Web Advanced:

Javascript

An introduction to the world of JavaScript .

We will learn JavaScript the right way and write code using modern coding patterns, integrate through REST APIs and get introduced to popular frameworks.”

SPRING 2024

HELLO.

jaink@newschool.edu

https://canvas.newschool.edu/courses/1755416

https://replit.com/@jaink/pgte-5505-s24

https://NewSchool.zoom.us/j/99213043874?pwd=QVBwVk5ldXZ0a2lVSG56N25Nczl2QT09

https://github.com/kujain/S24-5505_Javascript

INTRODUCTIONS

Why Learn Coding?

PRO

- Better understanding of what goes into creating computer web/applications.
- Better understanding of limits of design ideas.
- Create more efficient and empathic design.

CON

- Learning curve/mental block.
- Unnecessary specialization.
- Apps/AI tools available to help auto-generate code.
- Coding standards and patterns are constantly evolving - constantly playing catch-up.

QUIZ

OPEN SOURCE

What does a Program look like?

➔ Let’s look at Code written in different languages...

MACHINE LANGUAGE

```
01001000 01100101 01101100
01101100 01101111 00100000
01010111 01101111 01110010
01101100 01100100
```

C++

```cpp
#include <iostream>
using namespace std;

int main()  {
    float length, width, area;
    cout << "Enter The Length: ";
    cin >> length;
    cout << "Enter The Width: ";
    cin >> width;
    area = length*width;
    cout <<"Answer is : "<< area << endl;
    return 0;
}
```

JAVA

```
public static int fctl(int n)
{
      int result = 1;
      for(int i = 2; i <= n; i++)
         result *= i;
      return result;
}

factl(10)
```

PHP

```php
<?php

class Vegetable {
   var $veg;
   var $color;

   function __construct($veg, $color="green") {
       $this->veg = $veg;
       $this->color = $color;
   }

   function get_name() {
       return $this->veg;
   }

   function what_color() {
       return $this->color;
   }
} // end of class Vegetable

$Veg = new Vegetable( "tomato", "red");
echo $Veg->get_name() . " is " . $Veg->what_color();

?>
```

P5

```
function setup() {
  let d = 70;
  let p1 = d;
  let p2 = p1 + d;
  let p3 = p2 + d;
  let p4 = p3 + d;

  createCanvas(720, 400);
  background(0);
  noSmooth();

  translate(140, 0);

  // Draw
  stroke(150);
  line(p3, p3, p2, p3);
  line(p2, p3, p2, p2);
  line(p2, p2, p3, p2);
  line(p3, p2, p3, p3);
}
```

R

```
num = as.integer(readline(prompt="Enter a number: "))

flag = 0
if(num > 1) {
    flag = 1
    for(i in 2:(num-1)) {
        if ((num %% i) == 0) {
            flag = 0
            break
        }
    }
}
if(num == 2)    flag = 1
if(flag == 1) {
    print(paste(num,"is one"))
} else {
    print(paste(num,"is not one"))
}
```

JAVASCRIPT

```
let score = prompt('please enter your
score');             // Score
let msg;            // Message


if (score >= 50) {
  msg = 'Congratulations!';
  msg += ' Proceed to the next round.';
  let el = document.getElementById('answer');
  el.textContent = msg;
}
```

HTML:

```
<div class="var"
id="answer">'Congratulations….</div>
```

What Can Javascript do?

Generative

http://color-wander.surge.sh/

Informative

http://www.histography.io/

Apps

http://ubereats.com

https://www.facebook.com/

https://slack.com/

Entertainment

https://www.netflix.com/

https://www.hulu.com

3D

http://alteredqualia.com/three/examples/webgl_city.html

https://my-room-in-3d.vercel.app/

https://minitokyo3d.com/

Why Javascript?

The Beginning

Mocha or Java?

The Browser Wars

The AJAX revolution

The Standards War

Beyond the Browser

Javascript/Python/C#/R

Quick List of Features

- ➔ Written to enable both-way interaction in web browsers
- ➔ Interpretive: compiled at runtime
- ➔ Always backward-compatible by design
- ➔ Loose type declaration: makes it flexible and confusing at the same time
- ➔ Has functions that can be used as variable objects
- ➔ Allows both functional and object-oriented programming
- ➔ Single-threaded but allows asynchronous events
- ➔ Many ways to implement established design patterns
- ➔ Many popular frameworks: jQuery, Angular, Vue, React, Next
- ➔ Isomorphic - can be used in frontend and servers

Syllabus

- ➔ Syntax and Constructs
- ➔ Document Object Model
- ➔ Forms and AJAX
- ➔ Classes and Object Oriented Programming
- ➔ Functional Programming
- ➔ Modules and DevOps
- ➔ Web/HTML APIs
- ➔ JS in the Backend: Nodejs
- ➔ DevOps Workflows
- ➔ Advanced: Frameworks(React), Applications
- ➔ Final Project Development

Tools of the Trade

➔ Text Editors

Sublime Text:	https://www.sublimetext.com/
Brackets:	https://brackets.io/
MS Visual Studio	https://visualstudio.microsoft.com/vs/mac/
Chrome DevTools:	https://developer.chrome.com/devtools

and more…

➔ Browsers (latest versions)

Chrome:	https://www.google.com/chrome/
Firefox:	https://www.mozilla.org/en-US/firefox/
Safari:	included in macOS

➔ Debugger & Tools

Built in Browser Developer Console (Fn + F12)

Patterns Reference:	https://jstherightway.org/

➔ Automators

NPM, Babel, Gulp
(will be discussed during DevOps session)

Creating a Basic HTML Template

Download from:
https://github.com/kujain/S24-5505_Javascript/blob/main/class-1_html-boilerplate.zip

```
<!doctype html>
<html lang="en">
<head>
  <meta charset="utf-8">
  <meta name="viewport" content="width=device-width">

  <title>The Parsons Web Project</title>
  <meta name="description" content="Spring 23 Class">
  <meta name="author" content="Parsons">

  <link rel="stylesheet" href="css/styles.css">
</head>
<body>
  <header></header>
  <section>
      <h1>Image Test</h1>
      <form id="form" action="#" method="POST">
        <input type="text" name="prompt" id="prompt">
        <button name="submit">Start</button>
        <p id="message" class="error"></p>
      </form>
      <img src="" id="image">
  </section>

  <!-- script always before closing body tag -->
  <script src="js/scripts.js"></script>
</body>
</html>
```

Inline vs External

➔ INLINE:

```
  <body>

    <header>This is the header</header>

    <section>This is Section 1</section>

    <section>
      <button id="button">Click me</button>
    </section>

    <!-- script always before closing body tag -->
    <script>
         console.log(‘Hello’);
         // more stuff
   </script>
  </body>
</html>
```

➔ EXTERNAL:

```
     </section>

    <!-- script always before closing body tag -->
    <script src="js/scripts.js"></script>
  </body>
```

Our First Javascript Code

- https://github.com/kujain/S24-5505_Javascript/blob/main/class-1-exercises.md

- **Hello World!**

```
console.log(‘Hello’);
```

- **Using vars with Hello World!**

```
let greeting_container;

// assign greeting to variable

greeting_container = “Hello”;

console.log(greeting_container);
```

- **Generate an Alert**

```
alert(‘Greetings ’ + greeting_container);
```

- **Update the Document**

```
document.write('<p>’ + greeting_container
+ ’</p>');
```

Our Second Javascript Code

- **Event Listener**

```
/* event listener to change body
background */

const btn =
document.getElementById('button');


const rainbow =
['red','orange','yellow','green','blue','
rebeccapurple','violet'];

function change() {

  document.body.style.background =
rainbow[Math.floor(7*Math.random())];

}

btn.addEventListener('click', change);
```

Our Third Javascript Code

➔ **DOM Manipulation**

```
/* Simple DOM Manipulation example */
const now = new Date();
const hours = now.getHours();

document.write(`It's now: ${hours}. <br><br>`);
let bgColor = "black";

if (hours > 17 && hours < 20){
  bgColor = "orange";
}
else if (hours > 19 && hours < 22){
  bgColor = "orangered";
}
else if (hours > 21 || hours < 5){
  bgColor = "#C0C0C0";
}
else if (hours > 8 && hours < 18){
  bgColor = "lightblue";
}
else if (hours > 6 && hours < 9){
  bgColor = "skyblue";
}
else if (hours > 4 && hours < 7){
  bgColor = "steelblue";
}
else {
  bgColor = "white";
}

document.body.style.backgroundColor = bgColor;
```

Our 4th Javascript - Code

➔ Connect with API using AJAX

```
document.getElementById('form').addEventListener('submit', function(e) {
     e.preventDefault();
     const textprompt = document.getElementById('prompt').value;
     if ( ! textprompt ) {
           document.getElementById('message').innerHTML = 'Please enter
a prompt first';
     } else {
           let json = JSON.stringify({
                      text_prompts: [
                           {
                                text: textprompt,
                           },
                      ],
                      cfg_scale: 7,
                      height: 1024,
                      width: 1024,
                      steps: 30,
                      samples: 1,
                 });
           const apiKey =
'sk-sXqFPCq8PgwBE1JtPeSkFhGJ1djMlIR8maSi8wvQvYvnHtY3';
           const engineId = 'stable-diffusion-v1-6'
           fetch(
`https://api.stability.ai/v1/generation/${engineId}/text-to-image`,
                 {
                      method: 'POST',
                      headers: {
                           'Content-Type': 'application/json',
                           Accept: 'application/json',
                           Authorization: `${apiKey}`,
                      },
                      body: json,
                 })
                 .then(response => response.json())
                 .then(json => {
                      let img =json.artifacts[0].base64;
                      document.getElementById('image').setAttribute('src'
, 'data:image/jpeg;base64,'+img);
                 })
```

Our 4th Javascript Code

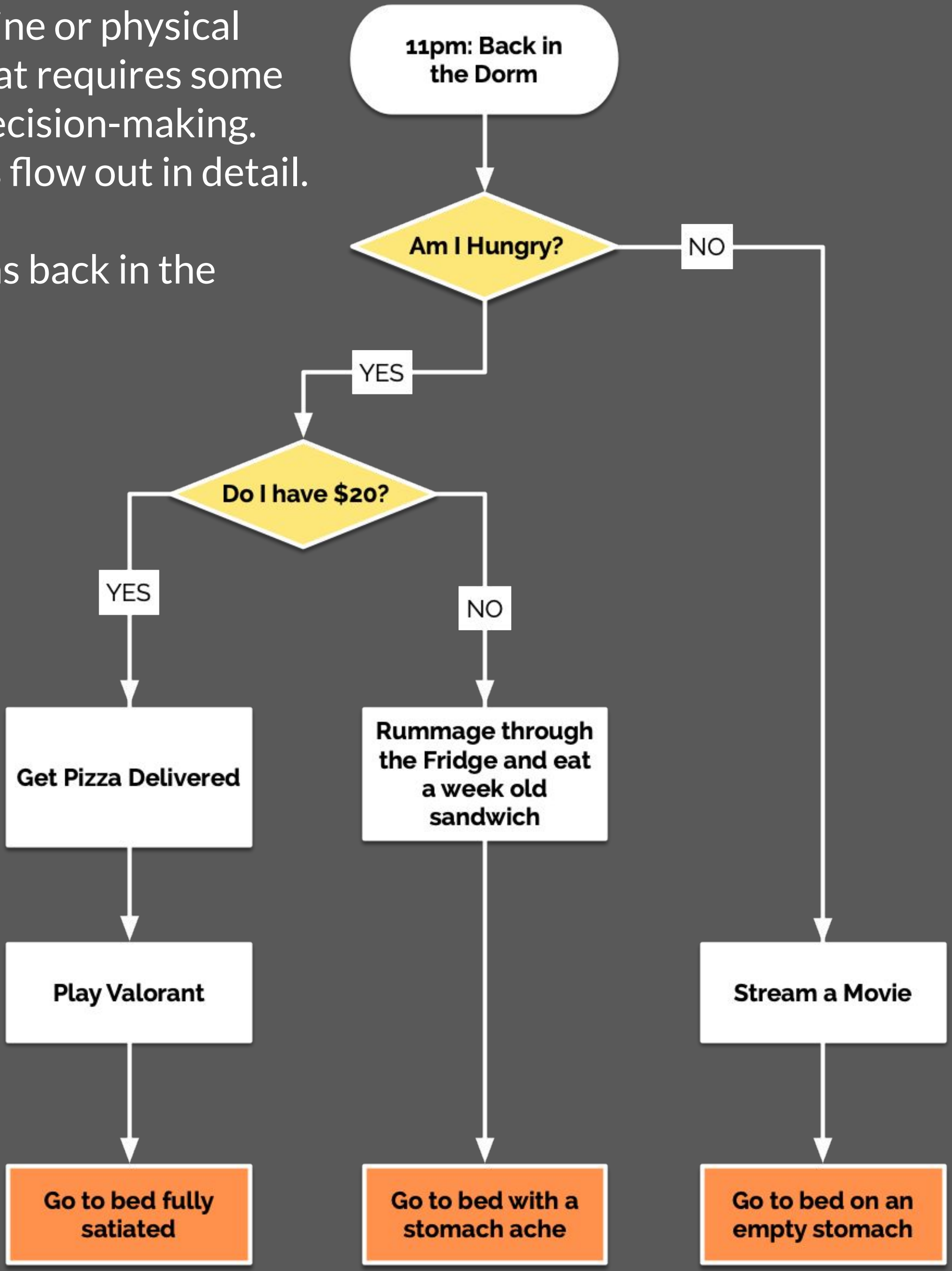
➔ Downloadable files:

➔ https://github.com/kujain/S24-5505_Javascript/blob/main/class-1_exercise-4.zip

Assignment: Decision Trees

Find a regular online or physical activity or task that requires some interaction and decision-making. Write the process flow out in detail.

eg. Dining Options back in the Dorm:



Next Class

- ➔ **Javascript Structure**
- ➔ **Javascript Syntax:**

  Data types: strings, numbers, variables, arrays

  Operators

  Conditional logic

  Loops etc.