOW.





CSS FLEXIBLE BOX LAYOUT MODEL (FLEXBOX)

CSS FLEXBOX

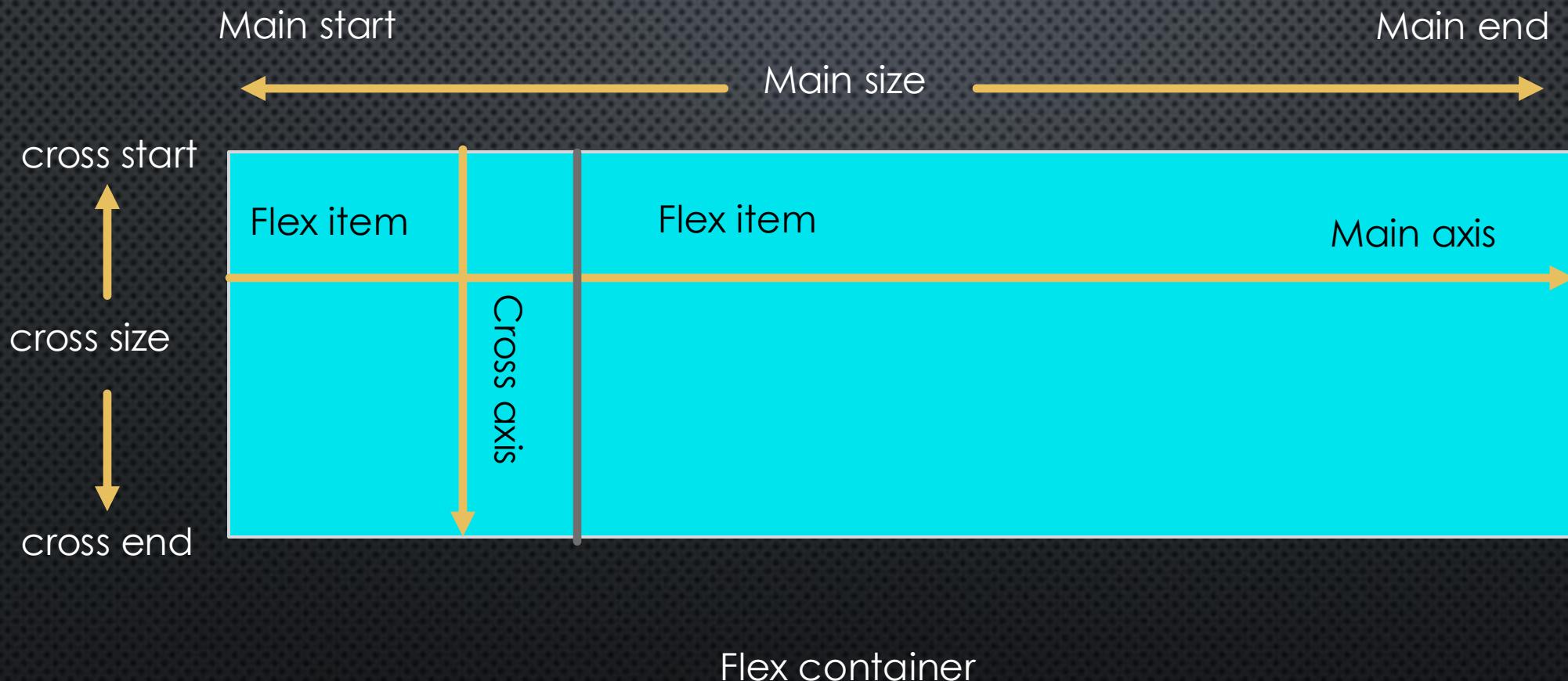
Flexbox element: All child (direct) elements line up on one axis

- Main axis
- Cross axis

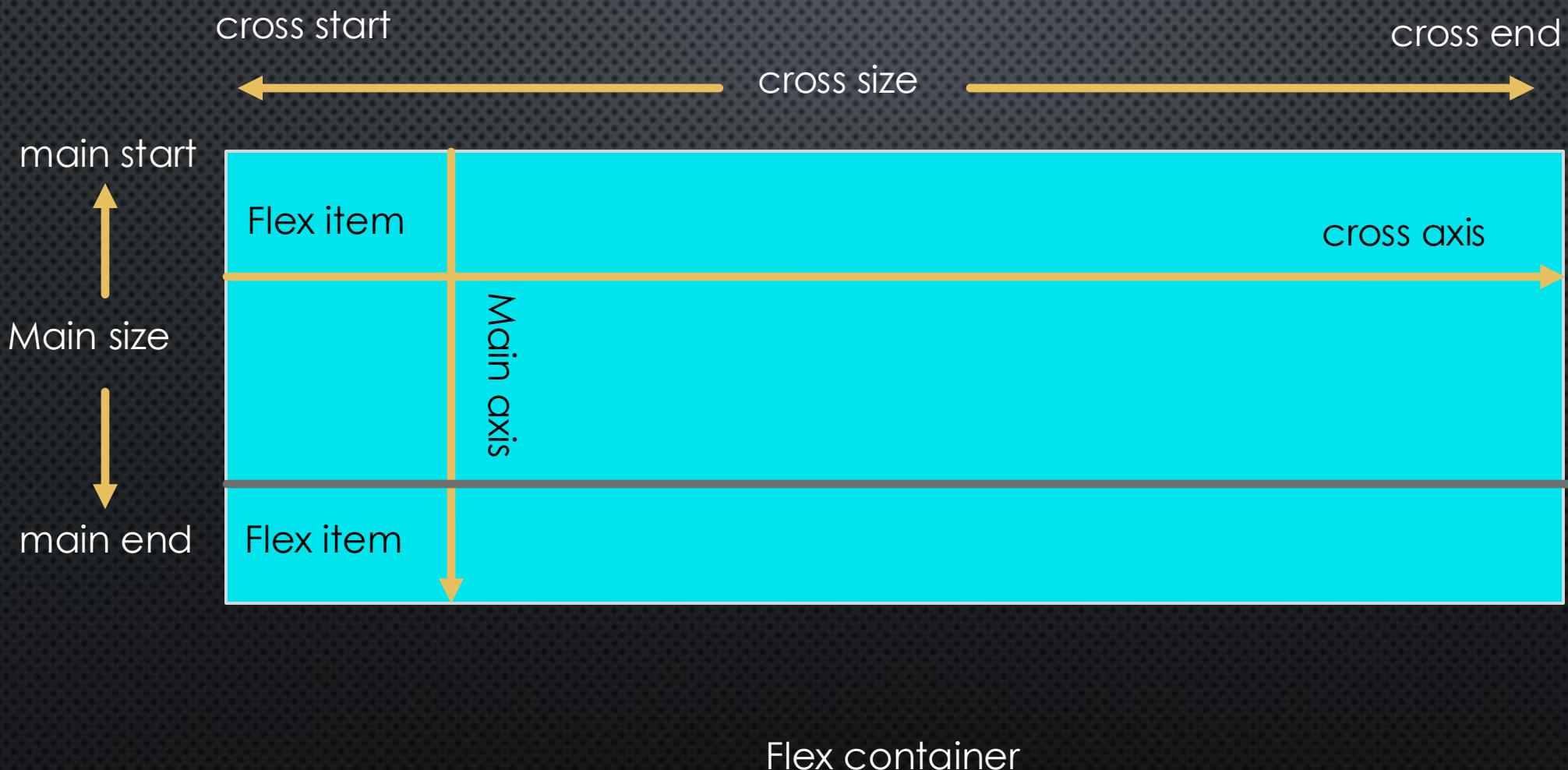
Flow within the container: flex-direction

- Row (default)
- Row-reverse
- Column
- Column-reverse

FLEX DIRECTION: ROW



FLEX DIRECTION: COLUMN





NOW LET'S LOOK AT WHAT
FLEXBOX CAN DO

SO, WHAT'S THE POINT OF THIS?

- YOUR ASSIGNMENT FOR THIS WEEK REQUIRES YOU TO STYLE A GOOGLE CLONE
- ALTHOUGH, IT DOESN'T NEED TO BE AN EXACT CLONE, MENU ITEMS AND FORM ALIGNMENT ARE IMPORTANT
- FLEXBOX IS A GREAT WAY TO APPROACH IT WITHOUT USING EXTERNAL LIBRARIES

In section navbar exercise

Your choice2 Your choice1 Your choice0

LET'S BUILD A SIMPLE WEB PAGE
USING FLEXBOX



THAT'S IT FOR THIS EVENING!

HAVE A GREAT WEEK!