JavaScript Extras: Scopes & Closures

Olivier Liechti & Simon Oulevay
COMEM Web Services 2016

heig-vd
Haute Ecole d'Ingénierie et de Gestion
du Canton de Vaud

JavaScript 101
Extras

heig-vd
Haute Ecole d'Ingénierie et de Gestion
du Canton de Vaud

How do functions, scope and closures work in JavaScript?

JavaScript 101
Extras

heig-vd
Haute Ecole d'Ingénierie et de Gestion
du Canton de Vaud

#1 There are 2 scopes in JavaScript (before ES6)

```
var globalVariable = "pollution...";
var f = function() {
  var localVariable = "good";
  globalVariable2 = "bad"; // "use strict" prevents this type of error
};
```

#2 Functions can be nested. This creates a scope chain.

```
var vInGlobalScope;
var f1 = function() {
  var vInScope1;
  var f2 = function() {
    var vInScope2;
    var f3 = function() {
      var vInScope3;
    };
  };
};
```

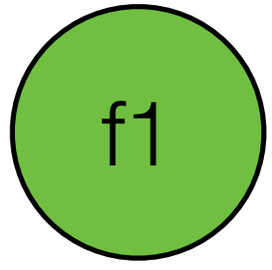
JavaScript 101
Extras

heig-vd
Haute Ecole d'Ingénierie et de Gestion
du Canton de Vaud

#3 Every function is an object

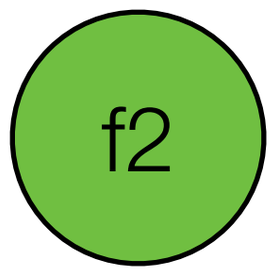
```
var f1 = function() {
  console.log("hello");
};

f1();
```




```
function f2() {
  console.log("hello");
};

f2();
```




```
var o = {
  foo: function() {}
};
var f3 = o.foo;

f3();
```

heig-vd
Haute Ecole d'Ingénierie et de Gestion
du Canton de Vaud

#4 Functions can return functions

```
function createMultiplier(factor) {
  return function(value) {
    return value * factor;
  };
}

var multiplyByThree = createMultiplier(3);

multiplyByThree(15); // 45
```

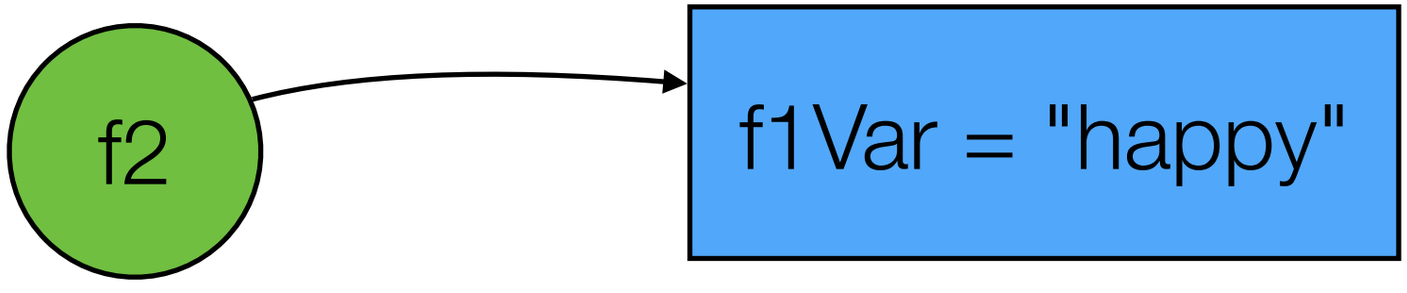
heig-vd
Haute Ecole d'Ingénierie et de Gestion
du Canton de Vaud

#5 A function which access a variable from an outer scope creates a **closure**. It keeps a reference to the outer variables forever.

```
var gVar = "world";

var f1 = function() {
  var f1Var = "happy";
  return function() {
    var f2Var = "hello"
    console.log(f1Var + " " + f2Var + " " + gVar);
  };
};

var f2 = f1()
f2();
```



heig-vd
Haute Ecole d'Ingénierie et de Gestion
du Canton de Vaud

#5 Beware of functions created in loops!

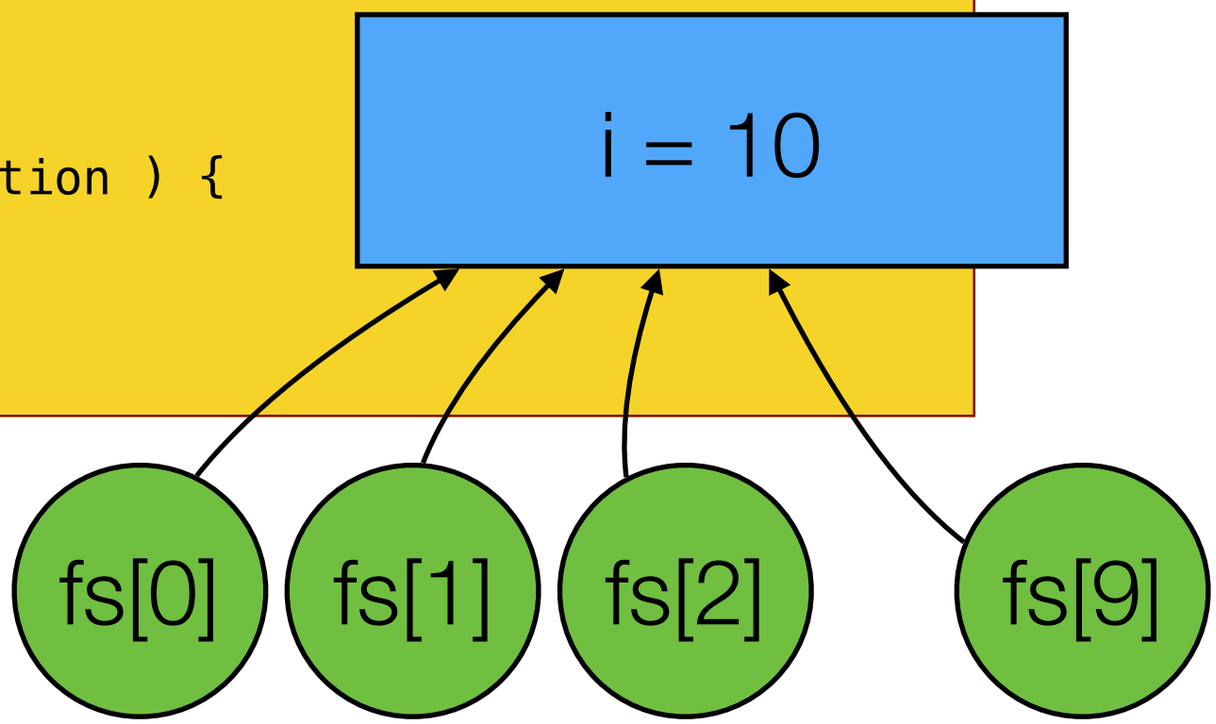
```
var generateFunctions = function() {
    var generatedFunctions = [];
    for (var i = 0; i < 10; i++) {
        var g = function() {
            console.log(i);
        };
        generatedFunctions.push(g);
    }
    return generatedFunctions;

};

var fs = generateFunctions();

fs.forEach(function( currentFunction ) {
    currentFunction();
});
```

closure



heig-vd
Haute Ecole d'Ingénierie et de Gestion
du Canton de Vaud

#6 Functions created in loops: capture variables

```
var generateFunctions = function() {
    var generatedFunctions = [];
    for (var i=0; i<10; i++) {
        var g = function(captureMe) {
            var captured = captureMe;
            return function() {
                console.log(captured);
            };
        };
        generatedFunctions.push(g(i));
    }
    return generatedFunctions;

};

var fs = generateFunctions();

fs.forEach(function( currentFunction ) {
    currentFunction();
});
```