

Переменные и типы данных

№ урока: 1 **Курс:** Переменные и типы данных

Средства обучения: Visual Studio Code
Web Browser

Обзор, цель и назначение урока

Научиться использовать переменные и понимать, какие типы данных существуют в JavaScript.

Изучив материал данного занятия, учащийся сможет:

- Создавать и использовать переменные.
- Создавать и использовать константы.
- Понимать принципы правильного именования переменных.
- Работать с типами данных.

Содержание урока

1. Переменные и константы
2. Использование консоли браузера
3. Рекомендации по именованию переменных
4. Типы данных в JavaScript

Резюме

- **Переменная** – именованная область памяти, которая хранит в себе некоторое значение, которое можно изменить.
Для создания переменных используют ключевое слово **let**. Также можно использовать ключевое слово **var**, но оно устаревшее и желательно избегать его применения.
Для того, чтобы задать значение для переменной используется знак присвоения **=**.

let x = 10; - создаем переменную с именем x и присваиваем ей значение 10.
- Для отслеживания работы сценария, часто используется консоль из инструментов разработчика. Чтобы получить доступ к инструментам разработчика в большинстве браузеров нужно нажать на клавишу F12.
Для вывода данных в консоль используется инструкция **console.log(сообщение)**.
- **Константа** – постоянная величина. В JavaScript константа – переменная, значение которой нельзя изменить. При попытке изменить константу в консоли браузера появится сообщение об ошибке. Пример значения, которое желательно сделать константой – число Пи.
- Язык JavaScript - регистрозависимый язык. Message и message – две разные переменные, из-за того, что первый символ имеет разный регистр. При именовании следует придерживаться следующих правил:

- Имена переменных должны содержать буквы, цифры или символы \$ и _
- Первым символом не должна быть цифра
- Если имя переменной содержит несколько слов, используется camelCasing
- **camelCasing** – принцип именования, при котором первый символ названия в нижнем регистре, а далее, каждый первый символ слова в составном имени - находится в верхнем регистре. Примеры: firstName, minWidthValue, errorMessage.
camelCasing принято использовать для большинства идентификаторов в JavaScript.
- **PascalCasing** – принцип именования идентификаторов, при котором все слова, которые формируют идентификатор, начинаются с символа в верхнем регистре. Примеры: FirstName, MinWidthValue, ErrorMessage.
PascalCasing используют при именованиях классов и функций конструкторов (эти темы входят в следующий курс).
- **kebab-casing** – каждое слово в идентификаторе отделяется дефисом и пишется в нижнем регистре. Примеры: first-name, min-width-value, error-message.
Обычно используется при написании CSS селекторов.
- При именованиях переменных стоит выбирать описательные имена, которые позволят понимать за что отвечает значение, находящееся в переменной.
- **Тип данных** – допустимое множество значений. Тип данных описывает набор операций, которые можно над этими данными проводить.
- JavaScript – **динамически типизированный** язык программирования. При создании переменной ее тип не указывается явно, но переменная связывается с типом при присвоении значения. При смене значения переменной может измениться и тип данных переменной. Каждая переменная имеет тип данных, просто мы не указываем этот тип явно, как это происходит в других языках со *статической типизацией*.
- Типы данных, которые есть в JavaScript и примеры того, как создать переменную указанного типа данных:

| | | |
|-----------|---------------------------------------|-------------------------|
| number | для хранения любых чисел | let x = 10; |
| bigint | целые числа произвольной длины | let x = 10n; |
| string | для хранения строк | let x = "hello"; |
| boolean | логические значения true или false | let x = true; |
| null | для определения неизвестного значения | let x = null; |
| undefined | для не присвоенных значений | let x = undefined; |
| object | сложные структуры данных | let x = { } |
| symbol | уникальные идентификаторы | let x = Symbol("test"); |

Типы данных, с которыми вы будете чаще всего встречаться на протяжении этого курса и на практике – number, string, boolean, undefined, object.

Object – сложный тип данных, который позволяет сохранить несколько значений в одной переменной.

- Оператор **typeof** позволяет получить тип данных переменной в виде строкового значения. Для переменных, в которых хранится функция, typeof вернет значение

"function", хотя это не отдельный тип данных. Переменные, в которых находится в качестве значения функция, являются типа object. Это особенность ключевого слова typeof, которая часто используется на практике. Функции будут рассмотрены в 11 уроке этого курса.

Закрепление материала

- Что такое переменная?
- Что такое константа? Как константу создать в JavaScript?
- Что такое camelCasing?
- Что такое PascalCasing и kebab-casing?
- Является ли JavaScript регистрозависимым языком программирования?
- Какого типа данных переменная, если при ее инициализации использовалось значение 10?
- Какого типа данных переменная, если при ее инициализации использовалось значение "hello world"?
- Назовите все типы данных JavaScript.
- Что такое динамическая типизация?
- Можно ли поменять тип данных переменной, которая уже была объявлена?
- Что делает оператор typeof?

Дополнительное задание

Задание

Создайте три переменные с именами a, b, c. Присвойте этим переменным значения 1, 2 и 3. Выведите на консоль сумму значений переменных a, b и c.

Самостоятельная деятельность учащегося

Выполните задания в файлах директории Exercises\Tasks\03 Variables and Data Types

Рекомендуемые ресурсы

Типы данных в JavaScript

https://developer.mozilla.org/ru/docs/Web/JavaScript/Data_structures

Ключевые слова JavaScript

https://developer.mozilla.org/en-US/docs/Web/JavaScript/Reference/Lexical_grammar#Keywords

Оператор typeof

<https://developer.mozilla.org/ru/docs/Web/JavaScript/Reference/Operators/typeof>