Звіт лабораторної роботи №2

Тема: Змінні та типи. Основи функцій.

Мета:

* Ознайомитися з поняттям змінних та їх призначенням.
* Навчитися оголошувати та ініціалізувати змінні різних типів даних.
* Вивчити основні типи даних та їх застосування в програмах.
* Ознайомитися з поняттям функції та її складовими.
* Навчитися створювати функції, передавати параметри та повертати значення.

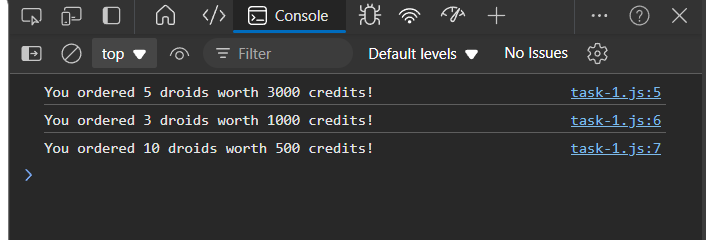
Хід роботи

Завдання 1. Замовлення дроїдів.

GitHub репозиторій - [Nouneimchik/lab2 (github.com)](https://github.com/Nouneimchik/lab2)

[lab2 (nouneimchik.github.io.)](https://nouneimchik.github.io./lab2/)

Скріни виконання:

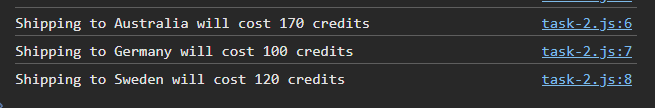


Завдання 2. Доставка товару

GitHub репозиторій - [Nouneimchik/lab2 (github.com)](https://github.com/Nouneimchik/lab2)

[lab2 (nouneimchik.github.io.)](https://nouneimchik.github.io./lab2/)

Скріни виконання:

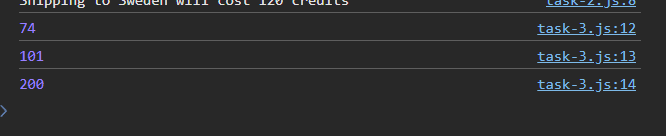


Завдання 3. Ширина елемента

GitHub репозиторій - [Nouneimchik/lab2 (github.com)](https://github.com/Nouneimchik/lab2)

[lab2 (nouneimchik.github.io.)](https://nouneimchik.github.io./lab2/)

Скріни виконання:



КОНТРОЛЬНІ ПИТАННЯ:

1. Яка різниця між оголошенням змінної з використанням var, let і const?

2. Які основні типи даних підтримуються в JavaScript?

3. Як можна перевірити тип даних змінної під час виконання програми?

4. Що таке параметри функції, і яким чином вони впливають на виконання функції?

ВІДПОВІДІ

1. Різниця між оголошенням змінної з використанням `var`, `let` і `const`:

- **var**:

- Існує в функціональному або глобальному контексті.

- Дозволяє перевизначати змінні.

- Може піддаватися "висхідному підняттю" (hoisting), тобто можна використовувати перед оголошенням.

- **let**:

- Має блочну область видимості, тобто існує тільки в межах блоку, де було оголошено.

- Дозволяє перевизначати змінні, але не може бути використаною перед оголошенням (відсутнє висхідне підняття в блочному контексті).

- **const**:

- Має таку ж блочну область видимості, як і `let`.

- Використовується для оголошення констант, тобто змінних, які не можуть бути перевизначені після ініціалізації.

- Однак, для об'єктів та масивів значення можуть змінюватися (можна змінювати вміст, але не саму змінну).

2. Основні типи даних в JavaScript:

- Примітивні типи:

- `Number`: числові значення (цілі числа та дроби).

- `String`: рядки символів.

- `Boolean`: логічні значення (true/false).

- `Null`: відсутність значення.

- `Undefined`: змінна, яка не має значення.

- `Symbol`: унікальні та незмінні значення, використовуються як ідентифікатори.

- `BigInt`: для представлення великих цілих чисел.

- Об'єкти: складні типи даних, що можуть містити ключ-значення (об'єкти, масиви, функції).

3. Для перевірки типу даних змінної можна використовувати:

- typeof`: повертає рядок, що описує тип.

console.log(typeof variable);

- Array.isArray() : перевіряє, чи є змінна масивом.

console.log(Array.isArray(variable));

- Для перевірки типу об'єкта (наприклад, для функцій, дат тощо) можна використовувати `instanceof`.

console.log(variable instanceof Array);

4. Параметри функції:

Параметри функції — це змінні, які передаються у функцію під час її виклику. Вони впливають на виконання функції, оскільки визначають, які дані функція може використовувати для обчислень чи обробки. Якщо параметри не передаються, вони отримують значення `undefined`. Приклад:

function greet(name) {

console.log(`Hello, ${name}!`);}

greet("Alice"); // Виведе: Hello, Alice!

Висновок: Під час цієї лабораторної роботи було ознайомленось з поняттям змінних та їх призначенням, навчинось оголошувати та ініціалізувати змінні різних типів даних, вивчено основні типи даних та їх застосування в програмах, ознайомленось з поняттям функції та її складовим та навчинось створювати функції, передавати параметри та повертати значення.