

Что собой представляет JavaScript программа

Программа на JavaScript – это последовательность инструкций (statements). Иногда инструкции ещё называют командами JavaScript.

Чтобы программа выполнилась её необходимо запустить. В случае если программа является клиентским сценарием, то её запуск осуществляется автоматически браузером, но только в том случае если она, конечно, присутствует в коде страницы. Осуществляет браузер выполнение программы обычно в процессе загрузки веб-страницы.

Реклама

При написании инструкций, их между собой необходимо разделять точкой с запятой. В JavaScript допускается не ставить точку с запятой в конце инструкции, если следующая инструкция будет начинаться с новой строки.

```
// НЕ РЕКОМЕНДУЕТСЯ
number1 = 10; number2 = 20; sum = a + b;
```

```
// НЕ РЕКОМЕНДУЕТСЯ
number1 = 10
number2 = 20
sum = a + b
```

Завершать инструкции точкой с запятой в JavaScript не требуется, если следующая инструкция будет записана с новой строки, но настоятельно рекомендуется. При этом желательно также каждую инструкцию располагать на отдельной строке.

```
// РЕКОМЕНДУЕТСЯ
number1 = 10;
number2 = 20;
sum = a + b;
```

Инструкции JavaScript состоят из: значений, операторов, выражений, ключевых слов и комментариев.

Например, эта инструкция говорит браузеру написать «Привет, мир!» внутри HTML элемента с id="main":

```
document.getElementById("main").textContent = "Привет, мир!";
```

JavaScript значения

Синтаксис JavaScript определяет следующие типы значений (values): фиксированные значения (литералы), значения констант и значения переменных.

Примеры литералов для представления различных значений в JavaScript:

```
17          // литерал целого числа
23.8        // литерал дробного числа
'HTML'      // литерал строки
"HTML"      // литерал строки
true        // логический литерал
[]          // литерал массива
```

```
[1,7,20] // литерал массива
{}       // литерал объекта
{
  name: 'Деннис',
  language: 'ru'
}        // литерал объекта
(ab|bc)  // литерал регулярного выражения
```

В языках программирования константы и переменные используются для хранения значений данных.

Для объявления констант используется ключевое слово `const`, переменных — `var` и `let`.

```
const MAX_LENGTH = 17; // константа
var section = 'HTML';  // переменная
let arr = ['HTML', 'CSS']; // переменная
```

`let` — это новый способ объявления переменных, который появился начиная с версии ES-2015 (ES6). Отличается `let` от `var` тем, что переменная, объявленная с помощью `let`, имеет другую область видимости (блочную), а также ещё тем, что к ней можно обратиться только после её создания.

Операторы JavaScript

В JavaScript имеется очень много различных операторов. Они обозначают операцию, которую нужно произвести.

В зависимости от количества операндов, с которыми оперирует оператор, они делятся на бинарные и унарные. Но в JavaScript также имеется ещё условный оператор, который состоит из 3 операндов. Данный оператор называется тернарным.

Например, оператор присваивания (знак `=`):

```
// operand1 = operand2;
coordX = 100; // например присвоить переменной coordX число 100
```

Данный оператор используется для присвоения (установки) `operand1` такого значения, которое имеет `operand2`.

Оператор присваивания является бинарным, т.к. он оперирует двумя операндами.

Например, оператор инкремента (знак `++`) используется для увеличения числового значения операнда на 1:

```
var num = 5;
++num; // 6
```

Оператор инкремента является унарным, т.к. он оперирует одним операндом.

JavaScript выражения

Выражение (expression) - это комбинация значений, переменных и операторов, которые либо присваивают переменной значение, либо возвращают какое-то значение без его присваивания.

Например, выражение `num = 955.47` использует оператор `=` для присваивания переменной `num` значения `955.47`.

Например, выражение `'Привет, ' + 'Тимофей'` использует оператор `+` для сложения строк `'Привет, '` и `'Тимофей'` без присваивания.

Ключевые слова JavaScript

Инструкции JavaScript часто начинаются с ключевого слова (keyword). Оно (ключевое слово) предназначено для того чтобы определить какое действие JavaScript необходимо выполнить.

Например, ключевое слово `var` указывает браузеру создать переменные:

```
var
  coordX = 5,
  coordY = 130;
```

Список некоторых ключевых слов: `break`, `continue`, `do ... while`, `for`, `function`, `if ... else`, `return`, `switch`, `try ... catch`, `var`.

Реклама

Комментарии в JavaScript

Не все инструкции JavaScript являются выполняемыми.

Код после двойных косых черт `//` или между `/*` и `*/` рассматривается как комментарий.

Комментарии игнорируются и не выполняются:

```
var direction = 'top'; // однострочный комментарий
// var speed = 10; эта инструкция не будет выполнена

/* Функция,
   добавляющая к элементу с id="myBtn" класс show */
function showBtn() {
  document.getElementById("myBtn").classList.add("show");
}
```

`//` - является однострочным комментарием, т.е. любой текст между `//` и концом строки будет игнорироваться JavaScript (не будет выполняться).

`/* */` - является многострочным комментарием, любой текст, находящийся между `/*` и `*/` будет игнорироваться JavaScript.

Блок инструкций JavaScript

JavaScript инструкции можно сгруппировать с помощью блока (фигурных скобок {...}).

Как правило, используется в функциях, управляющих структурах (for, while) и др.

```
function myFunction() {  
    document.getElementById('id__1').textContent = 'Текст 1...';  
    document.getElementById('id__2').textContent = 'Текст 2...';  
}
```

Чувствительность к регистру

JavaScript является регистрозависимым языком. Это значит, что ключевые слова, переменные, имена функций и другие идентификаторы языка должны содержать одинаковые наборы прописных и строчных букв. Например, метод alert должно набираться только как alert, а не Alert, ALERT и т.п.

Пробелы в JavaScript

JavaScript игнорирует несколько пробелов. Можно добавить пробелы в сценарий, чтобы сделать его более читаемым.

Следующие строки эквивалентны:

```
var str = 'Текст';  
var str='Текст';
```

Рекомендуется размещать пробелы вокруг операторов: =, +, -, * и /.

Длина строки кода

Для лучшей читаемости кода рекомендуется избегать строк с длиной более 80 символов.

Если JavaScript инструкция имеет большую длину, то её желательно разорвать. В большинстве случаев разрыв инструкции лучше осуществить после какого-нибудь оператора.

Например:

```
document.getElementById('aside__nav_header').innerHTML =  
    'Дополнительная навигация (по разделу)';
```

Рекомендации по именованию переменных и функций

Основные рекомендации:

- имя, состоящее из одного слова, следует писать строчными буквами (например: article);
- имя, состоящее из нескольких слов, следует писать слитно строчными буквами, кроме букв на стыке слов, их необходимо писать прописными (например: articleTitle);

- количество слов в имени не должно превышать 3 (например: `articleDatePublishedon`)
- переменные (не для хранения функций) и свойства объектов должны быть существительными (например: `textComment`);
- массивы и коллекции значений следует задавать существительными во множественном числе (например: `lastComments`);
- именовать функции и методы объектов необходимо глаголами (например: `getLastArticles`);
- именование классов необходимо начинать с прописной буквы (например: `Comments`).