

# ES6, Harmony, and JS++

Andrea Giammarchi

# Why ?

- JS developers love JS
- non-JS developers don't get JS: ES4 failed
- the most used programming language
- it's everywhere and highly addictive
- everybody wants to add something
- we all need and like sugar
- we all want more control as well
- well, we need to harmonize our ideas ...

# how to test

- Firefox Nightly ( aka: Minefield )

```
<!doctype html>
<html>
  <head>
    <title>JS5 and Harmony Quick Overview</title>
    <script>
      "Proxy" in this ?
      (function($){
        var
          E = $.documentElement,
          s = $.createElement("script")
          ;
        E.insertBefore(s, E.firstChild).type =
          "application/javascript;version=1.8"
          ;
        s.src = "harmony.js";
      }(document)) :
      alert("no harmony? no JS party")
      ;
    </script>
  </head>
  <body>
  </body>
</html>
```

# block scoped bindings

- “blocks” or inner/inline scope

# let

```
(function /* let */ () {"use strict"; // it will not really matter

    i; // not a reference error but undefined

    if (false) {
        // since it is defined here
        // even if never executed
        for (var i = 0; i);
    } else {
        for (let j = 1; j--;) j; // 0
        typeof j; // undefined
    }

})();
```

# const

```
(function /* const */ () {"use strict";

    const i;    // nope, the moment we need
    i = 1;     // a constant, we need to know
               // which value as well

    try {      // nope, a constant can be
               // defined only once
               // even try catch may fail
        //const i = 1;
    } catch(e) {
        "redeclaration of constant i";
    }

    const j = 1;
    ++j;
    j;         // 1

})();

typeof j;    // undefined
```

# scoped function

```
(function /* let + function */ () {"use strict";  
  
    for(let // block scoped declaration  
        i = 1,  
        fn = function () {  
            // reference only during this loop  
            return --i;  
        };  
        fn();  
    );  
  
    typeof i; // undefined  
    typeof fn; // undefined  
  
})();
```

# inline block scope

```
// inline block scope
{
  let a = 1;
  a; // 1
  var b = 2;
}
typeof a; // undefined
b; // 2
```

# Destructuring

- <http://wiki.ecmascript.org/doku.php?id=harmony:destructuring>

```
//      Array
{
  let [a, b] = [1, 2];
  a;    // 1
  b;    // 2

  [a, b] = [b, a];
  a;    // 2
  b;    // 1

  // proposal: coming soon

  let [c, ...d] = [1, 2, 3];
  c;    // 1
  d;    // [2, 3]
}
```

# Comprehensions

- [http://mdn.beonex.com/en/Core\\_JavaScript\\_1.5\\_Guide/Working\\_with\\_Arrays](http://mdn.beonex.com/en/Core_JavaScript_1.5_Guide/Working_with_Arrays)

```
{
  [odd for each (odd in [1, 2, 3, 4]) if (odd % 2)];
  // [1, 3]

  let slice = function (stack, start, end) {
    return [i for (i in stack) if (
      (start || 0) <= i &&
      i < (end || list.length)
    )];
  };

  slice([1,2,3], 0, 2); // [0, 1]);

  // Objects too baby!!!
  let
    me = {name: "Andrea", age: 32}
    ;
  [value for each (value in me)].join(" is ");
  // Andrea is 32
}
```

# Iterators

- [https://developer.mozilla.org/en/Core\\_JavaScript\\_1.5\\_Guide/Iterators\\_and\\_Generators](https://developer.mozilla.org/en/Core_JavaScript_1.5_Guide/Iterators_and_Generators)

```
{
  for (let [key, value] in Iterator({name: "Andrea", age: 32})) {
    [key, value];    // name, Andrea
                    // age, 32
  }
}
```

# Generators

- [http://mdn.beonex.com/en/New\\_in\\_JavaScript\\_1.8](http://mdn.beonex.com/en/New_in_JavaScript_1.8)

```
{
  let index = (i for (i in [1, 2, 3]));
  try {
    while(true) {
      index.next(); // 0, 1, 2
    }
  } catch(e if e instanceof StopIteration) {
    // something to do ?
  } catch(e) {
    // other Exceptions
  }
}
```

# Parameters

- default values, rest parameters, spread

```
{  
  // proposal: coming soon  
  
  let pow = function (num, exp ||= 2) {  
    return Math.pow(num, exp);  
  };  
  // OR  
  let pow = function (num, exp ??= 2) {  
    return Math.pow(num, exp);  
  };  
  
  let concat = function (first, ...others) {  
    return [first].concat(others);  
  };  
  concat(1, 2, 3); // [1, 2, 3]  
  
  // spread, aka: simplified apply  
  let twoAndThree = [2, 3];  
  concat(1, ...twoAndThree);  
  // same as  
  concat.apply(null, [1].concat(twoAndThree));  
  
  let all = [1, 2, 3];  
  concat(...all); // still the same result  
}
```

# Proxy

- <http://wiki.ecmascript.org/doku.php?id=harmony:proxies>

```
{
  let
  creator = function (obj) {
    "use strict";
    var
      getOwnPropertyDescriptor = function (name) {
        return Object.getOwnPropertyDescriptor(obj, name);
      },
      proxy = {
        // basic must have
        getOwnPropertyDescriptor: function (name) {
          return Object.getOwnPropertyDescriptor(obj, name);
        },
        getPropertyDescriptor: function (name) {
          return Object.getPropertyDescriptor(obj, name);
        },
        getOwnPropertyNames: function () {
          return Object.getOwnPropertyNames(obj);
        },
        getPropertyNames: function () {
          return Object.getPropertyNames(obj);
        },
        defineProperty: function (key, value) {
          return Object.defineProperty(obj, key, value);
        },
        delete: function (key) {
          return delete obj[key];
        },
        fix: function () {
          return Object.isFrozen(obj) ?
            Object.getOwnPropertyNames(obj).map(
              getOwnPropertyDescriptor
            ) :
            undefined
        };
      };
  }
};
```

# Proxy

- ... optionally ...

```
proxy.has = function (name) {
    return name in obj;
};
proxy.hasOwnProperty = function (name) {
    return proxy.hasOwnProperty.call(obj, name);
};
proxy.get = function (receiver, name) {
    return obj[name];
};
proxy.set = function (receiver, name, value) {
    obj[name] = value;
    return true;
};
proxy.enumerate = function () {
    return [key for (key in obj)];
};
proxy.keys = function () {
    return Object.keys(obj);
};
return Proxy.create(proxy);
}
;
```

# Weak Maps

- hopefully not as described right now: [http://wiki.ecmascript.org/doku.php?id=harmony:weak\\_maps](http://wiki.ecmascript.org/doku.php?id=harmony:weak_maps)

```
const WeakMap = function () {
  var
    key = [],
    value = [],
    undefined

  ;
  return Object.create(WeakMap.prototype, {
    get: {
      value: function (keyObject) {
        var i = key.indexOf(keyObject);
        return -1 < i ? value[i] : undefined;
      }
    },
    set: {
      value: function (keyObject, valueObject) {
        var i = key.indexOf(keyObject);
        key[-1 < i ? i : i = key.length] = keyObject;
        value[i] = valueObject;
      }
    },
    del: {
      value: function (keyObject) {
        var i = key.indexOf(keyObject);
        if (-1 < i) {
          key.splice(i, 1);
          value.splice(i, 1);
        }
      }
    }
  });
};
```

# Questions?

- <http://webreflection.blogspot.com>