Лабораторна робота №2

Тема: Особливості виконання операцій та операторів у мові JavaScript. Типи даних. Приведення типів. Регулярні вирази

Мета роботи: вивчити відмінності у роботі операцій та операторів у мові JavaScript від інших мов; ознайомитись з базовими типами даних; вивчити особливості приведення типів.

Завдання на лабораторну роботу

1. Ретельно вивчити теорію:

- https://uk.javascript.info/types-intro

https://uk.javascript.info/operators

- https://uk.javascript.info/operators
- https://uk.javascript.info/comparison
- https://uk.javascript.info/bitwise-operators
- https://uk.javascript.info/uibasic
- https://uk.javascript.info/ifelse
- https://uk.javascript.info/logical-ops
- https://uk.javascript.info/types-conversion
- https://uk.javascript.info/while-for
- https://uk.javascript.info/switch
- https://learn.javascript.ru/regular-expressions

2. Для самоперевірки, виконайте завдання:

2.1. Визначте, які значення будуть виведені на екран (файл *task2.1.js*):

```
let a = 2;
let b = 3;
let c, d;
c = ++a;
d = b++;
c = (2+ ++a);
d = (2+ b++);

console.log(a);
console.log(b);
console.log(c);
console.log(d);
```

2.2. Визначте, яке значення *x* буде отримано в результаті виконання коду (файл *task2.2.js*):

```
let a = 5;

let x = a *= (a *= 2);

console.log(x)
```

2.3. Які значення будуть виведені наступним кодом і чому (файл *task2.3.js*)?

```
console.log(console.log("Hello, world"))

console.log(null || undefined)

console.log(null || 2 || undefined)

console.log(1 && null && 2)

console.log(1 && console.log("Hello"))
```

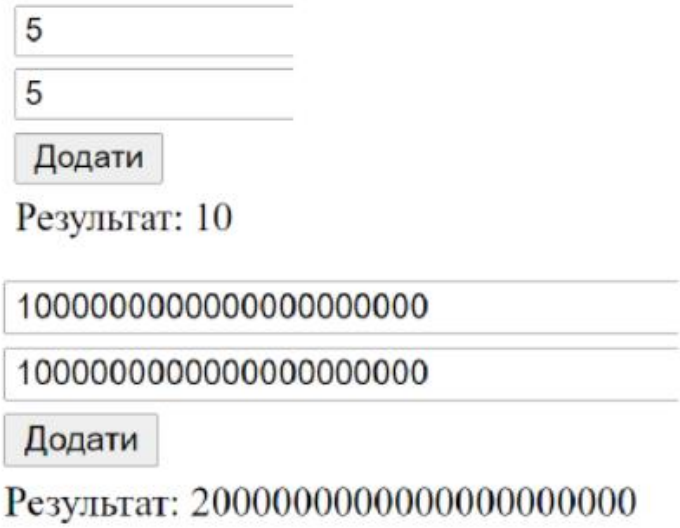
2.4. Пояснити які значення і яких типів будуть давати наступні вирази (файл *task2.4.js*):

```
//Розібрати на занятті

console.log("" + 1 + 0);

console.log("" - 1 + 0);

console.log(true + false);

console.log(6 / "3");

console.log("2" * "3");

console.log(4 + 5 + "px");

console.log("$" + 4 + 5);

console.log("4" - 2);

console.log("4px" - 2);

console.log(7 / 0);

console.log("  -9\n" + 5);

console.log("  -9\n" - 5);

console.log(5 && 2);

console.log(2 && 5);

console.log(5 || 0);

console.log(0 || 5);

console.log(null + 1);
```

```
console.log(undefined + 1);

console.log(null == "\n0\n");

console.log(+null == +"\n0\n");

//Для самостійного розбору:

console.log("5" + "1");

console.log("5" + 1);

console.log(5 + 1);

console.log("5" + true);

console.log(1 / 0);

console.log(-5 / 0);

console.log(1 / Infinity);

console.log(Infinity - Infinity);

console.log(Infinity + 10);

console.log(Infinity + Infinity);

console.log(Infinity + "3");

console.log(undefined + 5);

console.log(parseInt("10"));

console.log(parseInt("10.3"));

console.log(parseInt(10.3));

console.log(parseInt("34 38 23"));

console.log(parseInt("   60   "));

console.log(parseInt("x10"));

console.log(parseInt("0x10"));

console.log(parseInt("010"));
```

```
console.log(parseInt("24", 8));

console.log(parseInt("FF", 16));

console.log(parseFloat("0xFF"));

console.log(false && 0);

console.log(0 && false);

console.log(!0);

console.log(!5);

console.log(1 && 0);

console.log(1 && false);

console.log(1 || 5);

console.log(0 || 5);

console.log(5 + 4 + "!");

console.log(!null);

console.log(!!null);

console.log(!!5);

console.log(null == 0);

console.log(null === 0);

console.log(null > -5);

console.log(undefined > 5);

console.log(undefined > 5 != true);

console.log(a = null + 5);
```

3. Петрик вирішив написати свій перший калькулятор, який додає два цілих числа. Допоможіть йому це зробити. Калькулятор повинен додавати як завгодно великі числа (файл *task3.html*). Вивід результату здійснюється в десятковому представленні. *Приклад* тестів для перевірки:



4. Василькові потрібно написати функцію toCamelCase(), яка приймає значення імені змінної в стилі “Snake case”, та повертає це ж ім’я в стилі «Lower camel case» (файл *task4.js*). Використати **регулярні вирази**. Для прикладу:

Значення, що передається в функцію	Результат роботи функції
my_variable	myVariable
new_brand_product	newBrandProduct

5. Оленці в свою чергу потрібно написати функцію toSnakeCase(), яка працює як обернена до функції toCamelCase(). Допоможіть їй це зробити (файл *task5.js*). Використати **регулярні вирази**.

6. Андрійко повинен в тексті замінити всі дати формату yyyy/mm/dd на формат dd.mm.yyyy. Текст може бути як завгодно великий і одна з вимог - використати пошук і заміну з використанням **регулярних виразів** (файл *task6.html*).

7. Маючи дані, Миколка повинен згенерувати html-таблицю з них з такими полями FirstName, LastName, Gender, Address, BirthDate, Nickname, Password, Email (файл *task7.html*). Використати **регулярні вирази**.

Дані вводяться в поле <textarea>, після натискання кнопки, на основі цих даних генерується таблиця. Посилання для даних:
https://drive.google.com/file/d/1tZhN6n8Aseqosje528EG15OXWQsuFWrK/view?usp=sharing

8. Закомітити виконані завдання.