ЗАНЯТТЯ 11 ОГЛЯД СИНТАКСИСУ JAVASCRIPT ТА РОБОТИ З DOM

ТЕМИ

27. Javascript як мова програмування

- Виділення скрипта в окремий файл. Структура файлу js - функціональний стиль.
- Основні синтаксичні елементи JS - змінні, умовні конструкції, цикли.
- Типи змінних. Слабка типізація.
- Функції - базовий синтаксис, передача параметрів

Практика:

- Розглядаємо приклади роботи зі змінними та маніпулювання з DOM (на вибір ментора)

РЕКОМЕНДАЦІЇ ДО ПРОВЕДЕННЯ ЗАНЯТТЯ

1. **Обговорюємо разом домашнє завдання**
   - Дивимось варіанти учасників. Якщо вдається, то дороблюємо (демонструючи групі та закликаючи групу допомогти) два-три варіанти тих, в кого щось не вийшло (але напрацювання є).

2. **Підключення зовнішніх скриптів**
   - Пояснюємо, чому не варто писати більшість коду в середині тегу <script> в основному HTML.
   - Демонструємо, як підключати скрипт з окремого файлу, і разом з групою переробляємо приклад з попереднього заняття (виносимо код в окремий файл, і його підключаємо).

3. **Базовий синтаксис javascript.**
   - Розраховуємо на те, що учасники мають базові знання з програмування хоча б якоюсь мовою - тобто поняття “змінна”, “if”, “цикл” та “функція” вони розуміють.
   - Пропонуємо групі повторювати експерименти ментора (ймовірно, варто робити максимально прості демонстрації - без маніпуляції документом, на рівні арифметики та виводу значень в консоль браузера).
   - Демонструємо базові конструкції мови, без ООП та масивів. Змінні, деякі оператори (варто зупинитись на операторах “присвоєння” та “порівнняння” (=, ==, згадати про ===)) , if, for.
   - Після демонстрації змінних робимо теоретичний відступ щодо того, що js - це мова із слабкою типізацією.
   - Також можна згадати, що синтаксис подібний до мови “C”.
   - Демонструємо роботу з функціями на JS (в базовому вигляді, без анонімних функцій). Показуємо створення, виклик, передачу параметрів та повернення результату.

4. **Утилітарно вводимо поняття “об’єктів, які надає браузер” (window, document).** Якщо раптом група знайома з ООП - то все просто. Якщо, що більш імовірно, ні - то кажемо, що для зручності JS і багато інших мов дозволяють згрупувати змінні та функції разом, і таке групування іноді називають об’єктом (на розсуд ментора, чи варто намагатись пояснювати різницю між класом та об’єктом). Далі, як варіант, можна розповісти та продемонструвати приблизно наступним чином:
   - Якщо в деякій змінній зберігається об’єкт, то ми можемо через крапку отримувати чи змінювати значення його змінних (які часто називають полями (fields) або властивостями (properties)), а також викликати його функції (які часто називають методами (methods)).
   - При цьому поля об’єкта часто, в свою чергу, містять деякі об’єкти. І методи об’єкта також часто повертають об’єкти.
   - Сам браузер забезпечує, що в будь-якому місці коду ми можемо використовувати деякі вбудовані об’єкти. Насправді, доступний об’єкт window, а більшість іншого - це змінні в ньому, які є також об’єктами - але історично склалося, що ми можемо звертатися до них не лише як window.щось, а і просто як до “щось”. Наприклад, наш знайомий виклик console.log(“Hello, teachers!”) - насправді є викликом window.console.log(...) - тобто змінна з іменем window, яка надається нам браузером, містить об’єкт, в якому є змінна console, яка містить об’єкт, у якого є метод (функція) log(), яка виводить свої аргументи (те, що ми їй передамо) в консоль браузера.
   - зокрема в window (а отже і глобально) доступна змінна document, яка містить об’єкт, через який ми можемо викликати купу корисних функцій - знову продемонструвати отримання елементу по ІД.
   - також варто продемонструвати зміну або отримання location.

5. **Трохи глибше, ніж минулого разу, торкаємось поняття DOM.**
   - Розповідаємо, що браузер “парсить” html - тобто аналізує його, і створює всередині себе структуру, схожу на дерево, яку називають ДОМ. І надає нам, як розробникам, можливість цим “домом” маніпулювати, знаходячи в ньому елементи, змінюючи їх властивості, а також додаючи нові та видаляючи існуючі елементи.
   - Через цю призму ще раз дивимось на наш приклад, в якому ми заповнювали картинки в сторінці, або список з домашнього завдання.
   - Демонструємо, що можна будувати в такий спосіб і більш складні конструкції (блоки HTML) - наприклад, генеруємо список, кожен елемент якого - це з текстом та картинкою в середині - приблизно такий:
     - <h1>Image-1</h1>
     - <h1>Image-2</h1>
     - ……

ДОМАШНЄ ЗАВДАННЯ (ТЕОРІЯ)

- Курс з веб-розробки на W3Schools, тема “JavaScript Where To” (англ)
- Курс з веб-розробки на W3Schools, тема “JavaScript Syntax” (англ)
- Курс з веб-розробки на W3Schools, тема “JavaScript Functions” (англ)

- Mozilla Developer Network (MDN) «A first splash into JavaScript» (англ)
- Mozilla Developer Network (MDN) «JavaScript building blocks» (англ)
- Mozilla Developer Network (MDN) «Making decisions in your code — conditionals» (англ)
- Mozilla Developer Network (MDN) «Looping code» (англ)
- Mozilla Developer Network (MDN) «Functions — reusable blocks of code» (англ)
- Mozilla Developer Network (MDN) «Storing the information you need — Variables» (англ)
- Mozilla Developer Network (MDN) «Basic math in JavaScript — numbers and operators» (англ)
- Mozilla Developer Network (MDN) «Handling text — strings in JavaScript» (англ)

ДОМАШНЄ ЗАВДАННЯ (ПРАКТИКА)

- Реалізувати генератор “каталогу плиткою” на javascript. Маючи сторінку, яка містить весь html окрім самого набору елементів каталогу (можна взяти сторінку з відповідного завдання, та вирізати ці елементи з html), підключити до неї скрипт-файл, в якому написати код, який додасть в потрібне місце 100 елементів-плиток. Кожна плитка повинна як мінімум містити заголовок (текст вигляду “Елемент каталогу номер N”, де N змінюється для кожного елементу) і картинку. Окрім того, кожен другий елемент повинен мати заголовок зеленого коліру та картинку в рамочці також зеленого коліру.
- Для того, щоб не створювати сто реальних картинок, можна використати сервіс “picsum” - можна генерувати посилання на картинки вигляду https://picsum.photos/290/290/?N (тут 290 - це висота і ширина картинки, їх можна задати такі, які потрібно, а замість “N” в кінці може бути будь-яке ціле число, і тоді картинки будуть різними). Або ж можна кинути в папку кілька картинок з іменами img1.jpg, img2.jpg і т.д, і змусити скрипт генерувати їх по колу - тобто, картинки будуть повторюватись.
- Обговорюємо фінальні проекти та формуємо список тем робіт слухачів до захисту.

ДОДАТКОВІ КОРИСНІ МАТЕРІАЛИ

- Навчальна платформа Devionity, курс Основы CSS, заняття «Основы синтаксиса. Переменные и константы» (рос)
- Навчальна платформа Devionity, курс Основы CSS, заняття «Типы данных. Undefined и Null» (рос)
- Навчальна платформа Devionity, курс Основы CSS, заняття «Управляющие конструкции: условия» (рос)
- Навчальна платформа Devionity, курс Основы CSS, заняття «Циклы. Цикл for()» (рос)
- Навчальна платформа Devionity, курс Основы CSS, заняття «Основы функций» (рос)