

# Переменные

```
let message = "string";
```

Может хранить в себе любой тип данных (строки, числа, массивы и т.д.);

# Массивы

## Создание массива

```
let arr = new Array();
```

```
let arr = [];
```

## Хранимые типы данных

```
1 // разные типы значений
2 let arr = [ 'Яблоко', { name: 'Джон' }, true, function() { alert('привет'); } ];
3
4 // получить элемент с индексом 1 (объект) и затем показать его свойство
5 alert( arr[1].name ); // Джон
6
7 // получить элемент с индексом 3 (функция) и выполнить её
8 arr[3](); // привет
```

# Методы массивов

Методы, работающие с концом массива:

## pop

Удаляет последний элемент из массива и возвращает его:

```
1 let fruits = ["Яблоко", "Апельсин", "Груша"];
2
3 alert( fruits.pop() ); // удаляем "Груша" и выводим его
4
5 alert( fruits ); // Яблоко, Апельсин
```

## push

Добавляет элемент в конец массива:

```
1 let fruits = ["Яблоко", "Апельсин"];
2
3 fruits.push("Груша");
4
5 alert( fruits ); // Яблоко, Апельсин, Груша
```

## Методы, работающие с началом массива:

### `shift`

Удаляет из массива первый элемент и возвращает его:

```
1 let fruits = ["Яблоко", "Апельсин", "Груша"];
2
3 alert( fruits.shift() ); // удаляем Яблоко и выводим его
4
5 alert( fruits ); // Апельсин, Груша
```

### `unshift`

Добавляет элемент в начало массива:

```
1 let fruits = ["Апельсин", "Груша"];
2
3 fruits.unshift('Яблоко');
4
5 alert( fruits ); // Яблоко, Апельсин, Груша
```

Методы `push` и `unshift` могут добавлять сразу несколько элементов:

# Селекторы

```
1 | let primaryHeadings = document.getElementsByTagName('h1');
```

Данный код извлекает все заголовки h1 и, грубо говоря, помещает их в переменную primaryHeadings. Если на странице присутствует несколько заголовков, то все они будут помещены в массив.

```
1 | let ou812 = document.getElementById('eddie');
```

Данный код выбирает элемент с id = "eddie".

```
1 | let guitars = document.getElementsByClassName('axes');
```

Также мы можем выбирать элементы по названию их классов.

# События

| Событие     | Описание                                          |
|-------------|---------------------------------------------------|
| onchange    | HTML элемент был изменен                          |
| onclick     | Пользователь кликнул мышкой на HTML элемент       |
| onmouseover | Пользователь навел мышку на HTML элемент          |
| onmouseout  | Пользователь вывел мышку за пределы HTML элемента |
| onkeydown   | Пользователь нажал на клавишу клавиатуры          |
| onload      | Браузер закончил загружать страницу               |

```
<script>
  function countRabbits() {
    for(let i=1; i<=3; i++) {
      alert("Кролик номер " + i);
    }
  }
</script>

<input type="button" onclick="countRabbits()" value="Считать кроликов!">
```