

# Вступ



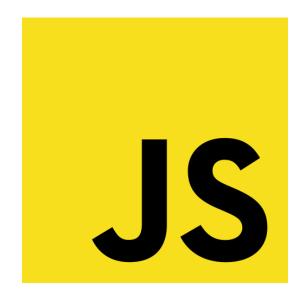
### План

- Що таке JS
- Для чого призначений
- Зовнішні скріпти
- Hello world
- Змінні
- Порядок виконання
- Типи даних
- Оператори

### Що таке JavaScript

• мова програмування з динамічною типізацією

https://uk.wikipedia.org/wiki/JavaScript



### Для чого призначена

- для створення сценаріїв веб-сторінок, що надає можливість взаємодіяти з користувачем
- асинхронно обмінюватися даними з сервером
- змінювати структуру та зовнішній вигляд веб-сторінки
- створення односторінкових веб-застосунків (ReactJS, AngularJS, Vue.js)
- програмування на стороні сервера (Node.js)
- стаціонарних застосунків (<u>Electron</u>, <u>NW.js</u>)
- мобільних застосунків (React Native, Cordova)

### Підключення скріптів на сторінці

• за допомогою тегу <script>...</script>

```
<script>
       alert('Hello world');
</script>
<script src="script.js"></script>
<script src="script.js">
       alert ('Hello world'); //якщо є атрибут src то цей скріпт
ігнорується
</script>
```

### Порядок підключення

- якщо браузер зустрічає тег <script> то він має його виконати і тоді йти далі
- поки браузер не виконає його, він не покаже решту сторінки, тому скріпти підключають перед закриваючтим </body>
- **атрибут** <script async src="..."> не зупиняє відображення сторінки
- атрибут <script defer src="..."> працює таксамо як і async але зберігається порядок виконання скріптів і виконається коли весь документ буде оброблений браузером

### Структура коду

• кожна команда переважно пишеться в новому рядку

```
1 alert('Привет');
2 alert('Мир');
```

 після кожної команди, рекомендується, ставити крапку з комою, але можна і упустити, про те не у всіх випадках

```
1 alert("Сейчас будет ошибка")
2 [1, 2].forEach(alert)
```

### Коментарі

```
// коментар
alert('Hello'); //коментар
/* і я коментар
i складаюсь з двох рядків */
```

### Змінні

- контейнер(комірка пам'яті) для зберігання значень даних
- для оголошення використовується ключове слово let та ім'я змінної
- в старих скриптах можна знайти оголошення через var майже теж саме що i let

```
let message = "Hello";
```



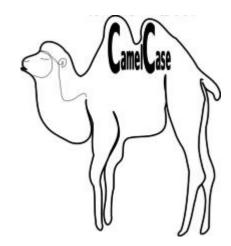
### Іменування змінних

- ім'я змінної може містити букви, числа і символи \$ і \_\_
- перший символ не може бути число
- не можна називати змінні ключовими словами, let for = 5;

https://learn.javascript.ru/variables

### Підходи до називання змінних

- first name
- FirstName
- firstName



https://learn.javascript.ru/variables#pridumyvayte-pravilnye-imena

### Константи

Змінні, оголошені за допомогою const, називаються «константами». Їх не можна змінити. Спроба зробити це призведе до помилки:

```
const PI = 3.14;
PI = 5; // помилка
```

#### Синтаксис

```
let a; - оголошення змінної
let b, c; - оголошення змінних
a = 1; b = 2; c = 3; - ініціалізація змінних
let d = 1; - оголошення і ініціалізація змінної
```

'Text' - строка задається через одинарні "Text" - або подвійні лапки

### Вивід в консоль

• console.log(a); виводить значення в консоль

### Типи даних

- number
- string
- boolean (логічний тип)
- null
- undefined
- object (об'єкт)
- symbol (символ)

### Number

```
let n = 123;
n = 12.345;
```

Infinity є математичною нескінченність ∞ і більше будь-якого числа alert(1 / 0); // Infinity

NaN означає обчислювальну помилку. Це результат неправильної або невизначеної математичної операції

alert("не число"/2); // NaN, не можливо виконати обчислення

# String

#### Варіанти задання строки

```
let str = "Привіт";
let str2 = 'Одинарні лапки';
let phrase = `Зворотні лапки`;
```

### Boolean

```
приймає два значення true(істина) і false(брехня)
let isMobile = true;
```

#### null

окремий тип, який містить тільки значення null:

```
let age = null;
```

спеціальне значення, яке представляє собою «нічого», «порожньо» або «значення невідомо»

### undefined

означає, що «значення не було присвоєно».

Якщо змінна оголошена, але їй не присвоєно ніякого значення, то її значенням буде undefined:

```
let x;
alert(x); // "undefined"
```

### Object

Згадані вище типи називаються «примітивними», тому що їх значеннями можуть бути тільки прості значення (будь то рядок або число, або щось ще). Об'єкти ж використовуються для зберігання колекцій даних або більш складних об'єктів.

# Symbol

використовується для створення унікальних ідентифікаторів об'єктів

### Оператор typeof

```
typeof undefined // "undefined"
typeof 0 // "number"
typeof true // "boolean"
typeof "foo" // "string"
typeof Symbol("id") // "symbol"
typeof Math // "object" (1)
typeof null // "object" (2)
typeof alert // "function" (3)
```

https://learn.javascript.ru/types

### Приведення типів

**Строкове перетворення** відбувається, коли потрібне представлення чогонебудь у вигляді рядка.

Наприклад, alert (value) перетворює значення до рядка.

```
let value = true;
alert(typeof value); // boolean
value = String(value); // τεπερ value - строка "true"
alert(typeof value); // string
```

#### Чисельне перетворення відбувається в математичних функціях і виразах.

```
alert( "6" / "2" ); // 3, Строки преобразуются в числа
```

#### явне приведення до числа

```
let str = "123";
alert(typeof str); // string
let num = Number(str); // число 123
alert(typeof num); // number
```

Якщо рядок не може бути явно приведена до числа, то результатом перетворення буде NaN. наприклад:

```
let age = Number("Строка");
alert(age); // NaN, привести не вийшло
```

Якщо при математичні операції один із доданків є строкою то результат теж строка. Така операція називається конкатинація

```
alert(1 + '2'); // '12'
alert('1' + 2); // '12'
```

### Логічне приведення

Відбувається в логічних операторах, але також явно за допомогою функції Boolean (value).

Значення, які інтуїтивно «порожні», на кшталт **0**, **порожнього рядка**, **null**, **undefined** і **NaN**, стають **false**. Всі інші значення стають **true**.

```
alert(Boolean(1)); // true
alert(Boolean(0)); // false
alert(Boolean("CTpoka!")); // true
alert(Boolean("")); // false
```

### Математичні оператори

```
+, -, *, /
% - остача від ділення
++ - інкремент(збільшує значення на 1)
-- дикримент(зменшує значення на 1)
** зведення в степінь
```

• існує дві форми префіксна і суфіксна ++а і а--

https://www.w3schools.com/js/js\_operators.asp https://goo.gl/6hkSdr

### Оператори присвоєння

Оператор	Приклад	Еквівалент
=	x = y	x = y
+=	x += y	x = x + y
-=	x -= y	x = x - y
*=	x *= y	x = x * y

```
txt1 = "John";
txt2 = "Doe";
txt3 = txt1 + " " + txt2; //John Doe
```

### Оператори порівняння

==	рівність	
===	строга рівність	
!=	нерівність	
!==	строга нерівність	
>	більше	
<	менше	
>=	більше рівне	
<=	менше рівне	
?:	тернарний оператор	

https://learn.javascript.ru/comparison

Оператори порівняння повертають значення логічного типу **true** або **false**.

```
alert( 2 > 1 ); // true
alert( 2 == 1 ); // false
alert( 2 != 1 ); // true
```

Строки порівнюються посимвольно і використовуючи Unicode символа

```
alert( 'A' > 'A' ); // true
alert( 'Z' > 'a' ); // false
```

При порівнянні значень різних типів JavaScript призводить кожне з них до числа.

```
alert('2' > 1); // true, строка '2' стає числом 2

alert('01' == 1); // true, строка '01' стає числом 1

alert(true == 1); // true

alert(false == 0); // true
```

### Строге порівняння

```
alert( 0 == false ); // true

alert( '' == false ); // true

Це відбувається через те, що операнди різних типів перетворюються оператором == до числа. У підсумку, і порожній рядок, і false стають нулем.
```

```
alert( 0 === false ); // false, порівнюються різні типи
```

### Тернарний оператор

умова? вираз1: вираз2

Якщо умова вірна(true) то повернеться вираз1, хибна(false) вираз2

```
let status = 2 > 1 ? 'ok' : 'not';
```

### Практика

```
a = 6; b = 3; a > b
a = 3; b = 3; a == b
a = 3: b = 6: a != b
a = 1: b="": a === b
'B' > 'A', 'a' > 'Z'
a = false; b = 0; a >= b
('2' === 2) ? 'ok' : 'not'
```

### Взаємодія з користувачем

- alert виводить повідомлення: alert (message);
- prompt виводить модальне вікно зі заголовком title, полем для введення тексту, заповненим рядком за замовчуванням default, введенне значення записується в змінну result: result = prompt(title, default); при кліку cancel повертається null
- confirm виводить вікно з питанням question з двома кнопками: OK i CANCEL: result = confirm(question);

https://learn.javascript.ru/uibasic

#### Посилання

https://learn.javascript.ru/

https://www.w3schools.com/js/default.asp

https://developer.mozilla.org/ru/docs/Web/JavaScript/Guide/%D0%92%D0%B2%D0%B5%D0%B5%D0%B5%D0%BD%D0%B8%D0%B5\_%D0%B2\_JavaScrip

#### Книги

http://kharchuk.ru/JavaScript.pdf подробное руководство

http://dev.watcom.ru:8088/trac/common/books/%D0%9A%D1%80%D0%BE%D0 %BA%D1%84%D0%BE%D1%80%D0%B4%20- %20JavaScript.%20%D0%A1%D0%B8%D0%BB%D1%8C%D0%BD%D1%8B%D0%B5%20%D1%81%D1%82%D0%BE%D1%80%D0%BE%D0%BD%D1%8B.p df JavaScript Сильні сторони

http://www.habucenter.ru/upload/iblock/76a/76a7f6a0d1fda54f72505775f5b9d864.
pdf JavaScript ниндзя

# Відео

https://www.youtube.com/watch?v=M\_pclb-58ZY