

Структура коду

№ уроку: 2 **Курс:** JavaScript Starter

Засоби навчання: Visual Studio Code
Web Browser

Огляд, ціль та призначення уроку

Вивчити способи застосування сценаріїв на HTML сторінці і розібрати основні принципи побудови сценарію.

Вивчивши матеріал даного заняття, учень зможе:

- Розуміти, що таке інструкція.
- Використання коментарів у коді.
- Підключати сценарії різними способами.
- Розуміти різницю між компільованими та інтерпретованими мовами програмування.

Зміст уроку

1. Інструкції
2. Коментарі
3. Способи підключення сценарію
4. Інтерпретація сценарію

Резюме

- **Інструкція** – синтаксична одиниця мови програмування, що виражає дію, яку виконує програмний код. Весь сценарій складається з набору інструкцій, які спрацьовують послідовно одна за одною. Інструкції можуть складатися із вкладених компонентів – виразів.
Кожна завершена інструкція повинна закінчуватися крапкою з комою. Перенесення на новий рядок також може бути розцінено як завершення інструкції, але для запобігання помилкам краще використовувати ;
- **Коментар** – пояснення тексту програми. Коментарі можуть знаходитись у будь-якому місці коду сценарію та не впливають на його виконання.

```
// однорядковий коментар  
/*  
    Багаторядковий коментар  
*/
```

У багатьох редакторах комбінація клавіш **Ctrl + /** коментує виділений фрагмент коду використовуючи `//` для кожного виділеного рядка.

Не рекомендується коментувати код. Завдання коментаря - пояснити іншому розробнику, чи нагадати вам у майбутньому, навіщо було обрано той чи інший алгоритм чи опис алгоритму загалом, якщо він складний.

При зміні коду слід також змінювати коментарі до цього коду, оскільки може виникнути ситуація, коли коментар описує те, що вже в коді не відбувається.

Слід намагатися писати самодокументований код – коли назви змінних, методів та інших ідентифікаторів максимально пояснюють призначення коду та уможливають його читання без спеціальних коментарів.

- Підключення сценарію виконується за допомогою `script`. Сценарій може бути скопійований між елементом відкриваючого та закриваючого тегу `script`, або доданий як посилання на зовнішній файл через атрибут `src`, при цьому вміст елемента `script` ігнорується.

Рекомендується використовувати підключення через атрибут `src`, тому що браузер може зберегти файл (помістити його в кеш, «закешувати») і при повторному завантаженні сторінки не завантажувати файл із сервера, а взяти його з кеш, що прискорює завантаження сторінки.

Оптимальне місце підключення сценарію в HTML документі – перед закриваючим елементом `body`. Кожен сценарій повинен бути проаналізований браузером перед тим, як відбудеться візуалізація елементів після `<script>`. Якщо `<script>` знаходиться на початку сторінки – сторінка не з'явиться, доки скрипт не буде проаналізований браузером. Якщо скрипт знаходиться в кінці сторінки, користувач буде бачити вміст і взаємодіяти з ним, в той час як браузер буде аналізувати JavaScript код.

Існує ситуація, коли сценарій повинен бути підключений на початку сторінки. Наприклад, коли дія, визначена в JavaScript коді, повинна обов'язково спрацювати перед тим, як користувач почне працювати зі сторінкою - імітація можливостей, які не підтримує браузер, облік відкриттів сторінки тощо.

- JavaScript код виконується послідовно, інструкція за інструкцією. JavaScript - мова програмування, що інтерпретується.

Інтерпретація – порядковий аналіз (по рядку), обробка та виконання вихідного коду програми.

Компіляція – складання програми з перетворенням вихідного коду, написаного високорівневою мовою програмування, в програмні модулі низькорівневою мовою.

При *інтерпретації*, про помилку можна дізнатися під час виконання коду. Код до помилки виконується, після помилки – перестає інтерпретуватися.

При компіляції наявність помилки в коді призведе до помилки компіляції.

Переваги інтерпретованих мов програмування (JavaScript, Python, Perl):

- Динамічна типізація
- Незалежність від платформи
- Можливість модифікувати програму під час виконання

- Відносно менші витрати часу на розробку та налагодження

Переваги компільованих мов програмування (C#, C++, Java):

- Швидкість виконання
- Визначення помилок на етапі компіляції
- Строга типізація

Закріплення матеріалу

- Що таке інструкція?
- Що таке коментар, які бувають коментарі?
- Що таке інтерпретована мова програмування, які переваги таких мов?
- Що таке компільована мова програмування, які її переваги?
- Як можна підключити сценарій JavaScript на сторінці?

Додаткові завдання

Завдання

Переробіть завдання з першого уроку таким чином, щоб завантажувався сценарій з окремого файлу.

Самостійна діяльність учня

Виконайте завдання у директорії Exercises\Tasks\02 JS Structure у матеріалах до цього уроку.

Рекомендовані ресурси

Елемент Script

<https://developer.mozilla.org/ru/docs/Web/HTML/Element/script>

Граматика та типи даних

https://developer.mozilla.org/ru/docs/Web/JavaScript/Guide/Grammar_and_Types