

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
КИЇВСЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ ІМЕНІ ТАРАСА
ШЕВЧЕНКА

ФАКУЛЬТЕТ ІНФОРМАЦІЙНИХ ТЕХНОЛОГІЙ

Кафедра інформаційних систем та технологій

Лабораторна робота № 2

з дисципліни «Технології програмування»

Тема: «JavaScript. Data types, typeof, type casting, arrays, date, object»

Виконав студент 2-го курсу

Микитенко Ілля Андрійович

Група: ІР-21

Варіант 11

Перевірила:

Бондаренко Ольга Сергіївна

Завдання 1

Вивести поточний час у форматі HH:MM:SS.

Код програми:

```
JS laba 2.1.js
1  const time = new Date();
2  console.log("Поточний час:", time.toLocaleTimeString());
```

Приклад виконання:

```
C:\Program Files\nodejs\node.exe --experimental-network-inspection .\laba 2.1.js
Поточний час: 13:57:04
```

Завдання 2

Згенерувати масив із 10 випадкових чисел.

Відсортувати масив у порядку спадання.

Код програми:

```
1  let sacrificedarr = [];
2  for (let i = 0; i < 100; i++) {
3    sacrificedarr.push(i);
4  }
5
6  for (let i = sacrificedarr.length - 1; i > 0; i--) {
7    let j = Math.floor(Math.random() * (i + 1));
8    [sacrificedarr[i], sacrificedarr[j]] = [sacrificedarr[j], sacrificedarr[i]];
9  }
10
11 let arr = sacrificedarr.slice(0, 10);
12
13 console.log("Массив: ", arr.join(", "));
14
15 let arrsorted = [];
16 arrsorted = arr.sort((a, b) => b - a);
17 console.log("Відсортований массив: ", arrsorted.join(", "));
```

Приклад виконання:

```
C:\Program Files\nodejs\node.exe --experimental-network-inspection .\laba 2.2.js
Массив: 99, 78, 96, 7, 39, 59, 36, 85, 25, 86
Відсортований массив: 99, 96, 86, 85, 78, 59, 39, 36, 25, 7
```

Контрольні питання:

1. Які типи даних існують у JavaScript і які з них належать до примітивних?

- Примітивні: string, number, boolean, null, undefined, bigint, symbol.
- Непримітивні (reference types): object (включає масиви, функції, дати тощо).

2. Чим відрізняється значення null від undefined у плані використання та перевірки?

- null - це "свідомо відсутнє значення", встановлене явно.
- undefined - це "значення не визначене", за замовчуванням для змінних, які ще не мають присвоєного значення.
- При перевірці:
 - null == undefined // true (нестроге порівняння)
 - null === undefined // false (строге порівняння)

3. Який результат дасть вираз: typeof NaN? Чому саме такий?

- Результат: "number".
- Пояснення: NaN ("Not-a-Number") є спеціальним значенням типу number, яке означає невдалу математичну операцію.

4. Яке значення поверне Number(" ") і чому?

- Результат: 0.
- Пояснення: рядок із пробілами приводиться до порожнього рядка "", а Number("") = 0.

5. Який результат дає Boolean("0") і чим він відрізняється від Boolean(0)?

- Boolean("0") → true, бо непорожній рядок завжди truthy.
- Boolean(0) → false, бо 0 є falsy.

6. У чому полягає різниця між масивом і об'єктом у JavaScript, якщо обидва є об'єктами?

- Масив - спеціалізований об'єкт з індексованими ключами (0, 1, 2...) та властивістю `length`. Використовується для зберігання послідовностей.
- Об'єкт - структура з довільними ключами (рядки або символи). Використовується для опису сутностей і властивостей.

7. Який результат виконання коду? Поясніть, у чому різниця між `slice` і `splice`.

```
let arr = [1, 2, 3];
```

```
console.log(arr.slice(1, 2)); // [2]
```

```
console.log(arr.splice(1, 2)); // [2, 3]
```

```
console.log(arr); // [1]
```

- `slice(start, end)` → створює копію вибраного підмасиву, **не змінюючи оригінал**.
- `splice(start, deleteCount)` → **видаляє елементи з оригіналу** та повертає видалені.

8. Як за допомогою `Math.random()` отримати випадкове ціле число в діапазоні від -50 до 50 включно?

```
let rand = Math.floor(Math.random() * 101) - 50;
```

- `Math.random() * 101` → число від 0 до 100.
- `-50` → зсув у діапазон `[-50, 50]`.

9. Який результат обчислень наступного коду? Чому саме так?

```
let d1 = new Date("2025-01-01");
```

```
let d2 = new Date("2025-01-02");
```

```
console.log((d2 - d1) / (1000 * 60 * 60 * 24));
```

- Результат: 1.
- Пояснення: різниця між датами - мілісекунди. Ділення на `(1000*60*60*24)` переводить у дні.

10. Який результат дасть наступний код? Чому змінилось значення в обох змінних?

```
let obj = { a: 1 };
```

```
let copy = obj;
```

```
copy.a = 5;
```

```
console.log(obj.a); // 5
```

- Результат: 5.
- Пояснення: obj і copy посилаються на **один і той самий об'єкт** у пам'яті, тому зміна через одну змінну впливає на іншу.