

A solid yellow square background.

JS

Scoping

```
// Beispiel 1  
var foo = 42;  
function doSomething() {  
    return foo;  
}  
doSomething(); // returns ?
```

```
// Beispiel 2  
function test() {  
    var foo = 1;  
}  
function test2() {  
    var bar = 2;  
}  
  
test();  
test2();  
  
foo + bar; // ?
```

```
// Beispiel 3  
function test() {  
    foo = 1;  
}  
function test2() {  
    bar = 2;  
}  
  
test();  
test2();  
  
foo + bar; // Warum?
```

```
// Beispiel 4  
var num = 2;  
function beispiel4() {  
    this.num = 4;  
}  
beispiel4();  
console.log(num);
```

```
// Beispiel 5  
var num = 2;  
function Beispiel5() {  
    this.num = 4;  
}  
new Beispiel5();  
console.log(num);
```

```
// Beispiel 6  
function beispiel6(value2) {  
    console.log(this.value1 + value2);  
}  
  
var scope = {  
    value1: 42  
};  
  
beispiel6.bind(scope)(23);  
beispiel6.apply(scope, [23]);  
beispiel6.call(scope, 23);
```



```
// Aufgabe Scopes  
// Implementiere Funktion sum  
sum(21)(2); // should return 23  
sum(11)(31); // should return 42
```

```
// Lösung: Aufgabe Scopes  
function sum(value1) {  
    return function(value2) {  
        return value1 + value2;  
    }  
}
```

functions

// Beispiel 1

```
var beispiel1 = function(bar) {  
    return 'beispiel1';  
};
```

// Beispiel 2

```
function beispiel2(bar) {  
    return 'beispiel2';  
}
```

```
// Beispiel 3  
// Welche der vorherigen Beispiele kann  
// ich folgendermaßen  
// an die folgende Funktion übergeben?  
function executeFunction(func) {  
    return func();  
}  
executeFunction(beispiel1); // Was passiert?  
executeFunction(beispiel2); // Was passiert?
```

```
// Beispiel 4  
(function foo() {  
    var foobar = 42;  
}) ();
```

```
// Beispiel 5  
sum(1, 2); // returns ?  
function sum(val1, val2) {  
    return val1 + val2;  
}
```

```
// Beispiel 6  
sum(1, 2); // returns ?  
var sum = function(val1, val2) {  
    return val1 + val2;  
};
```


// Beispiel 7

```
function Guy() {  
    this.isNinja = false;  
    this.weapons = [];  
}
```

```
function Ninja() {  
    this.isNinja = true;  
    this.weapons = [];  
}
```

```
Ninja.prototype = Guy.prototype;
```

```
Guy.prototype.train = function() {  
    this.isNinja = true;  
};
```

```
Guy.prototype.addWeapon = function(weapon) {  
    this.weapons.push(weapon);  
    return this.weapons;  
};
```

```
var julian = new Guy();  
julian.train();
```

```
var bela = new Ninja();
```

```
bela.addWeapon('javascript');  
julian.addWeapon('scala');
```

chaining

```
// Was macht Array.pop?  
var foo = [1, 2, 3];  
var num = foo.pop(); // 3
```

```
// Aufgabe 1  
// Implementiere die folgende Funktion  
foobar.pop() (); // should return 'foo'
```

```
// Beispiel 1
// $http von angular.js
$http.get('/foo.json').success(function(data) {
    console.log(data);
});
```

```
// Beispiel 2  
// expect.js  
expect(window).not.to.be.an(Image);
```

```
// Aufgabe 2  
// Implementiere die folgende Funktion  
expect(true).toBe(false); // should return false  
expect(true).toBe(true); // should return true
```

prototype


```
// Beispiel 1  
var obj = new Object();  
obj.foo = 42;  
obj instanceof Object; // ?
```

```
// Beispiel 2  
var obj = { foo: 42 };  
obj instanceof Object; // ?
```

// Beispiel 3

var obj = **function**() {};

obj **instanceof** Object; *// ?*

obj **instanceof** Function; *// ?*

```
// Beispiel 4  
var arr = [1, 2, 3];  
arr instanceof Object; // ?  
arr instanceof Array; // ?
```

// Beispiel 5

Object.**prototype**.isPrototypeOf(Array) ; // ?

```
// Beispiel 6  
Array.prototype.isPrototypeOf(Object); // ?
```

```
// Beispiel 7  
var str = 'test';  
String.prototype.isPrototypeOf(str); // ?
```

// Beispiel 8

var str = **new** String('test');

String.**prototype**.isPrototypeOf(str); // ?

// Beispiel 9

```
function Guy() {  
    this.addLanguages = function(lang) {  
        this.languages.push(lang);  
    }  
}
```

```
Guy.prototype.languages = [];
```

```
new Guy().addLanguages('java');
```

```
new Guy().addLanguages('scala');
```

```
var awesomeGuy = new Guy();
```

```
console.log(awesomeGuy.languages);
```

// Beispiel 10

```
function Guy() {  
    this.addLanguages = function(lang) {  
        this.languages.push(lang);  
    }  
    this.languages = [];  
}
```

```
new Guy().addLanguages('java');  
new Guy().addLanguages('scala');
```

```
var awesomeGuy = new Guy();  
console.log(awesomeGuy.languages);
```

```
// Beispiel 11  
Number.prototype.isTen = function() {  
    console.log(+this.toString() === 10);  
}
```