Functions

Mark Zamoyta

markzamoyta@gmail.com

Overview

Function Expressions

Constructor Functions

The "this" Keyword

Calling Functions (call and apply)

Closures

IIFE's

Naming Function Expressions

```
var hireEmployee = function (name) {
};
console.log(typeof hireEmployee);
```

Question

What shows in the console?

Answer

`function`

```
var hireEmployee = function (name) {
    throw ('Error');
};
hireEmployee('JJ');
```

Question

What shows in the console?

Answer

```
Uncaught Error -
 hireEmployee
```

```
var hireEmployee = function (name) {
    throw ('Error');
};
var action = hireEmployee;
action('JJ');
```

Question

What shows in the console?

Answer

```
Uncaught Error -
 hireEmployee
```

```
var hireEmployee = function myHireEmployeeFn(name) {
    throw ('Error');
};
var action = hireEmployee;
action('JJ');
```

Q What shows in the console?

A Uncaught Error - myHireEmployeeFn

Constructor Functions

```
console.log(typeof Object);
```

Question

What shows in the console?

Answer

`function`

```
var Employee = function (name) {
    this.name = name;
};
var newEmployee = new Employee('JJ');
console.log(typeof newEmployee);
```

Question

What shows in the console?

Answer

```
object
```

```
var Employee = function (name) {
    this.name = name;
};
var newEmployee = new Employee('JJ');
console.log(newEmployee.name);
```

Question

What shows in the console?

Answer

JJ

```
var Employee = function (name, boss) {
    this.name = name;
    this.boss = boss;
};
var newEmployee = new Employee('JJ', 'JD Hogg');
console.log(newEmployee.boss);
```

Question

What shows in the console?

Answer

JD Hogg

```
var Employee = function (name) {
    this.name = name;
};
var e1 = new Employee('JJ');
var e2 = new Employee('JV');
console.log(e1 === e2);
```

Question

What shows in the console?

Answer

`false`

```
var Employee = function (name) {
    this.name = name;
};
var e1 = new Employee('JJ');
var e2 = new Employee('JV');
console.log(e1.__proto__ === e2.__proto__);
```

Question

What shows in the console?

Answer

`true`

```
var Employee = function (name) {
    this.name = name;
    this.giveRaise = function () {
    };
};
var e1 = new Employee('JJ');
var e2 = new Employee('JV');
console.log(e1.giveRaise === e2.giveRaise);
```

Question

What shows in the console?

Answer

`false`

```
var Employee = function (name) {
    this.name = name;
    this.giveRaise = function () {
    };
};
console.log(typeof Employee.prototype);
```

Question

What shows in the console?

Answer

object

```
var Employee = function (name) {
    this.name = name;
    this.giveRaise = function () {
    };
};
console.log(Employee.prototype === Object.prototype);
```

Q What shows in the console?

A `false`

```
var Employee = function (name) {
    this.name = name;
};
Employee.prototype.giveRaise = function () {
};
var e1 = new Employee('JJ');
var e2 = new Employee('JV');
console.log(e1.giveRaise === e2.giveRaise);
```

Q What shows in the console?

A `true`

```
var Employee = function (name) {
    this.name = name;
    this.salary = 50000;
};
Employee.prototype.giveRaise = function (raise) {
    this.salary += raise;
};
var e1 = new Employee('JJ');
var e2 = new Employee('JV');
e1.giveRaise(100000);
console.log(e1.salary);
console.log(e2.salary);
```

Q What shows in the console?

A

```
150000
50000
```

The "this" Keyword

```
console.log(typeof this);
```

Question

What shows in the console?

Answer

```
object
```

```
console.log(this === window);
```

Question

What shows in the console?

Answer

`true`

```
var name = 'Jeff';
console.log(this.name);
```

Question

What shows in the console?

Answer

Jeff

```
var updateSalary = function () {
    console.log(this === window);
};
updateSalary();
```

Question

What shows in the console?

Answer

`true`

```
var employee = {
    name: 'Jeff',
    updateSalary: function () {
        console.log(this);
    }
};
employee.updateSalary();
```

Question

What shows in the console?

Answer

```
Object {name: "Jeff"}
```

```
var employee = {
    name: 'Jeff',
    updateSalary: function () {
        var fn = function () {
            console.log(this);
        };
        fn();
    }
};
employee.updateSalary();
```

Question

What shows in the console?

Answer

```
Window {...}
```

```
var employee = {
    name: 'Jeff',
    updateSalary: function () {
        var fn = function () {
            console.log(this === window);
        };
        fn();
    }
};
employee.updateSalary();
```

Question

What shows in the console?

Answer

`true`

```
var Address = function (line1) {
    this.line1 = line1;
    console.log(this);
};
var addr = new Address('123 State St.');
```

Question

What shows in the console?

Answer

```
Address {line1: "123 St
ate St."}
```

```
var Address = function (line1) {
    this.line1 = line1;
};
Address.prototype.updateZipCode = function () {
    console.log(this);
};

var addr = new Address('123 State St.');
addr.updateZipCode();
```

Q What shows in the console?

A

```
Address {line1: "123 State
 St."}
```

Calling Functions

Using call() and apply()

```
var updateZipCode = function () {
    console.log(this);
};
updateZipCode();
```

Question

What shows in the console?

Answer

```
Window { ... }
```

```
var updateZipCode = function () {
    console.log(this);
};
updateZipCode.call({});
```

Question

What shows in the console?

Answer

```
Object {}
```

```
var updateZipCode = function () {
    console.log(this);
};
updateZipCode.call({ zip: '11787'});
```

Question

What shows in the console?

Answer

```
Object {zip: "11787"}
```

```
var updateZipCode = function () {
    console.log(this);
};
var zipCode = {
    zip: '11787'
};
updateZipCode.call(zipCode);
```

Question

What shows in the console?

Answer

```
Object {zip: "11787"}
```

```
var updateZipCode = function (newZip, country) {
    console.log(newZip + ' ' + country);
};
var zipCode = {
    zip: '11787'
};
updateZipCode.call(zipCode, '11888', 'us');
```

Q What shows in the console?

A 11888 us

```
var updateZipCode = function (newZip, country) {
    console.log(newZip + ' ' + country);
};
var zipCode = {
    zip: '11787'
};
updateZipCode.apply(zipCode, ['11888', 'us']);
```

Q What shows in the console?

A 11888 us

```
var updateZipCode = function (newZip, country) {
    console.log(newZip + ' ' + country);
};
var zipCode = {
    zip: '11787'
};
updateZipCode.apply(zipCode, '11888', 'us');
```

Q What shows in the console?

A

```
Uncaught TypeError: Functi
on.prototype.apply: Argume
nts list has wrong type
```

Closures

```
var salaryUpdater = function (salary) {
    var currentSalary = salary;
    var generator = function () {
        currentSalary *= 2;
        return currentSalary;
    };
    return generator;
};

var updateFn = salaryUpdater(50000);

console.log(updateFn());
```

Q What shows in the console?

A 100000

```
var salaryUpdater = function (salary) {
    var currentSalary = salary;
    var generator = function () {
        currentSalary += 10000;
        return currentSalary;
    };
    return generator;
};

var updateFn = salaryUpdater(50000);
updateFn();
console.log(updateFn());
```

Q What shows in the console?

A 70000

IIFEs

Immediately Invoked Function Expressions

```
(function () {
    console.log('Executed!');
})();
```

Question

What shows in the console?

Answer

```
Executed!
```

```
(function () {
    console.log('Executed!');
}());
```

Question

What shows in the console?

Answer

```
Executed!
```

```
(function () {
    var employeeName = 'Jill';
})();

console.log(employeeName);
```

Question

What shows in the console?

Answer

```
Uncaught ReferenceError
: employeeName is not

defined
```

```
var app = {};
(function (ns) {
    ns.name = 'None';
})(app);

console.log(app.name);
```

Question

What shows in the console?

Answer

None

```
var app = {};
var jQuery = {};

(function (ns, $) {
    ns.name = 'None';
    console.log($ === jQuery);
})(app, jQuery);
```

Question

What shows in the console?

Answer

```
true
```

```
var app = {};
var jQuery = {};

(function (ns, $, undefined) {
    ns.name = 'None';
    console.log(undefined);
})(app, jQuery);
```

Question

What shows in the console?

Answer

undefined

```
+function () {
    var employeeName = 'Jill';
    console.log('Executed!');
}();

console.log(employeeName);
```

Question

What shows in the console?

Answer

```
Executed!

Uncaught ReferenceError
: employeeName is not
defined
```

Recursion

```
var orgChart = {
    name: 'Michael', subordinates: [
        {
            name: 'Andy', subordinates: [
                {
                    name: 'Dwight', subordinates: []
                },
                {
                    name: 'Kevin', subordinates: []
                }
            ]
        }
    ]
};

var fn = function (topEmployee) {
    console.log(topEmployee.name);
    for (var i = 0; i < topEmployee.subordinates.lengt
h; i++)
        fn(topEmployee.subordinates[i]);
};

fn(orgChart);
```

Question

What shows in the console?

Answer

Michael

Andy

Dwight

Kevin

```
var orgChart = {
    name: 'Michael', subordinates: [
        {
            name: 'Andy', subordinates: [
                {
                    name: 'Dwight', subordinates: []
                },
                {
                    name: 'Kevin', subordinates: []
                }
            ]
        }]
};
var fn = function (topEmployee) {
    console.log(topEmployee.name);
    for (var i = 0; i < topEmployee.subordinates.lengt
h; i++)
        fn(topEmployee.subordinates[i]);
};
var fn2 = fn;
fn = null;
fn2(orgChart);
```

Question

What shows in the console?

Answer

Michael

Uncaught TypeError: fn is not a function

```
var orgChart = {
    name: 'Michael', subordinates: [
        {
            name: 'Andy', subordinates: [
                {
                    name: 'Dwight', subordinates: []
                },
                {
                    name: 'Kevin', subordinates: []
                }
            ]
        }]
};
var fn = function showAllEmployees (topEmployee) {
    console.log(topEmployee.name);
    for (var i = 0; i < topEmployee.subordinates.lengt
h; i++)
        showAllEmployees(topEmployee.subordinates[i]);
};
var fn2 = fn;
fn = null;
fn2(orgChart);
```

Question

What shows in the console?

Answer

Michael

Andy

Dwight

Kevin

Summary

- Function Expressions
- The "this" Keyword
- Calling Functions Using Call and Apply
- Closures
- IIFE's
- Recursion