

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

Виконала:
студентка гр. ІР-21
Мар'яна Сідоріна.
Перевірила:
Ольга Бондаренко
Зараховано від

____.____._____

(підпис викладача)

Код до завдання 1:

```
let date = new Date();
let futureDate=new Date(date)

futureDate.setDate(futureDate.getDate()+100);

console.log(date.toLocaleString('uk-UA'));
console.log(futureDate.toLocaleString('uk-UA'));
```

Код до завдання 2:

```
let arr = [];

for (let i=0; i<10; i++) {
    arr.push(Math.floor(Math.random() * 100));
}
console.log(arr);

let min = Math.min(...arr);
console.log(min);

let index = arr.indexOf(min);
console.log(index);

arr.splice(index, 1);
console.log(arr);
```

Контрольні запитання:

1. Типи даних у JavaScript і примітивні

У JS є 7 основних типів даних:

- **Примітивні типи:**
 1. string — рядки
 2. number — числа (цілі та дробові, включно з NaN та Infinity)
 3. bigint — великі цілі числа
 4. boolean — логічні значення true/false
 5. undefined — відсутність значення
 6. symbol — унікальні ідентифікатори
 7. null — спеціальне значення «немає об'єкта»
- **Об'єктні типи:**

- object (масиви, функції, дати, звичайні об'єкти тощо)

2. Різниця між null і undefined

- undefined — змінна існує, але їй **не присвоєно значення**.
- null — змінній **навмисно присвоєне порожнє значення**.

Приклади перевірки:

```
let a;  
console.log(a === undefined); // true  
let b = null;  
console.log(b === null); // true
```

3. Результат typeof NaN

```
typeof NaN // "number"
```

- Хоч NaN означає «Not a Number», у JS він **належить до типу number**, тому typeof повертає "number".

4. Результат Number(" ")

```
Number(" ") // 0
```

- Порожній рядок або рядок з пробілами при конверсії у число перетворюється на 0.

5. Різниця Boolean("0") і Boolean(0)

```
Boolean("0") // true  
Boolean(0) // false
```

- Будь-який **непорожній рядок** — істинне значення (true).
- Число 0 — хибне значення (false).

6. Різниця між масивом і об'єктом

- Масив (Array) — це об'єкт із **порядкованими числовими індексами** і спеціальними методами (push, pop, slice, splice тощо).
- Об'єкт (Object) — **непорядкована колекція ключ-значення**, ключі можуть бути рядками або символами.

7. Різниця між slice і splice

```
let arr = [1, 2, 3];  
console.log(arr.slice(1, 2)); // [2] — повертає новий масив, **не змінює arr**
```

```
console.log(arr.splice(1, 2)); // [2, 3] — видаляє елементи з arr і повертає їх  
console.log(arr);           // [1] — масив після splice
```

- `slice(start, end)` — повертає **копію частини масиву**, не змінює оригінал.
- `splice(start, deleteCount, ...items)` — змінює **оригінальний масив**, видаляючи або додаючи елементи.

8. Випадкове ціле число від -50 до 50 включно

```
let randInt = Math.floor(Math.random() * 101) - 50;  
console.log(randInt);
```

- `Math.random()` → $[0, 1)$
- `* 101` → $[0, 101)$
- `Math.floor()` → $[0, 100]$
- `- 50` → $[-50, 50]$

9. Різниця дат у днях

```
let d1 = new Date("2025-01-01");  
let d2 = new Date("2025-01-02");  
console.log((d2 - d1) / (1000 * 60 * 60 * 24)); // 1
```

- `d2 - d1` дає різницю в **мілісекундах**.
- Ділення на `1000*60*60*24` конвертує мілісекунди у **дні**.

10. Зміни значень об'єктів при присвоєнні

```
let obj = { a: 1 };  
let copy = obj;  
copy.a = 5;  
console.log(obj.a); // 5
```

- `copy` та `obj` посилаються на **той самий об'єкт у пам'яті**.
- Зміни через `copy` впливають на `obj`.