CoffeeScript

by sedina.oruc@callistaenterprise.se

Cadec 2013-01-16

Today's Agenda

- Flaws of JavaScript
- Problems addressed by CoffeeScript
- Integrating CoffeeScript into current build
- TDD approach with presence of CoffeeScript

The flaws of JavaScript

Global variables:

```
usersCount = 1;           // Global
var groupsCount = 2;      // Global

(function(){
  pagesCount = 3;         // Global
  var postsCount = 4;     // Local
})()
```

The `with` statement:

```
var foo = "bar";
with (obj)
    foo;                  // bar, unless o
bj.foo exists
    undefined;            // new global va
riable
```

The flaws of JavaScript

Reserved words:

```
const enum class
```

Equality comparisons:

```
""          ==    "0"          // false
0           ==    ""           // true
0           ==    "0"          // true
false       ==    "false"      // false
false       ==    "0"          // true
false       ==    undefined    // false
false       ==    null         // false
null        ==    undefined    // true
```

The flaws of JavaScript

Function definitions:

```
wem();
function wem() {}
```

Is `else` ever executed?

```
if (true) {
  function declaration() {
    return "first";
  }
} else {
  function declaration() {
    return "second";
  }
}
declaration();
```

What is CoffeeScript?

- A language that compiles down to readable JavaScript

- CoffeeScript == Maintainable JavaScript

- The good parts of JavaScript, with a syntax that does not hurt

- CoffeeScript is not abstracting JavaScript away as GWT does

CoffeeScript identifiers

- Indentation sensitive
- Everything is an expression
- String interpolation
- OOP support
- Better functional syntax
- All variables are local
- Intuitive equality check
- No with statement
- Emits readable JavaScript

Let's try some CoffeeScript

DEMO

Integrating CoffeeScript into current build

- CoffeeScript with Maven == `true`

TDD approach with CoffeeScript

- compiled CoffeeScript == JavaScript
- JS-test-framework(s)

Putting it all together

DEMO

Conclusions

- CoffeScript gives you this

  - Less boilerplate
  - Better readability
  - Easier maintainability
  - Abstraction over JavaScript flaws
  - Chance to become a better JavaScript developer

- You still need to know JavaScript

  - Debugging
  - Testing

Thank you for listening!