

Курс: Frontend с нуля

Тренер:
Максим Шепель

Занятие 10
JavaScript: объекты и массивы



Содержание занятия

- Обработка ошибок
- Объекты: определение и работа со свойствами
- Сериализация и десериализация объектов. Формат JSON
- Массивы: определение и методы
- Массивоподобные объекты

Обработка ошибок

```
try {  
    if (!isFinite(x)) {  
        throw new Error('x is not valid number');  
    }  
} catch (err) {  
    console.log(err.toString());  
} finally {  
    console.log('operation finished');  
}
```

Создание объекта

```
var objLiteral = {  
    index: 1,  
    name: 'literal'  
},  
objConstructed = new Object({  
    index: 2,  
    name: 'constructor'  
}),  
objCreated = Object.create({  
    index: 3,  
    name: '"create" method'  
}),  
objNull = Object.create(null);
```

Создание объекта

```
console.log(objCreated); // >>> Object {}  
console.log(objCreated.index); // >>> 3
```

```
console.log(objCreated.__proto__);  
// >>> Object {index: 1, name: '"create" method'}
```

```
console.log(objCreated.__proto__.isPrototypeOf(objCreated));  
// >>> true
```

```
console.log(objConstructed.__proto__ == Object.prototype);  
// >>> true
```

```
console.log(objLiteral.toString()); // >>> "[object Object]"  
console.log(objConstructed.toString()); // >>> "[object Object]"  
console.log(objCreated.toString()); // >>> "[object Object]"
```

```
console.log(objNull.toString());  
// >>> Uncaught TypeError: objNull.toString is not a function
```

Удаление свойств

```
var protoObject = {  
    name: 'sample'  
},  
obj = Object.create(protoObject);  
  
delete obj.name;  
console.log(obj.name); // >>> sample  
delete protoObject.name;  
console.log(protoObject.name); // >>> undefined  
console.log(obj.name); // >>> undefined
```

Проверка свойств

```
var protoObject = {  
    name: 'sample'  
},  
obj = Object.create(protoObject);  
  
console.log(obj.hasOwnProperty('name')); // >>> false  
console.log(obj.hasOwnProperty('toString')); // >>> false  
obj.index = 0;  
console.log(obj.hasOwnProperty('index')); // >>> true
```

Геттеры и сеттеры

```
var circle = {  
  radius: 10,  
  get length () {  
    return 2 * Math.PI * this.radius;  
  },  
  set length (newLength) {  
    var ratio = newLength / this.length;  
    this.radius *= ratio;  
  }  
};
```

```
console.log(circle.length); // >>> 62.83185307179586  
circle.radius = 5;  
console.log(circle.length); // >>> 31.41592653589793  
circle.length = 62.83185307179586;  
console.log(circle.radius); // >>> 10
```


Атрибуты свойств

```
var obj = Object.create({  
    name: 'sample'  
});
```

```
Object.defineProperty(obj, 'index', {  
    value: 0,  
    enumerable: false,  
    writable: false,  
    configurable: true  
});
```

```
console.log(propertyIsEnumerable('name')); // >>> false  
console.log(propertyIsEnumerable('index')); // >>> false  
console.log(Object.keys(obj)); // []
```

Атрибуты свойств

```
Object.defineProperty(obj, 'index', {  
    enumerable: true  
});
```

```
console.log(propertyIsEnumerable('index')); // >>> true  
console.log(Object.keys(obj)); // ['index']
```

```
obj.index = 1;  
console.log(obj.index); // >>> 0
```

```
Object.defineProperty(obj, 'index', {  
    writable: true  
});
```

```
obj.index = 1;  
console.log(obj.index); // >>> 1
```

Атрибуты свойств

```
Object.defineProperty(obj, 'index', {  
    configurable: false  
});  
  
Object.defineProperty(obj, 'index', {  
    enumerable: false  
});  
// >>> Uncaught TypeError: Cannot redefine property: index  
  
console.log(Object.getOwnPropertyDescriptor(obj, 'index'));  
// >>> Object {value: 1, writable: true, enumerable: true,  
    configurable: false}
```

Атрибуты свойств

```
var obj = {};  
  
Object.defineProperty(obj, {  
  index: {  
    get: function () {  
      return Number(this.name);  
    },  
    enumerable: true,  
    configurable: true  
  },  
  name: {  
    value: '0',  
    enumerable: true,  
    writable: true,  
    configurable: true  
  }  
});
```

Атрибуты свойств при создании объекта

```
var obj = Object.create(Object.prototype, {  
  index: {  
    get: function () {  
      return Number(this.name);  
    },  
    enumerable: true,  
    configurable: true  
  },  
  name: {  
    value: '0',  
    enumerable: true,  
    writable: true,  
    configurable: true  
  }  
});
```

JSON: сериализация

JSON — JavaScript Object Notation

```
var obj = {  
    index: 10,  
    isReady: true,  
    name: 'sample',  
    childrenIds: [0, 10],  
    data: {  
        parent: null,  
        name: undefined  
    }  
};  
  
var serialized = JSON.stringify(obj);  
console.log(serialized);  
// >>> '{"index":10,"isReady":true,"name":"sample",  
        "childrenIds":[0,10],"data":{"parent":null}}'
```

Создание массива

```
var i = 10,  
    arr0 = [1, 2, ['Veni', 'Vidi', 'Vici'], null, i + 1],  
    arr1 = [1, 2, , 4],  
    arr2 = new Array(),  
    arr3 = new Array(4),  
    arr4 = new Array('Veni', 'Vidi', 'Vici');  
  
console.log(arr1.length); // >>> 4  
console.log(arr1[2]); // >>> undefined  
arr1[10] = 10;  
console.log(arr1.length); // >>> 11  
console.log(arr1[9]); // >>> undefined  
console.log(arr2); // >>> []  
console.log(arr3); // >>> [undefined × 3]  
console.log(arr4); // >>> ["Veni", "Vidi", "Vici"]
```

Непосредственное изменение длины массива

```
var arr = [1, 2];

arr.length = 4;
console.log(arr); // >>> [1, 2, undefined × 2]
arr.length = 1;
console.log(arr); // >>> [1]
arr.length = 0;
console.log(arr); // >>> [0]
```


Взаимное преобразование массивов и строк

```
var arr0 = [1, 2, 3],  
    str0 = arr0.join(),  
    str1 = arr0.join(' / '),  
    arr1 = str0.split(),  
    arr2 = str0.split(',');  
  
console.log(str0); // >>> "1,2,3"  
console.log(str1); // >>> "1 / 2 / 3"  
console.log(arr1); // >>> ["1,2,3"]  
console.log(arr2); // >>> ["1", "2", "3"]
```

Порядок элементов массива

```
var arr0 = ['2', '0', '1', '10'];

arr0.reverse();
console.log(arr0); // >>> ["10", "1", "0", "2"]

arr0.sort();
console.log(arr0); // >>> ["0", "1", "10", "2"]

arr0.sort(function (a, b) {
    return a - b;
});
console.log(arr0); // >>> ["0", "1", "2", "10"]
```

Объединение массивов

```
var arr0 = [0, 1, 2],  
    arr1 = ['Veni', 'Vidi', 'Vici'],  
    arr2 = ['Citius', 'Altius', 'Fortius'],  
    arr3 = arr0.concat(arr1),  
    arr4 = arr1.concat(arr0, arr2),  
    n = 4,  
    arr5 = arr0.concat(n);  
  
console.log(arr3);  
// >>> [0, 1, 2, "Veni", "Vidi", "Vici"]  
  
console.log(arr4);  
// >>> ["Veni", "Vidi", "Vici", 0, 1, 2, "Citius",  
        "Altius", "Fortius"]  
  
console.log(arr5); // >>> [0, 1, 2, 4]
```

Выделение фрагмента массива

```
var arr0 = [0, 1, 2, 3, 4, 5],  
    arr1 = arr0.slice(1, 3),  
    arr2 = arr0.slice(2),  
    arr3 = arr0.slice(1, -2),  
    arr4 = arr0.slice(-2, -1),  
    arr5 = arr0.slice(-2, -3),  
    arr6 = arr0.slice(3, 2),  
    arr7 = arr0.slice(3, 3);
```

```
console.log(arr1); // >>> [1, 2]  
console.log(arr2); // >>> [2, 3, 4, 5]  
console.log(arr3); // >>> [1, 2, 3]  
console.log(arr4); // >>> [4]  
console.log(arr5); // >>> []  
console.log(arr6); // >>> []  
console.log(arr7); // >>> []
```

Добавление и удаление элементов

```
var arr0 = [0, 1, 2, 3, 4, 5, 6, 7],  
    arr1 = arr0.splice(3);
```

```
console.log(arr0); // >>> [0, 1, 2]  
console.log(arr1); // >>> [3, 4, 5, 6, 7]
```

```
var arr2 = arr1.splice(2, 1);  
console.log(arr2); // >>> [5]  
console.log(arr1); // >>> [3, 4, 6, 7]
```

```
var arr3 = arr1.splice(2, 0, 8);  
console.log(arr3); // >>> []  
console.log(arr1); // >>> [3, 4, 8, 6, 7]
```

```
var arr4 = arr1.splice(2, 1, 9, '10');  
console.log(arr4); // >>> [8]  
console.log(arr1); // >>> [3, 4, 9, "10", 6, 7]
```

Обход элементов

```
var arr = ['Veni', 'Vidi', 'Vici'];

arr.forEach(function (item, index, array) {
    console.log(item, index, array.length);
}, this);

/*
>>> Veni 0 3
      Vidi 1 3
      Vici 2 3
*/
```

Создание нового массива на основе старого

```
var arr0 = ['Veni', 'Vidi', 'Vici'],
    arr1 = arr0.map(function (item, index, array) {
        return (index + 1) + '/' + array.length + ' ' +
            + item.toUpperCase();
    }, this),
    arr2 = [0, 1, 2, 3, 4],
    arr3 = arr2.filter(function (item, index, array) {
        return (item % 3 == 0 || index > 3);
    }, this);

console.log(arr1); // ["1/3 VENI", "2/3 VIDI", "3/3 VICI"]
console.log(arr3); // [0, 3, 4]
```

Проверка массива

```
var arr0 = [0, 2, 4],  
    arr1 = [1, 3, 6],  
    bool0 = arr0.some(function (item, index, array) {  
        return item % 2 == 0;  
    }, this),  
    bool1 = arr0.every(function (item, index, array) {  
        return item % 2 == 0;  
    }, this),  
    bool2 = arr1.some(function (item, index, array) {  
        return item % 4 == 0;  
    }, this),  
    bool3 = arr1.every(function (item, index, array) {  
        return item % 3 == 0;  
    }, this);
```

```
console.log(bool0); // true  
console.log(bool1); // true  
console.log(bool2); // false  
console.log(bool3); // false
```


Свертка

```
var arr = [0, 2, 4, 10],
    sum0 = arr.reduce(function (prevValue, item, index, array) {
        return prevValue + item;
    }, 1),
    diff0 = arr.reduceRight(function (prevValue, item, index, array) {
        return prevValue - item;
    }, 50),
    sum1 = arr.reduce(function (prevValue, item, index, array) {
        return prevValue + item;
    }),
    diff1 = arr.reduceRight(function (prevValue, item, index, array) {
        return prevValue - item;
    });

console.log(sum0); // >>> 17
console.log(diff0); // >>> 34
console.log(sum1); // >>> 16
console.log(diff1); // >>> 44
```

Получение позиции элемента

```
var arr = [0, 2, 10, 4, 10, 20, 10];

console.log(arr.indexOf(2)); // >>> 1
console.log(arr.indexOf(10)); // >>> 2
console.log(arr.lastIndexOf(10)); // >>> 6
console.log(arr.indexOf(10, 2)); // >>> 2
console.log(arr.indexOf(10, 3)); // >>> 4
console.log(arr.indexOf(10, -1)); // >>> 6
console.log(arr.indexOf(10, -4)); // >>> 4
console.log(arr.lastIndexOf(10, 3)); // >>> 2
console.log(arr.lastIndexOf(10, 1)); // >>> -1
console.log(arr.lastIndexOf(10, -1)); // >>> 6
console.log(arr.lastIndexOf(10, -4)); // >>> 2
console.log(arr.indexOf(10, 10)); // >>> -1
console.log(arr.lastIndexOf(10, -10)); // >>> -1
```

Полезные ресурсы

- https://developer.mozilla.org/ru/docs/Web/JavaScript/Reference/Global_Objects/Error
- https://developer.mozilla.org/ru/docs/Web/JavaScript/Reference/Global_Objects/JSON