

CSCI E-33a

CS50's Web Programming
with Python and JavaScript

Spring 2020

Barbara Karakyriakou · Teaching Fellow

- Email: karakyriakou@fas.harvard.edu
- Phone: 617 820 6930
- Section Meetings: Wednesdays 8:30pm-10:00pm EST
- Office Hours: Saturdays 1:00pm - 2:30pm EST

Section 5: User Interfaces

Agenda

- Single page applications
 - Exercise
- Window Object
 - Exercise
- Q&A on Problem Set 3

Single Page Applications

A single-page application is a web application that interacts with the web browser by rewriting a web page dynamically, instead of a browser loading entire new pages.

<https://blog.angular-university.io/why-a-single-page-application-what-are-the-benefits-what-is-a-spa/>

Single Page Example

```
1  <!DOCTYPE html>
2  <html lang="en">
3  <head>
4      <title>Single Page</title>
5      <style>
6          div {
7              display: none;
8          }
9      </style>
10     <script>
11
12         function showPage(page) {
13             // Hide all other pages
14             document.querySelectorAll('div').forEach(div => {
15                 div.style.display = 'none';
16             });
17
18             // Show requested page
19             document.querySelector(`#${page}`).style.display = 'block';
20         }
21
22         document.addEventListener('DOMContentLoaded', function() {
23             document.querySelectorAll('button').forEach(button => {
24                 button.onclick = function() {
25                     showPage(this.dataset.page);
26                 };
27             });
28         });
29     </script>
30 </head>
31 <body>
32     <button data-page="page1">Page 1</button>
33     <button data-page="page2">Page 2</button>
34     <button data-page="page3">Page 3</button>
35     <div id="page1">
36         <h1>This is page 1.</h1>
37     </div>
38     <div id="page2">
39         <h1>This is page 2.</h1>
40     </div>
41     <div id="page3">
42         <h1>This is page 3.</h1>
43     </div>
44 </body>
45 </html>
```

Single Page & Django Exercise

- Create a Django project: `django-admin startproject projectname`
- Create an app within your project: `python manage.py startapp singlepage`
- Add a template folder and a `urls.py` in your app folder
- Add your app to the `settings.py` of your project in the list of `INSTALLED_APPS`
- Add a path to your apps `urls` in the `urls.py` of your project folder
- Create a template for your single page app and save it in the templates folder. Your template should behave as a single page website
- Create python function(s) in `views.py` required to render your html template and its different sections
- Add paths to your views in the `urls.py` within your app

JavaScript Window Object

The JavaScript Window Object is a browser object model that allows JavaScript to “talk” to the browser. The Window object represents the browser’s window and it is supported by all browsers

Excercise

- Go back to your Django project and to your app folder. Open your HTML template and modify your script to adds a state to the browser's session history stack.
 - On your show section function add a line of code that uses the `history.pushState()` method
 - Add a popstate event of the JavaScript Window interface: `window.onpopstate`

JavaScript Window Object

Exercise cont'd

- `history.pushState()`

Syntax :

```
history.pushState(state object, title, url)
```

Example:

```
history.pushState({section:this.dataset.section}, "", `section${this.dataset.section}`)
```

- `window.onpopstate`

Example:

```
window.onpopstate = function(event) {  
    console.log(event.state.section);  
    showSection(event.state.section);  
}
```

JavaScript Window Object

Window Object Properties

Two properties can be used to determine the size of the browser window.

- `window.innerHeight` - the inner height of the browser window (in pixels)
- `window.innerWidth` - the inner width of the browser window (in pixels)
- `window.scrollY` - returns the number of pixels that the document is currently scrolled vertically (Y axis)

JavaScript Window Object

Window Object Other Methods

- `window.open()` - open a new window
- `window.close()` - close the current window
- `window.moveTo()` - move the current window
- `window.resizeTo()` - resize the current window
- `window.onscroll()` - execute an event when the window is being scrolled

Q&A