

# Типи даних

// Прості типи

// Числа

// Рядки

// Булеві змінні

// null

// undefined

// Об'єктні типи

// Об'єкти

// Масиви

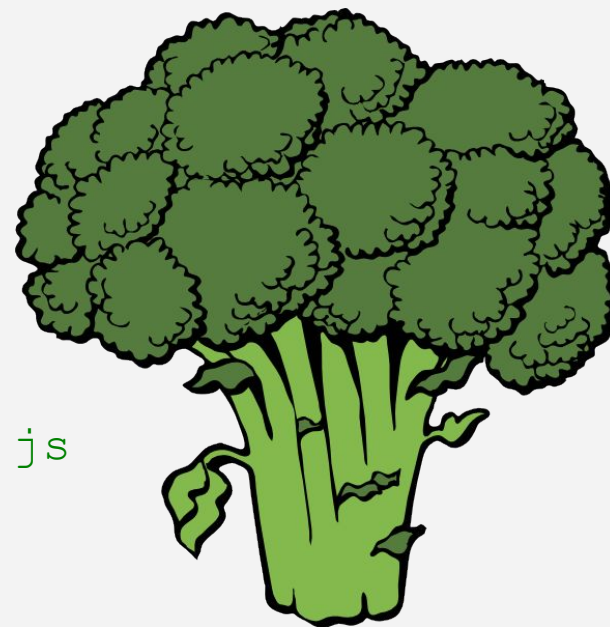
// Функції

// Регулярні вирази



# Оператор typeof

```
typeof 0 // "number"  
typeof "Селера" // "string"  
typeof true // "boolean"  
typeof null // "object" // помилка js  
typeof undefined // "undefined"  
typeof {} // "object"  
typeof [] // "object"  
typeof function(){} // "function"
```



Результатом **typeof** є рядок що містить тип.

# Числа

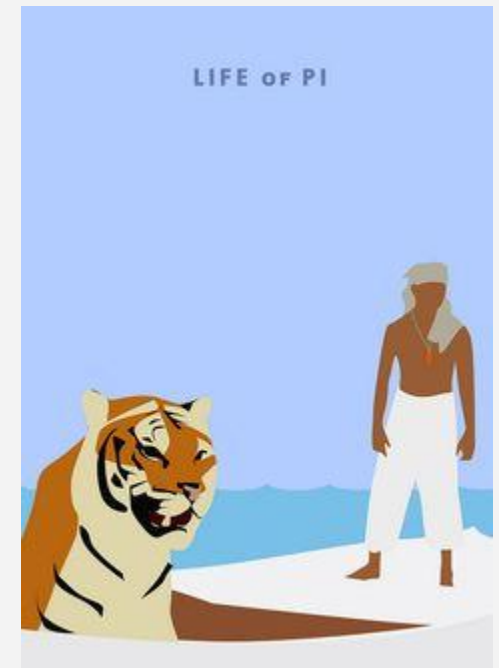
В JavaScript тип даних number використовується для цілих і дробових чисел

```
const age = 25;
```

```
const PI = 3.14;
```

Також для представлення чисел в 16 і 8 системах

```
const color = 0xffffffff;
```



# Унарні оператори

В JavaScript є два **унарні** оператори

```
console.log(+10);
```

```
console.log(-5);
```

Також є два оператори, які називаються  
оператори **інкременту** і **декременту**

```
console.log(i++);
```

```
console.log(i--);
```



# Префіксний і постфіксний інкремент

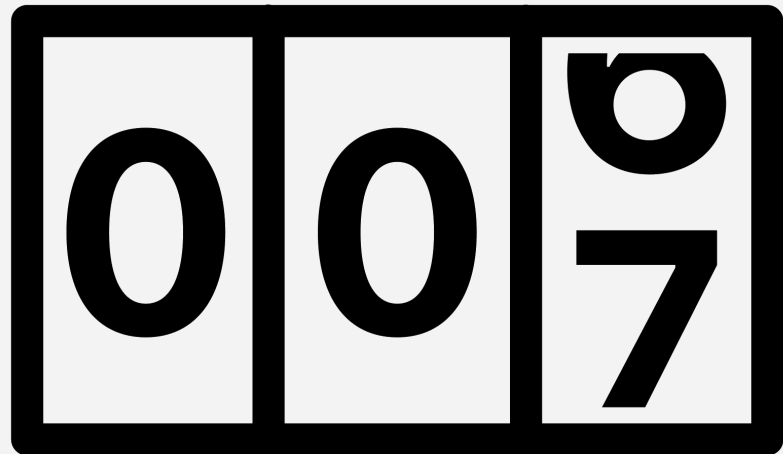
```
let i = 0;
```

```
console.log(++i); // 1
```

```
console.log(i); // 1
```

```
console.log(i++); // 1
```

```
console.log(i); // 2
```



# Бінарні оператори

```
console.log(5 + 5); // 10
```

```
console.log(8 - 5); // 3
```

```
console.log(8 * 2); // 16
```

```
console.log(9 / 3); // 3
```

```
console.log(10 % 3); // 1 остача від ділення
```



# Присвоювання з операцією

Якщо в нас є змінна з певним значенням, і нам потрібно додати до неї інше значення ми використовуємо такий запис:

```
let n = 100;  
n = n + 100;  
console.log(n); // 200
```

Для таких речей в JS є спеціальні оператори:

```
n += 100; // 200  
n *= 2; // 400  
n /= 4; // 100  
n -= 50; // 50
```

# Оператори порівняння

```
console.log(5 > 2); // більше
console.log(5 < 10); // менше
console.log(10 >= 10); // більше рівне
console.log(7 <= 10); // менше рівне

console.log(10 === 10); // рівне
console.log(10 !== 10); // не рівне

console.log(10 == '10'); // true
console.log(10 === '10'); // false
```



# Об'єкт Math

`Math.sqrt(25);` // 5 - квадратний корінь

`Math.pow(2,3);` // 8 - 2 в степені 3

`Math.PI;` // математична константа 3.14

[Весь список методів об'єкту Math](#)

# Infinity i Nan(not a number)

```
console.log(10 / 0); // Infinity
console.log(0 / 0); // Nan
console.log(Infinity / Infinity); // Nan
console.log(Math.sqrt(-25)); // Nan
```

# Рядки

```
const someString = 'Hello';
const myAwesomeString = 'I am JavaScript';

console.log(someString + myAwesomeString);
// HelloI am JavaScript;

console.log(someString + ', ' + myAwesomeString); // ES5
// Hello, I am JavaScript;

console.log(`${someString}, ${myAwesomeString}`); // ES6
// Hello, I am JavaScript;
```

# Спеціальні символи

```
const myAwesomeString = 'I'm JavaScript';
```

За допомогою зворотнього слеша (\) можна **екранувати** спеціальні символи

```
const myAwesomeString = 'I\'m JavaScript'; // ES5
```

```
const myAwesomeString = `I'm JavaScript`; // ES6
```

До спеціальні символи також належать

**\n** - перенос каретки

**\t** - табуляція

# Методи і властивості рядків



```
const cactus = 'Кактус';
```

```
console.log(cactus.length); // довжина рядка - 6
```

```
console.log(cactus.charAt(0)); // доступ до символу 'К'
```

```
console.log(cactus[0]); // доступ до символу 'К'
```

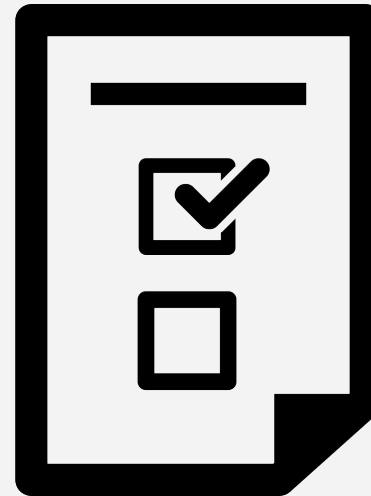
```
console.log(cactus.toUpperCase()); // верхній регістр 'КАКТУС'
```

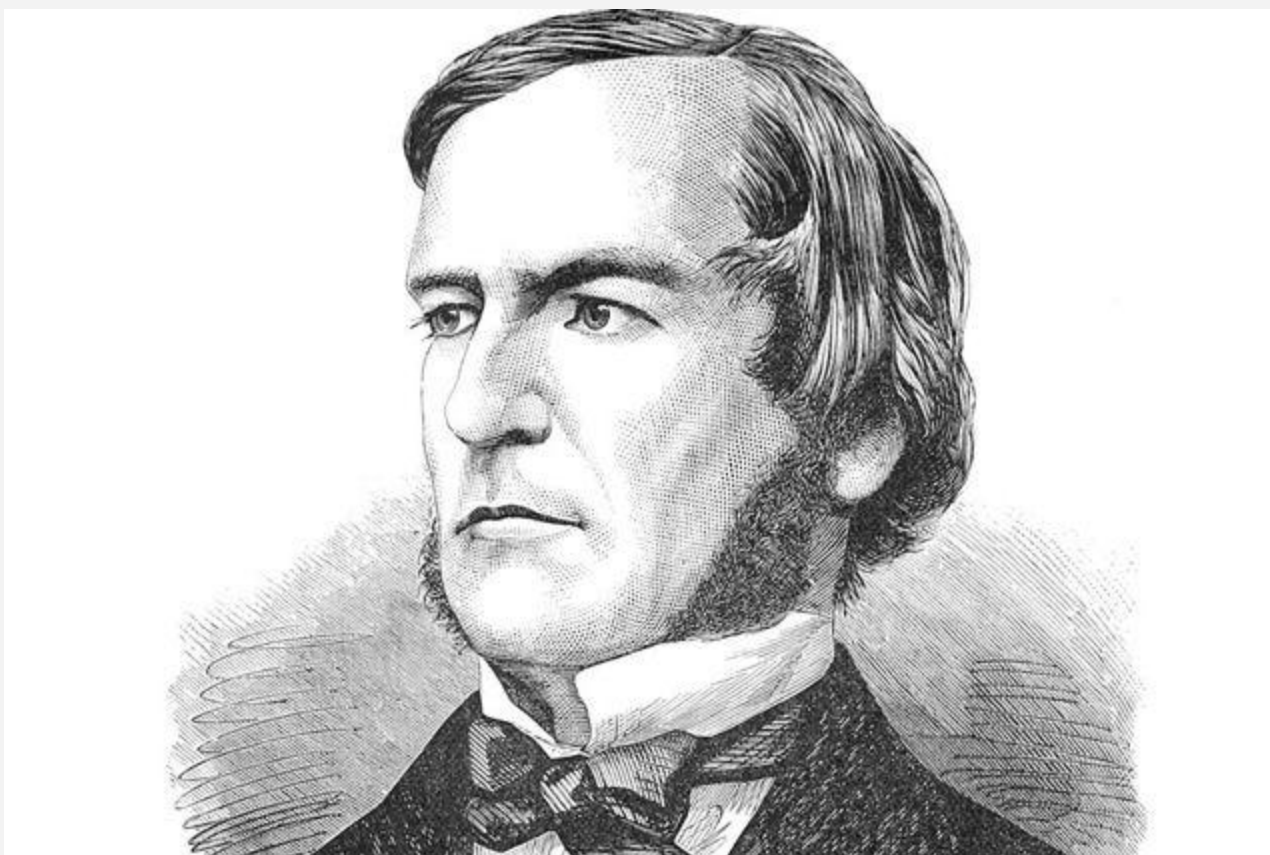
```
console.log(cactus.toLowerCase()); // нижній регістр 'кактус'
```

# Булевий (логічний) тип

У нього всього два значення: **true** (істина) і **false** (брехня)

```
const firstItem = true;  
const secondItem = false;
```





Джордж Буль

# null i undefined

**null** - спеціальне значення, яке означає **нічого, немає, невідомо**

```
const meaningOfLife = null;
```

**undefined** буквально означає що значення не присвоєно

```
let color;
```

```
console.log(color); // undefined
```



# Приведення (переведення) типів

Приведення до рядка

```
const number = 5;
```

```
console.log("" + number); // '5'
```

```
console.log(String(number)); // '5'
```

```
const boolean = false;
```

```
console.log("" + boolean); // 'false'
```

```
console.log(String(boolean)); // 'false'
```

# Приведення до чисел

```
const string = '505';
```

```
console.log(+string); // 505
```

```
console.log(Number(string)); // 505
```

```
const boolean = false;
```

```
console.log(+boolean); // 0
```

```
console.log(Number(boolean)); // 0
```

# Приведення до логічних значень

Приведення до логічних значень відбувається в якомусь логічному контексті, тобто якась перевірка, наприклад `if(value)`

```
const checked = 'true';  
if(checked) { // true  
  console.log(checked);  
}  
console.log(Boolean(checked)); // true
```

undefined, null	false
Числа	Всі true, крім 0, NaN
Рядки	Всі true, крім пуского рядка



# Умовні оператори, If

```
const age = 20;  
  
if (age > 18) {  
    watchMovieWithRatingR();  
}
```

Оператор if, отримує умову, приводить її до логічного значення, якщо воно - true, виконує команди у фігурних дужках

# Хибна умова, Else

```
const age = 14;  
  
if (age > 18) {  
    watchMovieWithRatingR();  
} else {  
    watchCartoon();  
}
```



Якщо умова не виконується, виконуються команди у фігурних дужках після оператора **else**

# Декілька умов, Else if

```
const age = 8;

if (age > 18) {
    watchMovieWithRatingR();
} else if (age > 10 && age < 18) {
    watchCartoon();
} else {
    watchMovieWithParents();
}
```

Спочатку перевіряється перша умова, потім наступна і так далі, аж до оператора **else**

# Умовна конструкція switch

```
const experience = 3;
```

```
switch (experience) {  
  case 1: console.log( 'junior' ); break;  
  case 3: console.log( 'middle' ); break;  
  case 5: console.log( 'senior' ); break;  
  default: console.log( 'trainee or architect' );  
}
```



# Логічні оператори

	Логічне АБО
&&	Логічне І
!	Логічне НЕ

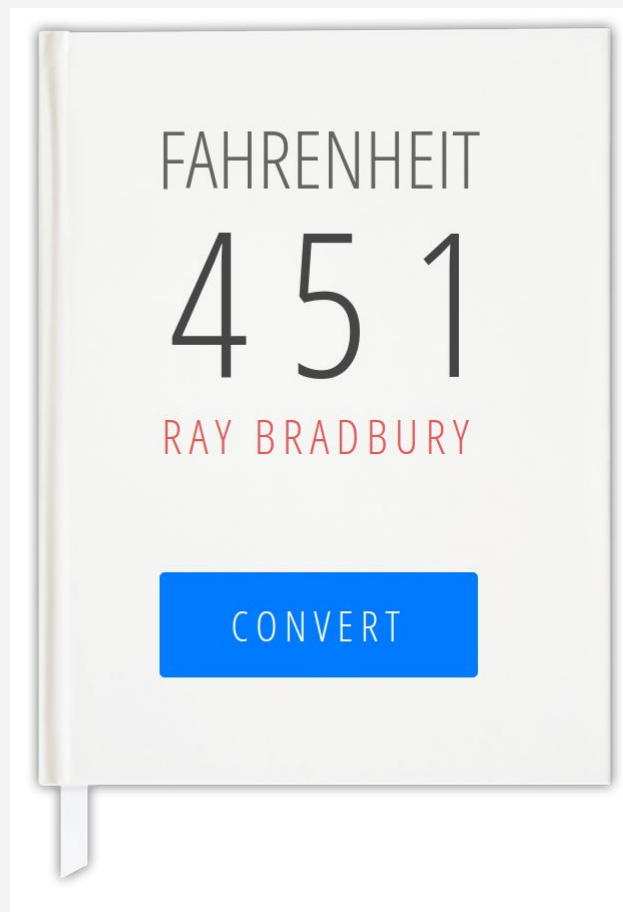
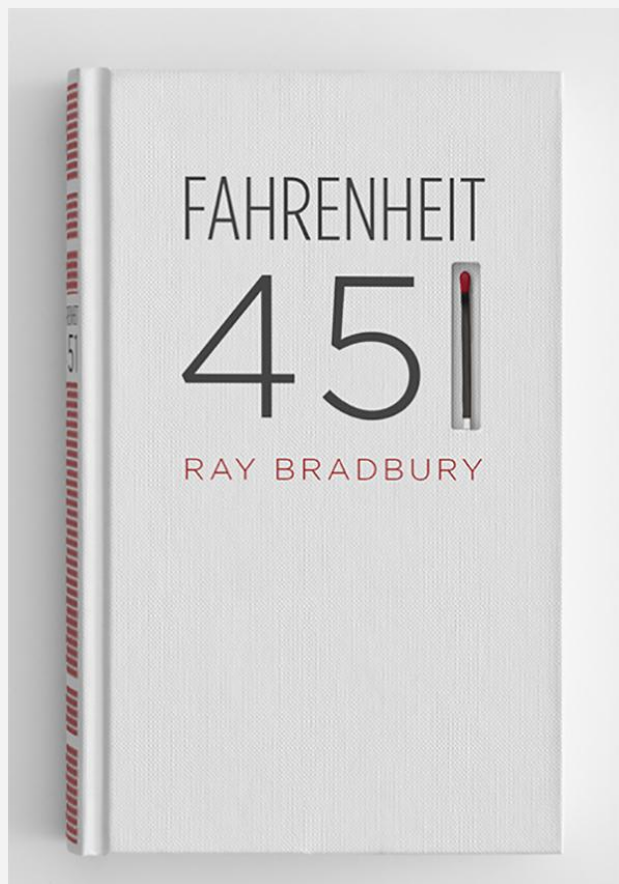
Логічні оператори використовуються для порівняння логічних значень

```
if (true || false) // true
```

```
if (true && false) // false
```

```
if (!false) {} // true
```





# Практичні завдання

- Реалізувати Конвертер для переведення градусів по фаренгейту в градуси цельсія і навпаки.
- Знайти площу кола за його радіусом

# Необхідно запам'ятати

- Оператор typeof, унарні, бінарні оператори, інкремент, декремент
- Оператори порівняння
- Рядки, методи рядків, логічний тип даних, null і undefined
- Приведення типів
- Умовні оператори
- Логічні оператори

Рекомендовано прочитати

[JavaScript - David Flanagan: розділ 2, розділ 3 \(пункти 3.1 - 3.3\).](#)

# Завдання для закріплення

- Створити константу з Вашим ім'ям, прізвищем, по-батькові, вивести в консоль довжину константи та аббревіатуру. // можна робити без UI
- Реалізувати ділення двох чисел ( $a/b$ ), якщо числа діляться націло вивести результат, інакше вивести повідомлення (а не можна поділити на b) // можна робити без UI
- Задано двохзначне число. Визначити, чи його цифри однакові. // з UI
- Написати програму, яка на введення розділового знаку виводить на екран його назву. Наприклад, на введення точки виводиться повідомлення: "Це точка". // з UI (розділові знаки .,?!:;- бажано використати switch)
- Додати до програми, яка рахує площу трикутника UI частину (поля для вводу висоти та сторони і кнопку "Визначити"). Також додати валідацію.