

ПЕРЕМЕННЫЕ

ПРИМИТИВНЫЕ ТИПЫ ДАННЫХ

Каждая программа в конечном итоге сводится к манипулированию данными.

Для того, чтобы осуществлять манипуляции с данными, используя возможности языка, программа должна сначала где-то сохранить эти данные.

Данные хранятся в переменных.

Переменная - поименованная выделенная область памяти.

Комментарии и отладка

Комментарии служат для заметок, не влияют на ход выполнения программы.

```
1 // однострочный комментарий
2 /* многострочный
3    комментарий */
```

Для вывода информации в консоль браузера используется:

```
1 console.log("Текст", varName);
```

Дополнительные возможности вывода

отличия только в формате вывода

```
1  /* выводит информационное сообщение */
2  console.info("Информация!");
3  /* выводит предупреждающее сообщение */
4  console.warn("Предупреждение!!!");
5  /* выводит сообщение об ошибке */
6  console.error("Здесь ошибка... Исправьте");
7
8  /* замер времени выполнения кода */
9  console.time('Счетчик №1');
10 /* код для которого измеряем время выполнения */
11 console.timeEnd('Счетчик №1');
```

В JavaScript переменные должны быть объявлены, перед тем, как они будут использоваться.

Объявление переменных:

1. оператор `let` / `const` / `var`
2. имя переменной

По имени переменной и осуществляется доступ к данным, которые в ней хранятся.

Объявление переменных, присваивание значений

```
1 let messageId; // объявили переменную с именем messageId
2
3 let a, b; // объявили несколько переменных
4 a = 4; // переменной a присвоили значение 4
5 a = 500000; // переопределили значение переменной a
6
7 const pi = 3.14; // объявили константу и присвоили ей значение
8 /* константам необходимо присваивать значения при инициализации,
9    их нельзя изменить в дальнейшем */
10
11 let size = 20, page = 3; // объявили переменные и присвоили значения
12 size = 10; // переопределили значение переменной size
13 page = 5; // переопределили значение переменной page
14
15 // обратились к переменной pi по имени, вывели его в консоль
16 console.log(pi);
```

Требования к именам переменных

1. имя может состоять из: букв, цифр, символов \$ и _
2. первый символ не должен быть цифрой
3. имена переменных должны быть именами существительными и описывать сущность хранимого значения
4. регистр букв в именах переменных имеет значение
5. для имен переменных нельзя использовать зарезервированные слова, например, function, return и тд.

Типы данных, используемые в JavaScript:

- Примитивные типы данных
- Ссылочные типы данных

Примитивные типы данных.

Числа

Объявление

целые числа

```
let a = 90, b = -12;
```

числа с плавающей точкой

```
let height = 5.67;
```

NaN (Not-A-Number)

```
let sum = 5 / "пять";
```

Примитивные типы данных.

Строки

Объявление

в двойных кавычках

```
let login = "wind";
```

в одинарных кавычках

```
let pwd = 'qw123'
```

в обратных кавычках

```
let p = `

${login}</p>`


```

Примитивные типы данных

Логические значения	Объявление
---------------------	------------

<code>true</code>	<code>let selected = true;</code>
<code>false</code>	<code>let pressed = false;</code>

Преобразование к `true/false` происходит в логическом контексте и при применении логических операторов.

При этом к `false` преобразуются: `null`, `undefined`, пустые строки, `0`, `NaN`.

Все остальные значения преобразуются к `true`.

Примитивные типы данных

Значение неизвестно

Объявление

`null`

`let elem = null;`

Примитивные типы данных

Значение не присвоено	Объявление
-----------------------	------------

undefined

let age;

Определение типа данных переменной

```
1  
2 let dataType = typeof login;  
3 console.log(dataType);  
4
```

Некоторые особенности примитивных типов данных:

- если математическая операция не может быть совершена, возвращается NaN. На NaN можно проверить функцией `isNaN`(выражение/переменная)

```
1 console.log(isNaN(5 / "ноль")); // true
2 console.log(isNaN("45")); /* false */ console.log(isNaN(45)); /* false
```

- деление чисел на ноль

```
1 let a = 8.1, b = 0;
2 let res = a / b; /* значение res будет равно Infinity */
```