

CS 498RK

SUMMER 2017

# CS498RK: Art of Web Programming

CS 498RK

SUMMER 2017

# COURSE OVERVIEW

# WEB SKILL SETS

Design

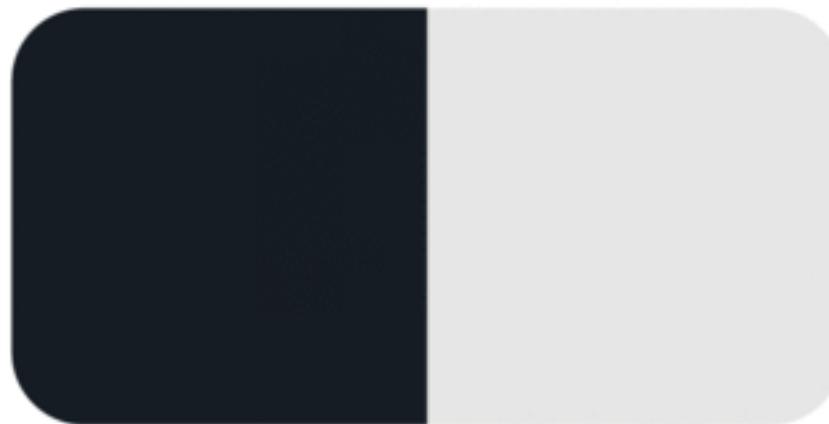
Front end

Back end

~1989: Unix-based web browsers



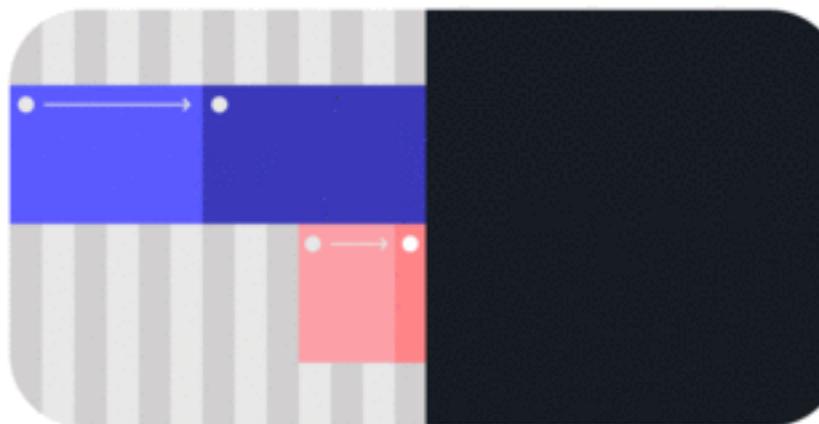
~1995: First graphical Web browsers



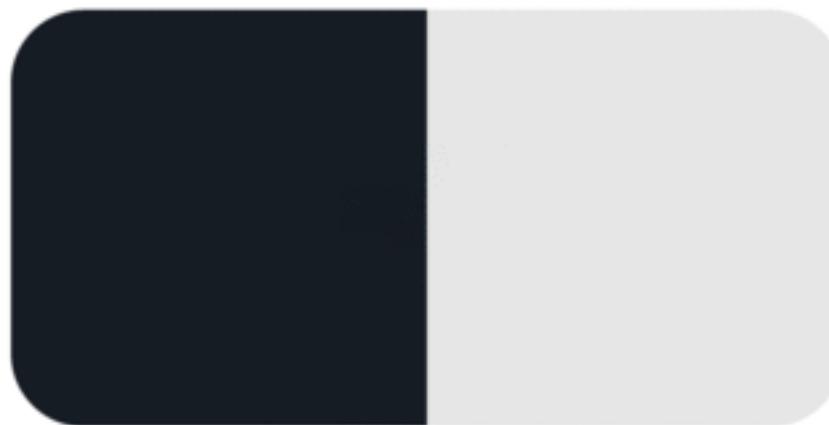
~1995: Javascript & Dynamic Content



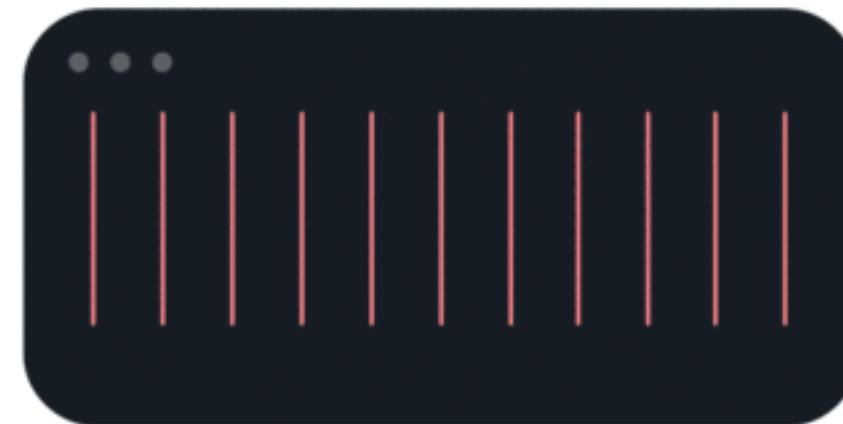
~1996: Flash animations



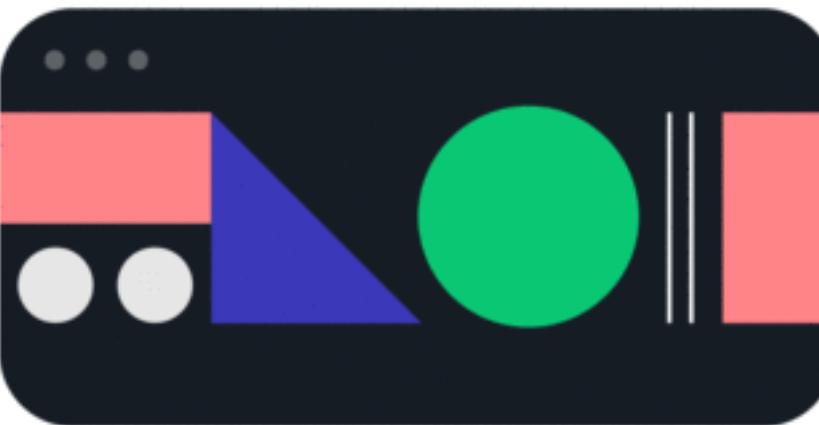
~1998: CSS came on the scene



~2007: Grid systems

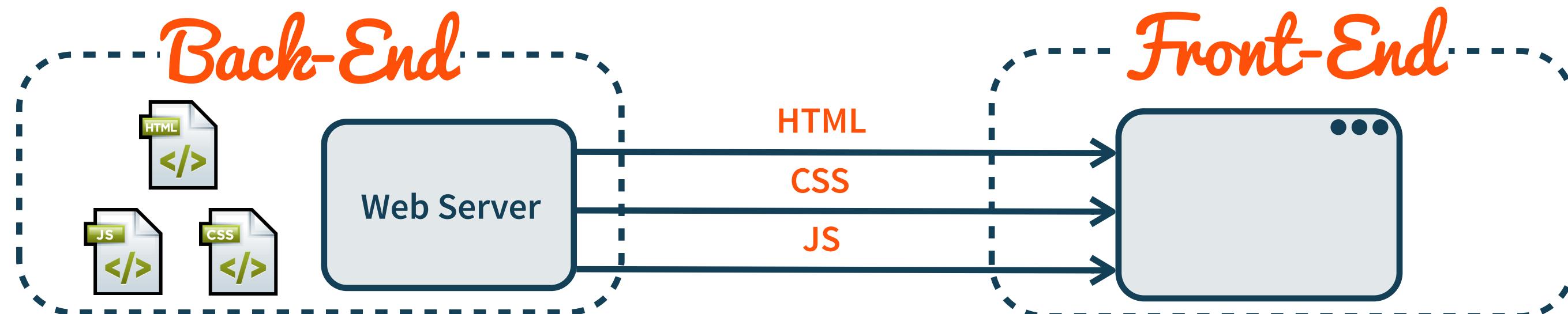
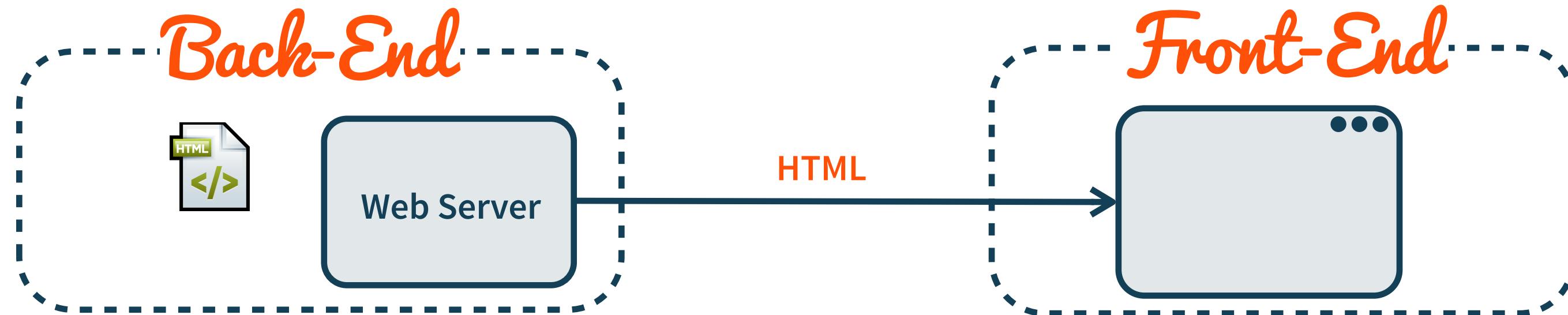


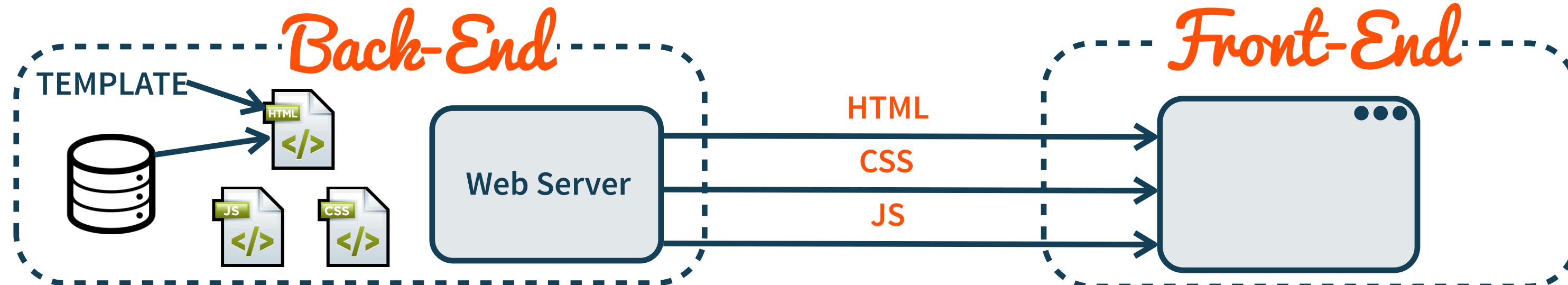
# ~2010: Responsive design



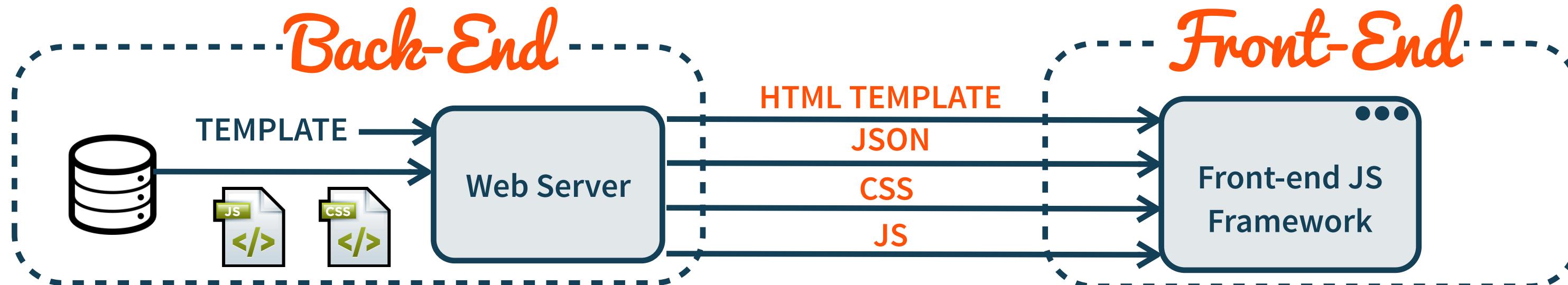
# ~2013: Flat design







Back-End Templating



Front-End Templating

TEMPLATE

*Server*

SQL vs NoSQL

Mongo

Event-Driven  
Programming  
Node.js, Express

REST

HTTP

AJAX

RESTful APIs  
Data Binding  
Sockets

*Client*

HTML

CSS

CSS preprocessors

Javascript

jQuery  
Angular

# HTML

Most web pages are written in HTML

Content is embedded in a set of nested HTML tags

Layout engine parses HTML into a Document Object Model

Web browsers use DOM to render pages

```
<!DOCTYPE html>
<html>
  <head>
    <title>Photo Gallery</title>
  </head>

  <body>
    <div class="photo">
      <h3>My first photo</h3>
      
    </div>
    ...
  </body>
</html>
```

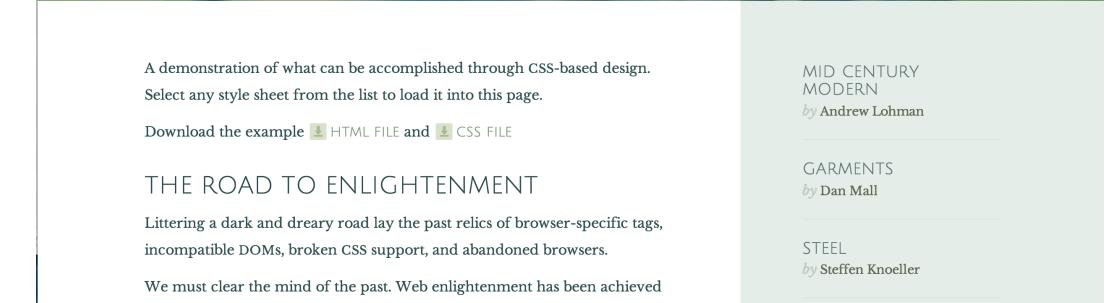
# CSS

Language for specifying presentation

Selectors map styles to markup

Describe how to render

Separation of content from presentation



[csszengarden.com](http://csszengarden.com)

# JAVASCRIPT

front-end interactions

dynamic content

server-side programming (node.js)

object-oriented, imperative, functional

# JAVASCRIPT IS WEIRD

```
[10, 20, 9, 8, 30].sort()
```

# JAVASCRIPT IS WEIRD

[10, 20, 30, 8, 9]

# CLIENT SERVER COMMUNICATION

HTTP: request-response protocol

AJAX: send and receive data without  
reloading page

JSON: data exchange format

Model-View-Controller

# APIs AND SERVER LOGIC

HTTP requests: GET, POST, DELETE

Designing a RESTful API

Node.js and Express

Web Sockets

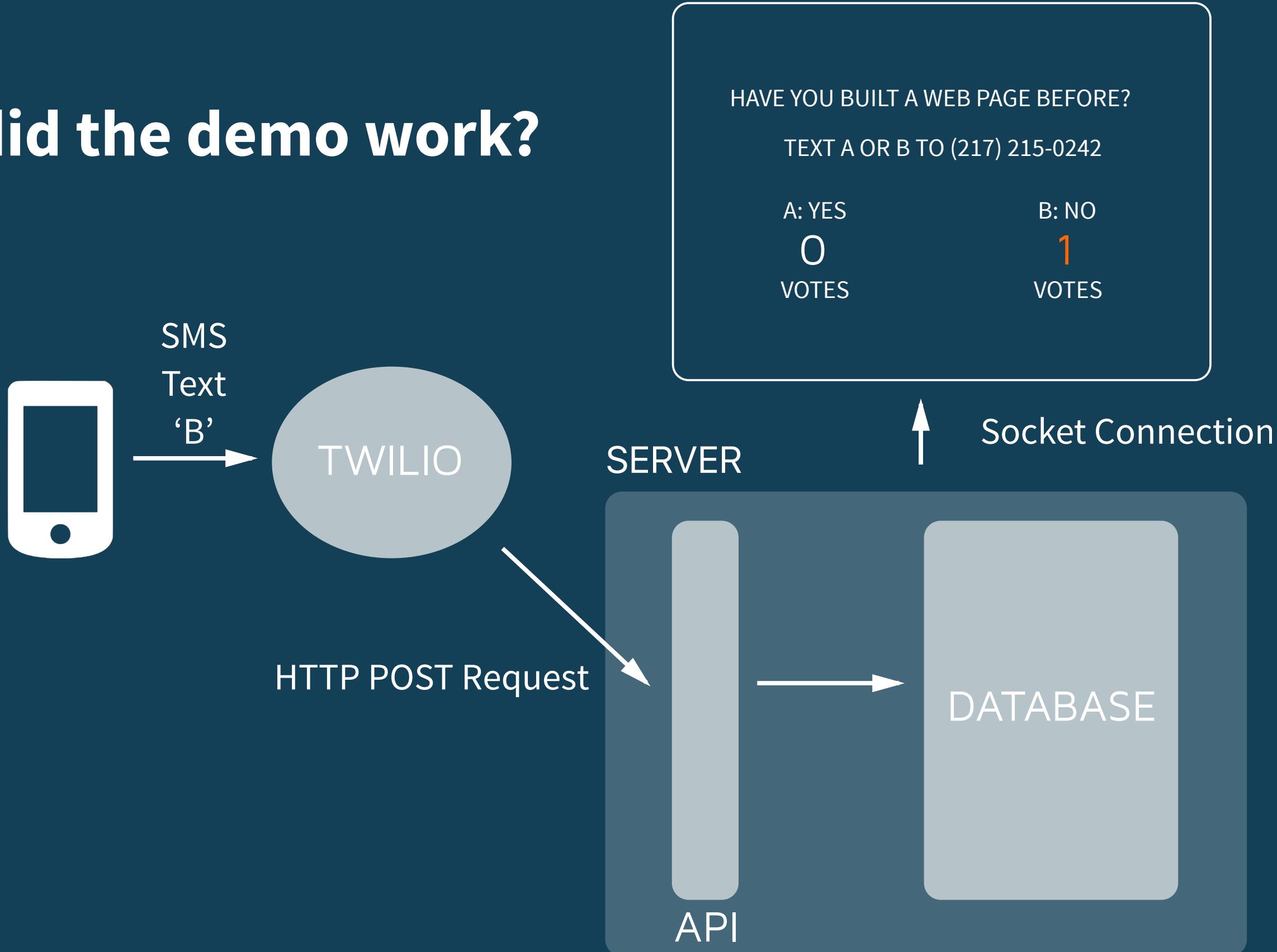
# DATABASES

SQL (MySQL)

NoSQL (MongoDB)

Graph (Neo4j)

# How did the demo work?

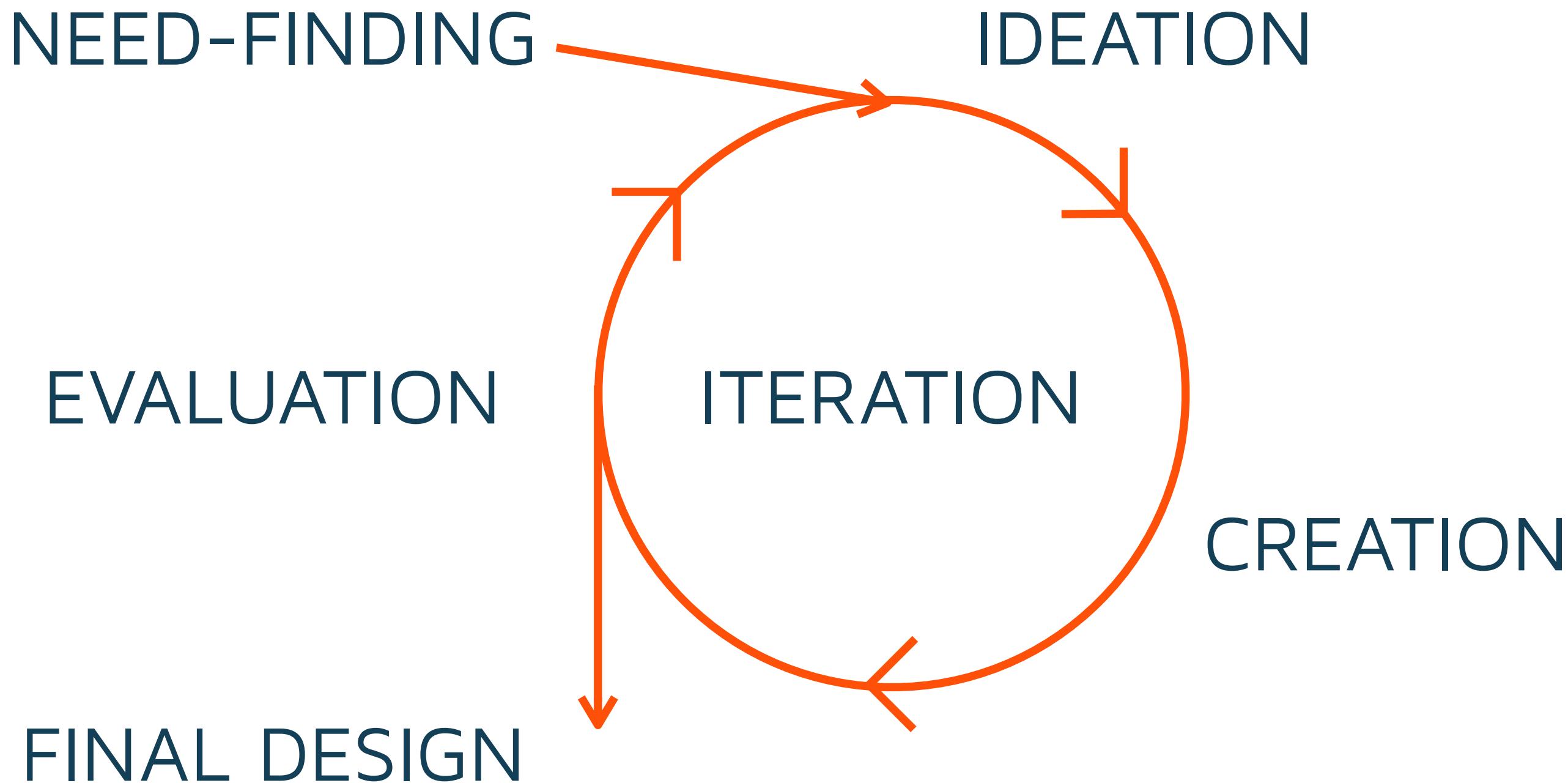


# WILL THIS COURSE BE OUTDATED NEXT YEAR?

Focus on concepts not just specific technologies

Understand how trends arose and have changed

# DESIGN EMPHASIS



ADMINISTRIVIA

# COURSE WEBPAGE

<https://uiuc-web-programming.github.io/su17/>

# COURSE STAFF



Biplab



Krishna



Goutham

# OFFICE HOURS

Biplab: Thursday 5-6pm, SC 0207

Goutham & Krishna: will be here after class  
on Mondays

Avoid emailing us directly – use **Piazza**

# LECTURES & LABS

Lectures cover theory and concepts

Labs walk through concrete code examples

Bring your laptops to labs and follow along

In-class warm-up problems count toward participation

# ASSIGNMENTS

5 MPs (60% of grade)

Learn the entire Web stack

Late assignments receive **no credit**

**Three** 24-hour late days

**60% OF GRADE**

# EXAMS

In-class midterm on **July 5th**

Final exam on **Aug 5th**

Alternative arrangements must be  
made **two-weeks** prior to exam

**40% OF GRADE**

# ACADEMIC INTEGRITY

Consult external resources to complete assignments

Clearly cite any contributing source

Failure to cite any contributing source will be  
considered **cheating**

Verbatim duplication of any source will always be  
considered **plagiarism**

# PROFESSIONAL DEVELOPMENT

Guest Lectures:

1. PrairieLearn

2. TBD

# QUESTIONS

# NEXT STEPS

MP 0 due in a week

You'll need to:

Learn HTML + CSS + Basic Javascript

Setup dev environment

Use submission system

# NEXT CLASS: HTML + CSS + Setup

<https://uiuc-web-programming.github.io/su17/>