

— JAVASCRIPT —

Таймеры. JSON. Исключения. Web-Storage. Cookie.

setTimeout() / clearTimeout()

```
setTimeout(function() {  
    alert( 'Привет' );  
}, 1000);
```

```
function myFunction() {  
    alert( 'Привет' );  
}  
  
setTimeout(myFunction, 1000);
```

```
var timerId = setTimeout(myFunction, 1000);  
  
clearTimeout(timerId);
```

setInterval() / clearInterval()

```
function myFunction() {  
    alert( 'Привет' );  
}  
  
var timerId = setInterval(myFunction, 1000);
```

```
clearInterval(timerId);
```

JSON

формат данных,

который используется для представления объектов в виде строки

```
{  
  "firstName": "Иван",  
  "lastName": "Иванов",  
  "isAdmin": true,  
  "age": null,  
  "address": {  
    "streetAddress": "Московское ш., 101, кв.101",  
    "city": "Ленинград",  
    "postalCode": 101101  
  },  
  "phoneNumbers": [  
    "812 123-1234",  
    "916 123-4567"  
  ]  
}
```

JSON

`JSON.stringify()`

возвращает строку JSON

```
var user = {
  firstName: 'Иван',
  lastName: 'Иванов',
  isAdmin: true,
  age: null,
  address: {
    streetAddress: 'Московское ш., 101, кв.101',
    city: 'Ленинград',
    postalCode: 101101
  },
  phoneNumbers: [
    '812 123-1234',
    '916 123-4567'
  ]
}
```

```
var userString = JSON.stringify(user);
```

userString

```
"
{"firstName":"Иван","lastName":"Иванов","isAdmin":true,"age":null,"address":
{"streetAddress":"Московское ш., 101, кв.101","city":"Ленинград",
"postalCode":101101},"phoneNumbers":["812 123-1234","916 123-4567"]}"
```

JSON

JSON.parse()

Разбирает строку JSON и возвращает разобранное значение

```
var userString = '{"firstName": "Иван", "lastName": "Иванов", "isAdmin": true, "age": null, "address": {"city": "Ленинград", "postalCode": 101101, "streetAddress": "Московское ш., 101, кв.101"}}';  
var user = JSON.parse(userString);
```

```
user  
  Object {firstName: "Иван", lastName: "Иванов",  
    ▼ isAdmin: true, age: null, address: Object...}  
    i  
    ▼ address: Object  
      city: "Ленинград"  
      postalCode: 101101  
      streetAddress: "Московское ш., 101, кв.101"  
      ► __proto__: Object  
    age: null  
    firstName: "Иван"  
    isAdmin: true  
    lastName: "Иванов"  
    ► phoneNumbers: Array(2)  
    ► __proto__: Object
```

JSON

toJSON()

```
function User (name, skills){
  this.name = name,
  this.skills = skills,
  this.toJSON = function() {
    return this.skills;
  }
}

var user = new User('Alex', ['js', 'html']);
var userString = JSON.stringify(user);
```

userString

"["js","html"]"

```
function User (name, skills){
  this.name = name,
  this.skills = skills,
  this.toJSON = function() {
    return this.name;
  }
}

var user = new User('Alex', ['js', 'html']);
var userString = JSON.stringify(user);
```

userString

""Alex""

— try..catch —

```
var userString = '{some Bad Json}';

try {
  var user = JSON.parse(userString);
} catch (err) {
  console.log(err);
}
```

```
var userString = '{some Bad Json}';

try {
  var user = JSON.parse(userString);
} catch (err) {
  console.log(err);
}
```

SyntaxError: Unexpected token s in JSON at position 1 VM124:6
at JSON.parse (<anonymous>)
at <anonymous>:4:19

undefined

|

— try..catch —

```
var userString = '{ "age": 30 }'; // данные неполны

try {

    var user = JSON.parse(userString); // <-- выполнится без ошибок

    if (!user.name) {
        throw new SyntaxError('В данных отсутствует "name"');
    }

    console.log( user.name );

} catch (e) {
    console.log( "Извините, в данных ошибка" );
}
```

Извините, в данных ошибка

Window.sessionStorage

Данные `sessionStorage` очищаются в момент окончания сессии текущий страницы

```
// Сохранение данных в sessionStorage
sessionStorage.setItem('key', 'value');

// Получение данных из sessionStorage
var data = sessionStorage.getItem('key');
|
console.log(sessionStorage.length); //1

// Удаление данных из sessionStorage
sessionStorage.removeItem('key');
console.log(sessionStorage.length); //0
```

Window.localStorage

Данные в *localStorage* не имеют ограничений по времени хранения и могут быть удалены только с помощью JavaScript

```
localStorage.setItem('key', 'value');  
|  
var data = localStorage.getItem('key');  
console.log(localStorage.length);  
  
localStorage.removeItem('key');  
console.log(localStorage.length);
```

WebStorage

```
var user = {  
  name: 'Alex',  
  age: 30,  
  toJSON: function () {  
    return this.name;  
  }  
}
```

```
sessionStorage.setItem('user', JSON.stringify(user));
```

```
var userString = sessionStorage.getItem('user');
```

document.cookie

не объект, а строка в специальном формате,
для удобной манипуляций с которой нужны дополнительные функции

```
function getCookie(name) {  
    var matches = document.cookie.match(new RegExp(  
        "(?:^|; )" + name.replace(/[\. $? * | { } \ ( \) \[ \] \\ \/ \+ ^ ]/g, '\\$1') + "=([^\;]*)"  
    ));  
    return matches ? decodeURIComponent(matches[1]) : undefined;  
}
```

document.cookie

```
function setCookie(name, value, options) {
  options = options || {};

  var expires = options.expires;

  if (typeof expires == "number" && expires) {
    var d = new Date();
    d.setTime(d.getTime() + expires * 1000);
    expires = options.expires = d;
  }
  if (expires && expires.toUTCString) {
    options.expires = expires.toUTCString();
  }

  value = encodeURIComponent(value);

  var updatedCookie = name + "=" + value;

  for (var propName in options) {
    updatedCookie += "; " + propName;
    var propValue = options[propName];
    if (propValue !== true) {
      updatedCookie += "=" + propValue;
    }
  }

  document.cookie = updatedCookie;
}
```

```
function deleteCookie(name) {
  setCookie(name, "", {
    expires: -1
  })
}
```

— JAVASCRIPT —

Таймеры. JSON. Исключения. Web-Storage. Cookie.