

Утки, JSON, таймеры и try ... catch

Занятие 11

План занятия

- Полиморфные функции
- JSON
- setTimeout, setInterval
- eval
- try ... catch



Типизация объектов

typeof

```
alert( typeof 1 );           // 'number'
alert( typeof true );        // 'boolean'
alert( typeof "Текст" );      // 'string'
alert( typeof undefined );    // 'undefined'
alert( typeof null );         // 'object' (ошибка в языке)
alert( typeof alert );        // 'function'
```

```
alert( typeof {} ); // 'object'
alert( typeof [] ); // 'object'
alert( typeof new Date ); // 'object'
```

[[Class]]

```
var toString = {}.toString;
```

```
var arr = [1, 2];  
alert( toString.call(arr) ); // [object Array]
```

```
var date = new Date;  
alert( toString.call(date) ); // [object Date]
```

```
var user = { name: 'Name' };  
alert( toString.call(user) ); // [object Object]
```

[[Class]]

```
function getClass(obj) {  
    return {}.toString.call(obj).slice(8, -1);  
}
```

```
alert( getClass(new Date) ); // Date  
alert( getClass([1, 2, 3]) ); // Array
```

instanceof

```
function User() {}
```

```
var user = new User();
```

```
alert( user instanceof User ); // true
```


Утиная типизация



Утиная типизация

```
var something = [1, 2, 3];
```

```
if (something.splice) {  
    alert( 'Кря! Кря! Это массив!' );  
}
```

```
var x = new Date();
```

```
if (x.getTime) {  
    alert( 'Date!' );  
    alert( x.getTime() );  
}
```

Полиморфная функция

```
function sayHi(who) {  
    // если есть forEach  
    if (who.forEach) {  
        // предполагаем, что он ведёт себя 'как надо'  
        who.forEach(sayHi);  
    } else {  
        alert( 'Привет, ' + who );  
    }  
}
```

Полиморфная функция

```
function sayHi(who) {  
    if (Array.isArray(who)) {  
        who.forEach(sayHi);  
    } else {  
        alert( 'Привет, ' + who );  
    }  
}  
  
// Вызов с примитивным аргументом  
sayHi( 'Вася' ); // Привет, Вася  
  
// Вызов с массивом  
sayHi( [ 'Саша', 'Петя' ] ); // Привет, Саша... Петя  
  
// Вызов с вложенными массивами – тоже работает!  
sayHi( [ 'Саша', 'Петя', [ 'Маша', 'Юля' ] ] );  
// Привет Саша..Петя..Маша..Юля
```



JSON

Формат JSON

- JavaScript-объекты { ... } или
- Массивы [...] или
- Значения одного из типов:
 - строки в двойных кавычках,
 - число,
 - логическое значение true/false,
 - null.

Метод JSON.parse

```
var numbers = '[0, 1, 2, 3]';
```

```
numbers = JSON.parse(numbers);
```

```
alert( numbers[1] ); // 1
```

```
var user = '{ "name": "Бася", "age": 35,  
  "isAdmin": false, "friends": [0,1,2,3] }';
```

```
user = JSON.parse(user);
```

```
alert( user.friends[1] ); // 1
```

Метод JSON.stringify

```
var event = {  
    title: 'Конференция',  
    date: 'сегодня'  
};
```

```
var str = JSON.stringify(event);  
alert( str ); //  
{"title":"Конференция","date":"сегодня"}
```

```
// Обратное преобразование.  
event = JSON.parse(str);
```

Метод JSON.stringify

```
var room = {
  number: 23,
  occupy: function() {
    alert( this.number );
  }
};

event = {
  title: 'Конференция',
  date: new Date(Date.UTC(2014, 0, 1)),
  room: room
};

alert( JSON.stringify(event) );
/*
{
  "title":"Конференция",
  "date":"2014-01-01T00:00:00.000Z",    // (1)
  "room": {"number":23}              // (2)
}
*/
```


Advanced

JSON.parse(str, reviver)

JSON.stringify(value, replacer, space)

A large, multi-story stone building with a green roof and a stone wall in the foreground. The building has many windows and a prominent green dome on the right side. The sky is blue with white clouds. The foreground is a grassy field.

Таймеры

setTimeout

```
var timerId = setTimeout(func, delay[, arg1, arg2...])
```

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

clearTimeout

```
var timerId = setTimeout(function() { alert(1) }, 1000);  
alert(timerId); // число – идентификатор таймера  
  
clearTimeout(timerId);  
alert(timerId);  
// всё ещё число, оно не обнуляется после отмены
```


setInterval

```
var timerId = setInterval(func, delay[, arg1, arg2...])
```

```
// начать повторы с интервалом 2 сек  
var timerId = setInterval(function() {  
    alert( 'пиу' );  
}, 2000);
```

```
// через 5 сек остановить повторы  
setTimeout(function() {  
    clearInterval(timerId);  
    alert( 'STOP' );  
}, 5000);
```

Рекурсивный setTimeout

*/** ВМЕСТО:*

```
var timerId = setInterval(function() {  
    alert( 'пиу' );  
}, 2000);  
*/
```

```
var timerId = setTimeout(function tick() {  
    alert( 'пиу' );  
    timerId = setTimeout(tick, 2000);  
}, 2000);
```

Задержка таймера

`setTimeout(...,1) setTimeout(...,4)`



eval

GO IT

goit.com.ua

WWW.TUR-174.RU

eval

```
var a = 1;  
  
(function() {  
    var a = 2;  
    eval(' alert(a) '); // 2  
})();
```



try..catch

try..catch

```
try {  
    // код ...  
} catch (err) {  
    // обработка ошибки  
}
```

try..catch

```
try {  
    alert('Начало блока try'); // (1) <--  
    lalala; // ошибка, переменная не определена!  
    alert('Конец блока try'); // (2)  
} catch(e) {  
    alert('Ошибка ' + e.name + ':' + e.message + '\n' + e.stack); // (.  
}  
  
alert('Потом код продолжит выполнение...');
```

Оператор throw

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

try {

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

    if (!user.name) {
        throw new SyntaxError('Данные некорректны');
    }

    alert( user.name );

} catch (e) {
    alert( 'Извините, в данных ошибка' );
}
```

План занятия

- Полиморфные функции
- JSON
- setTimeout, setInterval
- eval
- try ... catch



2015