

Змінні та типи даних

№ уроку: 3 Курс: JavaScript Starter

Засоби навчання: Visual Studio Code
Web Browser

Огляд, мета та призначення уроку

Навчитися використовувати змінні та розуміти, які типи даних наявні в JavaScript.

Вивчивши матеріал цього заняття, учень зможе:

- Створювати та використовувати змінні.
- Створювати та використовувати константи.
- Розуміти принципи правильного найменування змінних.
- Працювати із типами даних.

Зміст уроку

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

Резюме

- **Змінна** – іменована частина пам'яті, яка зберігає в собі певне значення, що можна змінити. Для створення змінних використовують ключове слово **let**. Також можна використовувати ключове слово **var**, але воно застаріле, тому бажано уникати його застосування. Для того, щоби встановити значення для змінної, використовується знак рівності (присвоєння) «=».

let x = 10; – створюємо змінну з іменем 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** – стала величина. У 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 Пі.
- Мова JavaScript – реєстрозалежна мова. Message і message – дві різні змінні через те, що перший символ має різний регістр. Під час найменування варто дотримуватися таких правил:
 - Імена змінних мають містити літери, цифри або символи «\$» та «_».
 - Цифра не має бути першим символом.
 - Якщо ім'я змінної містить кілька слів, використовується **camelCasing**.

- **camelCasing** – принцип найменування, у якому перший символ назви в нижньому регістрі, а кожен перший символ слова в складовому імені розташований у верхньому регістрі. Приклади: firstName, minWidthValue, errorMessage. camelCasing заведено використовувати для більшості ідентифікаторів JavaScript.
- **PascalCasing** – принцип найменування ідентифікаторів, за якого всі слова, що формують ідентифікатор, починаються із символу у верхньому регістрі. Приклади: FirstName, MinWidthValue, ErrorMessage. PascalCasing використовують під час найменування класів і функцій конструкторів (ці теми входять до наступного курсу).
- **kebab-casing** – кожне слово в ідентифікаторі відокремлюється дефісом і пишеться в нижньому регістрі. Приклади: first-name, min-width-value, error-message. Зазвичай використовується під час написання CSS-селекторів.
- Під час найменування змінних варто вибирати описові імена, які дадуть змогу розуміти, за що відповідає значення, яке розташоване в змінній.
- **Тип даних** – допустима множина значень. Тип даних описує 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 під час присвоєння значення. Під час зміни значення змінної може змінитися і тип даних змінної. Кожна змінна має тип даних, просто ми не вказуємо цей тип явно, як це відбувається в інших мовах зі *статичною типізацією*.
- Типи даних, які наявні в JavaScript, і приклади того, як створити змінну вказаного 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 чи 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 дає змогу зберегти кілька значень в одній змінній.

- Оператор **typeof** дає змогу отримати тип даних змінної у вигляді рядкового значення. Для змінних, у яких зберігається функція, typeof поверне значення «**function**», хоча це не окремий тип даних. Змінні, у яких функція розташована як значення, є типом об'єкта. Це особливість ключового слова typeof, яка часто використовується на практиці. Функції будуть розглянуті на 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 реєстрозалежною мовою програмування?
- Якого типу даних змінна, якщо під час її ініціалізації використовувалося значення 10?
- Якого типу даних змінна, якщо під час 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?

Додаткове завдання

Створіть три змінні з іменами a, b і 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 у 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>