

Структура кода

№ урока: 2 **Курс:** Структура кода

Средства обучения: Visual Studio Code
Web Browser

Обзор, цель и назначение урока

Изучить способы внедрения сценариев на HTML странице и разобрать основные принципы построения сценария.

Изучив материал данного занятия, учащийся сможет:

- Понимать, что такое инструкция.
- Использовать комментарии в коде.
- Подключать сценарии разными способами.
- Понимать разницу между компилируемыми и интерпретируемыми языками программирования.

Содержание урока

1. Инструкции
2. Комментарии
3. Способы подключения сценария
4. Интерпретация сценария

Резюме

- **Инструкция** – синтаксическая единица языка программирования, выражающая действие, выполняемое программным кодом. Весь сценарий состоит из набора инструкций, которые срабатывают последовательно, друг за другом. Инструкции могут состоять из вложенных компонентов – выражений.
Каждая завершенная инструкция должна заканчиваться точкой с запятой. Перенос на новую строку также может быть расценен как завершение инструкции, но для предотвращения ошибок лучше использовать ;
- **Комментарий** – пояснения к тексту программы. Комментарии могут находиться в любом месте кода сценария и не влияют на его выполнение.

```
// однострочный комментарий  
/*  
    Многострочный  
    комментарий  
*/
```

Во многих редакторах комбинация клавиш **Ctrl + /** комментирует выделенный фрагмент кода используя // для каждой выделенной строки.

Не рекомендуется комментировать очевидный код. Задача комментария - объяснить другому разработчику, или напомнить вам в будущем, зачем был выбран тот или иной алгоритм или описание алгоритма в целом, если он сложный.

При изменении кода, следует также изменять комментарии к этому коду, так как может возникнуть ситуация, когда комментарий описывает то, что уже в коде не происходит.

Следует стараться писать само документируемый код – когда названия переменных, методов и других идентификаторов максимально объясняют назначение кода и делают возможным его чтение без специальных комментариев.

- Подключение сценария выполняется с помощью элемента `script`. Сценарий может быть скопирован между открывающим и закрывающим элементами тега `script` или добавлен как ссылка на внешний файл через атрибут `src`, при этом содержимое элемента `script` игнорируется.

Рекомендуется использовать подключение через атрибут `src`, так как браузер может сохранить файл (поместить его в кэш, «закешировать») и при повторной загрузке страницы не скачивать файл с сервера, а взять его из кэш, что ускоряет загрузку страницы.

Оптимальное место подключения сценария в HTML документе – перед закрывающим элементом `body`. Каждый сценарий должен быть проанализирован браузером перед тем, как произойдет визуализация элементов, находящихся после `<script>`. Если `<script>` находится в начале страницы – страница не отобразится, пока скрипт не будет проанализирован браузером. Если скрипт находится в конце страницы, то пользователь будет видеть содержимое и взаимодействовать с ним, в то время как браузер будет анализировать JavaScript код.

Есть ситуации, когда сценарий должен быть подключён в начале страницы. Например, когда действие, определенное в JavaScript коде, должно обязательно сработать, перед тем как пользователь начнет работать со страницей – имитация возможностей, которые не поддерживает браузер, учет открытых страницы и т.д.

- JavaScript код выполняется последовательно, инструкция за инструкцией. JavaScript – интерпретируемый язык программирования.

Интерпретация – построчный анализ, обработка и выполнение исходного кода программы.

Компиляция – сборка программы с преобразованием исходного кода, написанного на высокоуровневом языке программирования, в программные модули на низкоуровневом языке.

При *интерпретации*, об ошибке можно узнать во время выполнения кода. Код до ошибки выполняется, после ошибки - перестает интерпретироваться.

При *компиляции*, наличие ошибки в коде приведет к ошибке компиляции.

Преимущества интерпретируемых языков программирования (JavaScript, Python, Perl):

- Динамическая типизация
- Независимость от платформы
- Возможность модифицировать программу во время выполнения

- Относительно меньше затрат времени на разработку и отладку

Преимущества компилируемых языков программирования (C#, C++, Java):

- Скорость выполнения
- Определение ошибок на этапе компиляции
- Строгая типизация

Закрепление материала

- Что такое инструкция?
- Что такое комментарий, какие бывают комментарии?
- Что такое интерпретируемый язык программирования, какие преимущества таких языков?
- Что такое компилируемый язык программирования, какие его преимущества?
- Как можно подключить JavaScript сценарий на странице?

Дополнительное задание

Задание

Переделайте задание из первого урока таким образом, чтобы сценарий загружался из отдельного файла.

Самостоятельная деятельность учащегося

Выполните задания в директории Exercises\Tasks\02 JS Structure в материалах к этому уроку.

Рекомендуемые ресурсы

Элемент Script

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

Грамматика и типы данных

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