Объявление переменных

Смотреть материал на видео

На предыдущем занятии мы поговорили в целом о языке JavaScript и о способах его подключения к HTML-документам с помощью тега script. Также мы запускали в браузере скрипт, записанный в отдельном файле. На этом занятии мы начнем непосредственное изучение этого языка программирования и все примеры я буду писать в этом отдельном файле. Вы можете поступить также, либо записывать скрипты непосредственно в HTML-документе внутри тега script. На данном этапе обучения это не принципиально.

Правилом хорошего тона среди программистов считается написание понятного текста программы не только разработчику, но и всем другим членам команды. И первое, что для этого нужно – это писать комментарии в тексте скриптов: уместные и понятные. Что такое комментарии? Это произвольный текст, который видит программист, но игнорируется при компиляции. Например, простой однострочный комментарий выглядит так:

```
//Это моя первая программа
```

Здесь вначале стоят две косые черты, которые и говорят JavaScript-компилятору, что это комментарий, а не текст программы и его нужно игнорировать. Для создания многострочных комментариев используются символы:

```
/*Это моя
первая программа
*/
```

Причем, смотрите, все комментарии текстовый редактор Sublime Text отображает серым цветом, что удобно для быстрой ориентации в программе.

Также комментарии часто используют при отладке скриптов, когда часть кода помещают в комментарии и он перестает существовать для компилятора кода. Например, вот так:

```
//console.log(obj);
```

Если мы теперь запустим программу, то увидим, что она выполняется без ошибок, так как мы поместили в комментарий строчку, которая вызывала ошибку.

Начиная со стандарта ES6 от 2015 года, JavaScript заметно изменился и современным браузерам теперь приходится учитывать как «старый код», написанный по стандарту ES5 и ниже, так и новый – по стандарту ES6 и выше. Если вы пишите программу по новым стандартам, то лучше явно сообщить об этом браузеру, чтобы его JavaScript машина переключилась в «современный» режим работы. Это делается с помощью директивы

"use strict" или 'use strict'

которая записывается вначале скрипта. Обратите внимание, что эта директива должна идти первой, если до нее написать какую-либо команду JavaScript, то строгий режим не включится! Допускается перед этой директивой писать только комментарии. После этой директивы лучше поставить еще точку с запятой, иначе в некоторых очень редких случаях это может приводить к некоторым проблемам.

Я рекомендую всегда использовать эту директиву и программировать в стандарте ES6+. Во всех наших занятиях я буду исходить из того, что эта директива включена.

Если ранее вы не изучали никакой язык программирования, то следующая информация будет для вас полезной. Давайте ответим на вопрос: что в целом умеют делать программы на компьютере? На самом деле не так уж и много. Они могут:

- хранить данные в переменных;
- выполнять арифметические операции;
- проверять условия (реализация операторов ветвления);
- организовывать циклы.

И по большому счету – это все! Да, все многообразие программ – это комбинация таких простых операций. И изучать язык программирования следует с этих моментов. Начнем с самого начала – с описания переменных в JavaScript.

В новых стандартах переменные задаются с помощью ключевого слова let по следующему синтаксису:

let <имя переменной> [= присваиваемое значение];

Как это было в старых стандартах я здесь говорить не буду, так как лучше учиться сразу с современным уклоном. Итак, в самом простом случае переменную можно объявить так:

```
let message;
```

Здесь message – это имя переменной и ей ничего не присваивается. Почему имя переменной именно message? Да, в общем-то, не почему, просто такое имя мне пришло в голову. Ее можно было бы назвать и иначе, например,

msg или var или a или str и т.д.

Здесь важно лишь следовать правилам задания таких имен. Во-первых, пишите их только на латинице, не используя другие символы (например, кириллицу). Далее, в качестве первого символа можно использовать или буквы (по нашей договоренности – это символы латиницы от a до z или от A до Z), а также символ подчеркивания _ и $. Никакие другие символы писать не стоит. Далее, вторым и последующими символами могут быть еще и цифры. Вот примеры правильных и неправильных имен переменных.

Правильные имена	Неправильные имена
var_i	1var
_a	@a
$1	1_
__msg	don't
Arg$_	_#_
c1, a4, f546	email@bk.ru
ARG	a-b

Причем, переменная arg и Arg – это две разные переменные, т.к. язык JavaScript чувствителен к регистру и, например, буквы 'a' и 'A' – это два разных символа.

И последнее – имена переменных должны отражать смысл хранимых данных, то есть, они должны быть осмысленными и быть существительными (то есть, отвечать на вопрос: кто, что). Например,

если переменная хранит emai-адрес, то так ее и назовите:

```
let email;
```

но не sendEmail – это уже глагол, действие. Далее, если это какой-либо цвет, то пусть она называется

```
let color;
let clr;
```

и так далее. Помните, что программу, возможно, будут читать другие программисты из вашей команды и им должно быть все предельно ясно что за переменная и какую роль она играет в скрипте.

С именами разобрались. Следующий момент – это точка с запятой. Строго говоря, если каждую конструкцию записывать с новой строчки, то интерпретатор JavaScript автоматически их добавит в конце каждой строки. Но это порочная практика и потенциальный источник возможных ошибок. Интерпретатор не всегда корректно определяет места их размещения, поэтому рекомендуется ставить точки с запятой после каждой конструкции языка. Именно так я буду делать на наших занятиях.

С основными моментами определились, давайте теперь выведем значение нашей переменной message в консоль:

```
console.log(message);
```

Мы видим значение

undefined

и это логично, так как в JavaScript любая переменная, которой не было присвоено никакого значения, принимает значение undefined. Присвоим ей значение, запишем в таком виде:

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

Теперь, при выводе в консоль мы видим строчку «Hello». Строки можно записывать еще и такими способами:

```
let message = 'Hello';
let message = `Hello`;
```

Об особенностях последнего способа мы подробнее поговорим позже. Но все это одна и та же строка. Или же переменную можно сначала объявить:

```
let message;
```

а, затем, присвоить значение:

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

Обратите внимание, что я здесь не пишу второй раз ключевое слово let. Оно указывается только один раз при объявлении переменной, а затем, мы обращаемся к ней просто по имени. Далее, равно здесь – это оператор присваивания, который присваивает операнду слева значение, стоящее справа от него. По аналогии, можно задать сразу несколько переменных:

```
let user = 'Alex', age = 25, emai
```

их часто записывают еще и в таком более наглядном виде:

```
let user = 'Alex',
         age = 25,
         email='1@my.ru';
```

все это одно и то же. Главное здесь, чтобы имена переменных были уникальными: нельзя объявлять две переменные с одинаковыми именами. Теперь, если мы хотим изменить значение какой-либо переменной, то это делается так:

```
age = 17;
email='mymail@my.ru';
```

Мы даже можем присвоить им такие значения:

```
age = 'Возраст 17 лет';
email = 0;
```

То есть, переменная age, которая содержала число, теперь хранит строку, а переменная email, наоборот, вместо строки – число. Это возможно, так как JavaScript относится к языкам программирования со слабой типизацией, в отличие, например, от C++, Java, Pascal и другим подобным. Здесь, во-первых, при объявлении переменной не указывается ее тип, а во-вторых, переменной можно присваивать самые разные данные.

Далее, мы можем сделать и так:

```
email = age;
```

в этом случае данные из age будут скопированы в переменную email. Такие операции можно делать с любыми переменными.

Если же в программе требуется создать неизменяемую переменную, то есть, константу, например, PI = 3.1415, то для этого ее следует объявить с помощью ключевого слова const:

const <имя константы> = значение;

например, так:

```
const PI = 3.1415;
```

Теперь, при попытке ее изменения

```
PI = 2;
```

мы в консоли увидим ошибку в этой строчке, означающую, что константу менять нельзя.

Обычно, в скриптах имена констант записываются заглавными буквами. Это помогает программисту, во-первых, лучше видеть их в тексте программы и, во-вторых, понимать, что это константа, а не переменная.

Вот мы с вами в целом узнали что такое переменные и константы в JavaScript. На следующем занятии продолжим эту тему и узнаем какие типы данных существуют в этом языке программирования.

Видео по теме

JavaScipt #1: что это такое, с чего начать, как внедрять и запускать

JavaScipt #2: способы объявления переменных и констант в стандарте ES

← Предыдущая Следующая →