Практична робота №2

Тема: Динамічна типізація та механізми порівняння даних у JavaScript.

Мета: Закріпити знання про примітивні типи даних, зрозуміти механізми неявного приведення типів та навчитися коректно порівнювати значення, уникаючи логічних помилок.

КОРОТКІ ТЕОРЕТИЧНІ ВІДОМОСТІ

Неявне приведення типів (Type Coercion) у JavaScript.

Ця таблиця демонструє, як JavaScript автоматично змінює типи даних під час виконання операцій.

1. Арифметичні операції

Вираз	Результат	Пояснення
5 + "2"	"52"	Оператор + при наявності рядка виконує конкатенацію.
5 - "2"	3	Математичні оператори (крім +) переводять рядок у число.
5 * "a"	NaN	Рядок "a" неможливо перетворити на число.
true + 1	2	true перетворюється на 1, false − на 0.
null + 1	1	null стає 0.
undefined + 1	NaN	undefined неможливо перетворити на число.

2. Логічне приведення (у if або логічних операторах)

Всі значення у JavaScript є «правдивими» (true), за винятком списку «хибних» (false):

Значення	Перетворюється на...
false, 0, -0	false
"" (порожній рядок)	false
null, undefined	false
NaN	false
Усе інше (включаючи [], {}, " ")	true

3. Оператори порівняння

Для уникнення помилок з неявним приведенням завжди використовуйте суворе порівняння (===).

Порівняння	Результат	Чому так?
0 == false	true	Обидва стають числами (0).

0 === false	false	Різні типи даних (число vs булеве).
"" == 0	true	Порожній рядок стає числом 0.
null == undefined	true	Спеціальне правило мови для нестрогого порівняння.
null === undefined	false	Різні типи даних.

Якщо ви бачите дивний результат математичної операції, скористайтеся правилом: + "тяжіє" до рядків, а -, *, / — до чисел.

ЗАВДАННЯ ДЛЯ САМОСТІЙНОГО ВИКОНАННЯ

Завдання 1. Ідентифікація типів (Оператор typeof)

Створіть змінні з наступними значеннями: 42, "42", true, null, undefined, {name: "Ivan"}, Symbol("id"), 9007199254740991n.

Виведіть у консоль тип кожної змінної у форматі: Значення: [value], Тип: [type].

Завдання з підвохом: Створіть змінну let result = 10 / "apple";. Виведіть її значення та тип. Поясніть у коментарях, чому тип саме такий, хоча значення не є числом у звичному розумінні.

Завдання 2. Математична магія (Неявне приведення)

Проаналізуйте та виконайте наступні операції. Виведіть результат та пояснення (коментарем) для кожного виразу:

- console.log("10" + 5);
- console.log("10" - 5);
- console.log(true + 2);
- console.log(false + "1");
- console.log(null + 5);
- console.log(undefined + 5);
- console.log(5 + "5" - 5);

Завдання 3. Суворе vs Нестроге порівняння

Виконайте порівняння та поясніть отримані true або false:

- Порівняйте число 0 з порожнім рядком "", використовуючи == та ===.
- Порівняйте false з порожнім рядком "" через ==.
- Порівняйте null та undefined через == та ===.
- Виконайте порівняння: 5 != "5" та 5 !== "5".

Висновок: Сформулюйте правило, коли варто використовувати ==, а коли ===.

Завдання 4. Логічне приведення (True vs False)

Використовуючи функцію Boolean(), перевірте, як JavaScript перетворює наступні значення на логічний тип:

0, "0", " ", [], {}, NaN, null.

Побудуйте умовну конструкцію if, яка перевіряє порожній масив []. Якщо умова виконується, виведіть: "Масив — це true". Поясніть результат.

Завдання 5. Комплексна вправа "Валідатор даних"

Напишіть невеликий сценарій для реєстрації користувача:

- Запитайте через prompt вік користувача.
- Перетворіть отримане значення на число (явне приведення).
- Перевірте, чи не є результат NaN (використайте isNaN()).
- Якщо введено коректне число, порівняйте його: якщо вік >= 18 — виведіть "Доступ дозволено", інакше − "Доступ заборонено".
- Якщо введено не число − виведіть помилку: "Будь ласка, введіть числове значення".