

Структура коду

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

Засоби навчання: Visual Studio Code
Web Browser

Огляд, мета та призначення уроку

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

Вивчивши матеріал цього заняття, учень зможе:

- Розуміти, що таке інструкція.
- Використати коментарі в коді.
- Під'єднувати сценарії різними способами.
- Розуміти різницю між компільованими й інтерпретованими мовами програмування.

Зміст уроку

1. Інструкції.
2. Коментарі.
3. Способи під'єднання сценарію.
4. Інтерпретація сценарію.

Резюме

- **Інструкція** – синтаксична одиниця мови програмування, яка виражає дію, що виконується програмним кодом. Весь сценарій складається з набору інструкцій, які спрацьовують послідовно, одна за одною. Інструкції можуть складатися із вкладених компонентів – виразів.
Кожна завершена інструкція має закінчуватися крапкою з комою. Перенесення на новий рядок також може бути сприйнято як завершення інструкції, але для запобігання помилок краще використовувати «;».
- **Коментар** – пояснення тексту програми. Коментарі можуть розташовуватися 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 очевидний код. Завдання коментаря – пояснити іншому розробнику або нагадати вам у майбутньому, навіщо було обрано той чи інший алгоритм або загалом опис алгоритму, якщо він складний.

Під час зміни коду варто також змінювати коментарі до цього коду, оскільки може виникнути ситуація, коли коментар описує те, що вже в коді не відбувається.

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

- Під'єднання сценарію здійснюється за допомогою елемента **script**. Сценарій може бути скопійований між відкривальним і закривальним елементами тегу **script** або бути доданим як посилання на зовнішній файл через атрибут **src**, водночас вміст елемента **script** ігнорується.

Рекомендується використовувати під'єднання через атрибут **src**, оскільки браузер може зберегти файл (помістити його в кеш, «закешувати») і під час повторного завантаження сторінки не завантажувати файл із сервера, а взяти його з кешу, що прискорює процес завантаження сторінки.

Оптимальне місце під'єднання сценарію в HTML-документі – перед закривальним елементом body. Кожен сценарій має бути проаналізований браузером перед тим, як відбудеться візуалізація елементів, які розташовані після **<script>**. Якщо **<script>** розташований на початку сторінки, вона не з'явиться, доки **<script>** не буде проаналізований браузером. Якщо **<script>** розташований наприкінці сторінки, користувач бачитиме вміст і взаємодіятиме з ним, у той час, як браузер аналізуватиме JavaScript-код.

Є ситуації, коли сценарій має бути під'єднаний на початку сторінки. Наприклад, коли дія, яка визначена в JavaScript-коді, має 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 – інтерпретована мова програмування.

Інтерпретація – рядковий аналіз, обробка та виконання вихідного коду програми.

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

При *інтерпретації* про помилку можна дізнатися під час виконання коду. Код до помилки виконується, а 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.
- Порівняно менше витрат часу на розробку та налагодження.

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

- Швидкість виконання.
- Визначення помилок на етапі компіляції.

- Сувора 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 під'єднати JavaScript-сценарій на сторінці?

Додаткове завдання

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

Самостійна діяльність учня

Виконайте завдання в 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>