

JavaScript: переменные и операции

**WEB
COURSE
ORT DNIPRO**

ORTDNIPRO.ORG/WEB

1. Наши планы на JavaScript

Наши планы на JavaScript

Переменные и операции

Ветвления (условные операторы)

Циклы / Массивы (структуры данных)

Функции

Объекты

Основы программирования



Управление документом (DOM)

Событийная модель в JavaScript

Разработка интерактивных виджетов.

Прикладное применение

2. Переменные

Задача любой программы – обработка данных

```
1
2
3   let price = 799;
4
5   let quantity = 10;
6
7   let totalCost = price * quantity;
8
9   console.log('Total Cost:', totalCost, 'UAH.');
```

Для хранения **данных** (*информации*), в JavaScript используются **переменные**. **Переменные** можно представить как «коробочку» у которой есть название и в которой хранится какое-нибудь значение. Значением может быть число, строка или другие **типы данных** поддерживаемые JavaScript.

Перед тем как использовать, **переменную** нужно **объявить**. Сказать браузеру, что мы хотим создать еще одну «коробочку» для значений и дать ей имя. Объявляются переменные при помощи ключевого слова **let**.

$x = 1$

Variable = Value

3. Ввод/вывод данных

Ввод/вывод данных

```
1
2 //Варианты ввода данных (без разметки)
3 let userName = prompt('Please enter user name');
4
5 let canDrive = confirm('are you can drive?');
6
7 //Варианты вывода данных (без разметки)
8 alert('Hello');
9
10 console.log('User name:', userName);
11
12 document.write(`

# Hello ${userName}</h1>`); 13 14


```

Основной способ ввода данных в **JavaScript** приложения – формы и элементы ввода, а вывод – в разметку документа. В тоже время в **учебных** (и **отладочных**) целях нам будут полезны следующие способы **ввода/вывода** данных (информации).

4. Операции, операторы, операнды и выражения

Оператор присваивания

10

11

12

13

14

```
let a = 2;
```

```
let b = a + 3 * 4;
```



Чтобы указать, что именно нужно записать в переменную используется **оператор присваивания**.

Оператор присвоения берёт то, что справа от него и записывает в переменную имя которой расположено слева от него. Если справа расположено **выражение**, то первым делом оно будет **рассчитано**, и в переменную попадёт уже **результат** расчёта выражения.

Выражения

```
1
2   let a = 3;
3
4   let b = a + 6;
5
6   let c = b + 1;
7
8   c = a + b * c + 7;
9
10  console.log('In Variable C:', c);
11
12
```

По правую сторону от оператора присвоения может быть как конкретное значение, а также может быть **выражение** – формула расчета которую компьютер получит результат который будет записан в переменную имя которой стоит слева от знака присвоения. В **выражении** могут участвовать как и **конкретные значения (константы)** так и другие **переменные**.

Операторы, операнды и операции...

Для выполнения действий (**операций**) над переменными (или значениями) используются **операторы** (которых) существует довольно много).
С некоторыми из них все знакомы, например с арифметическими операторами.

```
12  
13   let a = (2 + 2) * 2;  
14
```

У **операторов** есть **приоритеты**, какой приоритет выше, какой ниже запомнить непросто. Поэтому в случае сомнений какая операция будет первой а какая второй – смело используйте скобки. Принцип их применения такой же как и в математике – скобки повышают приоритет операции в них записанной.

«Скобками программу не испортишь» (с)

Операторы и их приоритеты

Level	Operators
1	() [] .
2	! ~ - ++ --
3	* / %
4	+ -
5	<< >> >>>
6	< <= > >=
7	== !=
8	&
9	^
10	
11	&&
12	
13	?:
14	= += -= *= /= %= <<= >>= >>>= &= ^= =

У операторов есть **приоритеты**, какой приоритет выше, какой ниже запомнить непросто. Поэтому в случае сомнений какая операция будет первой а какая второй – смело используйте скобки. Принцип их применения такой же как и в математике – скобки повышают приоритет операции в них записанной.

5. Типы данных (string & number)

Типы данных (переменных)

```
1
2 let a = 7;
3
4 let b = 8;
5
6 let c = a + b;
7
8 let d = a * b;
9
10 console.log('Value in C:', c); //15
11
12 console.log('Value in D:', d); //56
13
```

```
1
2 let a = '7';
3
4 let b = '8';
5
6 let c = a + b;
7
8 let d = a * b;
9
10 console.log('Value in C:', c); //'78'
11
12 console.log('Value in D:', d); //56
13
```

```
1
2 let a = '7';
3
4 let b = 8;
5
6 let c = prompt('Enter Some Number');
7
8 console.log(typeof(a), typeof(b), typeof(c)); //string, number, string
9
```

В **JavaScript** отсутствует жёсткая **типизация** данных, при которой **тип переменной** определяется при её объявлении. В **JavaScript** **тип переменной** определяется при присваивании ей значения. И может **меняться** при каждом новом присвоении. Мы можем узнать **тип переменной** воспользовавшись функцией **typeof(...)**.

Типы данных (переменных)

Тип данных – пометка для компьютера как относиться к тем или иным данным и **какие операции с ними возможно проводить.**

Тип определяет **возможные значения** и их «смысл», а также **операции** которое возможно выполнять над этими значениями.

undefined, **number**, **string**,
boolean, **function**, **object**, **symbol**, **bigint**

6. Преобразование ТИПОВ

Преобразование типов

Функции **parseInt()/parseFloat()** позволяют преобразовать тип переменной со строкового на числовой (*это возможно если в строке действительно содержатся хоть какие-нибудь цифровые символы, иначе результатом будет значение NaN*). **parseInt()** – работает с целыми числами, **parseFloat()** поддерживает дробные числа.

Также для преобразования типов может быть использован оператор **+** в унарном виде, но в отличии от **parse*-функций** любые нецифровые символы в строке приведут к получению значения **NaN**.

```
3
4 let a = '7.99 $';
5
6 console.log(a, typeof(a)); // '7.99 $' string;
7
8 a = parseFloat(a); // parseInt(a) – для целых чисел;
9
10 console.log(a, typeof(a)); // 7.99 number;
11
```

```
3
4 let a = '7.99';
5
6 console.log(a, typeof(a)); // '7.99' string;
7
8 a = +a; // оператор + в унарном виде;
9
10 console.log(a, typeof(a)); // 7.99 number;
11
```

```
3
4 let a = '7.99 $';
5
6 console.log(a, typeof(a)); // '7.99 $' string;
7
8 a = +a; // оператор + в унарном виде;
9
10 console.log(a, typeof(a)); // NaN number;
11
```

7. NaN

(Not a Number)

NaN – Not a Number

```
4   let a = 'hello';
5
6   let b = 7;
7
8   let c = a * b; //NaN - в результате
    выполнения арифметической операции среди
    операндов оказалось значение, которое
    невозможно преобразовать к числу;
9
10  let d = parseFloat(a); //NaN - невозможно
    даже часть строки преобразовать к числу;
11
```

NaN (Not a Number) –

специальное значение типа number которое показывает, что в результате выполнения арифметической операции (или явного преобразования к числу) один из операндов **не удалось успешно преобразовать к числу**. Поскольку **JavaScript** не типизированный язык то ошибок преобразования типов в нём быть не может, потому и существует такая конструкция как **NaN**.

8. Немного практики #1

Задача: Разработать скрипт который на основании дохода физ. лица рассчитает суммы налогов которые ему необходимо заплатить и сколько у него останется после уплаты налогов.

9. Немного практики #2

Задача: Разработать скрипт который рассчитывает **индекс массы тела** пользователя.

Будет полезным

ВЫРАЗИТЕЛЬНЫЙ JAVASCRIPT

3-Е ИЗДАНИЕ

Современное
веб-программирование

Марейн
Хавербеке



Выразительный JavaScript.

Современное веб-программирование

Марейн Хавербеке

*Замечательная книга,
как для введения в
программирование, так и
освоения JavaScript.*

«Учебник» по JavaScript

JAVASCRIPT.INFO

Мова програмування JavaScript

Тут ви можете вивчити JavaScript, починаючи з нуля і закінчуючи просунутими концепціями, як ООП.
Ми зосередимось на самій мові, зрідка роблячи примітки щодо середовищ її виконання.

Вступ

- | | |
|------------------------------|----------------------------|
| 1.1 Вступ до JavaScript | 1.3 Редактори коду |
| 1.2 Довідники й специфікації | 1.4 Інструменти розробника |

Основи JavaScript

- | | | |
|---------------------------------------|--------------------------------------|--------------------------------|
| 2.1 Привіт, світ! | 2.7 Перетворення типу | 2.13 Цикли: while і for |
| 2.2 Структура коду | 2.8 Базові оператори, математика | 2.14 Конструкція "switch" |
| 2.3 Сучасний режим, "use strict" | 2.9 Оператори порівняння | 2.15 Функції |
| 2.4 Змінні | 2.10 Умовні розгалуження: if, '?' | 2.16 Функціональні вирази |
| 2.5 Типи даних | 2.11 Логічні оператори | 2.17 Стрілкові функції, основи |
| 2.6 Взаємодія: alert, prompt, confirm | 2.12 Оператор об'єднання з null '??' | 2.18 Особливості JavaScript |

Якість коду

- | | | |
|------------------------------|----------------|---|
| 3.1 Налаштування в браузері | 3.3 Коментарі | 3.5 Автоматичне тестування за допомогою Mocha |
| 3.2 Стандарт оформлення коду | 3.4 Ніндзя-код | 3.6 Поліфіли та транспілятори |

Об'єкти: основи

- | | | |
|--------------------------------------|----------------------------------|---------------------------------------|
| 4.1 Об'єкти | 4.4 Методи об'єкта, "this" | 4.7 Тип даних Символ |
| 4.2 Копіювання об'єктів та посилання | 4.5 Конструктори, оператор "new" | 4.8 Перетворення об'єктів в примітиви |
| 4.3 Збирання сміття | 4.6 Опціональний ланцюжок '?.' | |

Типи даних

- | | | |
|-----------------------|------------------------|----------------------------------|
| 5.1 Методи примітивів | 5.5 Методи масивів | 5.9 Object.keys, values, entries |
| 5.2 Числа | 5.6 Ітеративні об'єкти | 5.10 Деструктуроване присвоєння |
| 5.3 Рядки | 5.7 Map та Set | 5.11 Дата і час |
| 5.4 Масиви | 5.8 WeakMap та WeakSet | 5.12 Методи JSON, toJSON |

<https://uk.javascript.info/>

Домашнее задание
Тренируемся!

Программирование ремесло и требует тренировки...

- 1) *Задаётся сторона квадрата. Найти его периметр;*
- 2) *Задаётся длина ребра куба. Найти объем куба и площадь его боковой поверхности;*
- 3) *Задаётся радиус окружности. Найти длину окружности и площадь круга;*
- 4) *Задаются объем и масса вещества. Определить плотность материала этого вещества;*
- 5) *Известны количество жителей в государстве и площадь его территории (в км²). Определить плотность населения в этом государстве.*
- 6). *Даны катеты прямоугольного треугольника. Найти его гипотенузу.*

7) Рассчитать значение y , при любых введенных значениях a :

$$y = \frac{a^2 + 10}{\sqrt{a^2 + 1}}$$



На следующем занятии

JS: условия и ветвление

```
{ if  
  else }
```