

Змінні та типи даних

№ уроку: 3 Курс: JavaScript Starter

Засоби навчання: Visual Studio Code
Web Browser

Огляд, ціль та призначення уроку

Навчитися використовувати змінні та розуміти, які типи даних існують у JavaScript.

Вивчивши матеріал даного заняття, учень зможе:

- Створювати та використовувати змінні.
- Створювати та використовувати константи.
- Розуміти принципи правильної назви змінних.
- Працювати з типами даних.

Зміст уроку

1. Змінні та константи
2. Використання консолі браузера
3. Рекомендації щодо іменування змінних
4. Типи даних в JavaScript

Резюме

- **Змінна** – назва пам'яті, що зберігає певне значення, яке можна змінити. Для створення змінних використовують ключове слово **let**. Також можна використовувати ключове слово **var**, але воно застаріле та бажано уникати його застосування. Щоб встановити значення для змінної, використовується знак присвоєння **=**.

`let x = 10;` - створюємо змінну `x` і присвоюємо їй значення 10.
- Для відстеження роботи сценарію часто використовується консоль з інструментів розробника. Щоб отримати доступ до інструментів розробника, у більшості браузерів потрібно натиснути на клавішу F12. Для виведення даних у консоль використовується інструкція **console.log(повідомлення)**.
- **Константа** – постійна величина. JavaScript константа – змінна, значення якої не можна змінити. Під час спроби змінити константу в консолі браузера з'явиться повідомлення про помилку. Приклад значення, яке бажано зробити константою, – число π .
- Мова JavaScript - реєстрозалежна мова. `Message` і `message` – дві різні змінні через те, що перший символ має різний регістр. При назві слід дотримуватися таких правил:

- Імена змінних повинні містити літери, цифри або символи \$ та _
- Першим символом не повинна бути цифра
- Якщо ім'я змінної містить кілька слів, використовується camelCasing
- **camelCasing** – принцип іменування, при якому перший символ назви в нижньому регістрі, а далі, кожен перший символ слова в складовому імені знаходиться у верхньому регістрі. Приклади: firstName, minWidthValue, errorMessage.
camelCasing прийнято використовувати для більшості ідентифікаторів в JavaScript.
- **PascalCasing** – принцип іменування ідентифікаторів, коли всі слова, які формують ідентифікатор, починаються з символу у верхньому регістрі. Приклади: FirstName, MinWidthValue, ErrorMessage.
PascalCasing використовують при назві класів і функцій конструкторів (ці теми входять у наступний курс).
- **kebab-casing** – кожне слово в ідентифікаторі відокремлюється дефісом та пишеться в нижньому регістрі. Приклади: first-name, min-width-value, error-message.
Зазвичай використовується під час написання CSS селекторів.
- При іменуванні змінних варто вибирати описові імена, які дозволять розуміти, за що відповідає значення, яке знаходиться в змінній.
- **Тип даних** – допустима кількість значень. Тип даних визначає набір операцій, які можна проводити над цими даними.
- JavaScript – **динамічно типізована** мова програмування. Під час створення змінної її тип не вказується явно, але змінна зв'язується з типом при присвоєнні значення. При зміні значення змінної може змінитися тип даних змінної. Кожна змінна має тип даних, ми не вказуємо цей тип явно, як це відбувається в інших мовах зі *статичною типізацією*.
- Типи даних, які є в JavaScript, та приклади того, як створити змінну зазначеного типу даних:

number	для зберігання будь-яких чисел	let x = 10;
biging	цілі числа довільної довжини	let x = 10n;
string	для зберігання рядків	let x = "hello";
boolean	логічні значення true або false	let x = true;
null	для визначення невідомого значення	let x = null;
undefined	для не наданих значень	let x = undefined;
object	складні структури даних	let x = { }
symbol	унікальні ідентифікатори	let x = Symbol("test");

Типи даних, з якими ви будете найчастіше зустрічатися протягом цього курсу та на практиці, – number, string, boolean, undefined, object.

Object – складний тип даних, який дозволяє зберегти кілька значень в одній змінній.

- Оператор **typeof** дозволяє отримати тип даних змінної у вигляді рядкового значення. Для змінних, в яких зберігається функція, typeof поверне значення

"function", хоча це не окремий тип даних. Змінні, в яких знаходиться як значення функції, є тип object. Це особливість ключового слова typeof, яка часто використовується на практиці. Функції будуть розглянуті у 11 уроці цього курсу.

Закріплення матеріалу

- Що таке змінна?
- Що таке константа? Як створити константу в JavaScript?
- Що таке camelCasing?
- Що таке PascalCasing та kebab-casing?
- Чи є JavaScript регістрозалежною мовою програмування?
- Якого типу даних є змінна, якщо при її ініціалізації використовувалося значення 10?
- Якого типу даних є змінна, якщо при її ініціалізації використовувалося значення "hello world"?
- Назвіть усі типи даних JavaScript.
- Що таке динамічна типізація?
- Чи можна змінити тип даних змінної, яка вже була оголошена?
- Що робить оператор typeof?

Додаткові завдання

Завдання

Створіть три змінні з іменами a, b, c. Дайте цим змінним значення 1, 2 і 3. Виведіть на консоль суму значень змінних a, b і c.

Самостійна діяльність учня

Виконайте завдання у файлах директорії Exercises\Tasks\03 Variables and Data Types

Рекомендовані ресурси

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

Ключові 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>