

CoffeeScript für Java Entwickler

Michael Heinrichs

canoo Engineering AG

0 == ''

0 === '0'

'' === '0' ?





THIS IS
JAVA SCRIPT
+

0 == ''

0 === '0'

'' !== '0'



```
speaker =  
  name: 'Michael Heinrichs'  
  company: 'Canoo Engineering AG'  
  web: 'blog.netopyr.com'  
  twitter: '@net0pyr'  
  description: 'UI Magician,  
               Active writer and speaker'
```



Inhalt

- JavaScript Fixes
- Syntactic Sugar und andere Goodies
- Installation und Nutzung

CoffeeScript

```
cube = (x) -> x * x * x
```

JavaScript

```
var cube;  
  
cube = function(x) {  
    return x * x * x;  
};
```

JavaScript Fixes



0 == ''

0 === '0'

'' !== '0'



0 == ''

0 !== ''

0 === '0'

0 !== '0'

'' != '0'

'' !== '0'



```
class Frog  
    sayHello: () -> alert('quak')
```

```
frog = new Frog();  
frog.sayHello();
```

```
class Frog  
    sayHello: () -> alert('quak')
```

```
frog = new Frog();  
frog.sayHello();
```



```
class Frog  
    sayHello: -> alert 'quak'
```

```
frog = new Frog  
frog.sayHello()
```



```
class Frog
  constructor: (newValue) ->
    this.color = newValue
```

```
frog = new Frog('pink')
alert(frog.color)
```



```
class Frog
  constructor: (newValue) ->
    @color = newValue
```

```
frog = new Frog('pink')
alert(frog.color)
```



```
class Frog
  constructor: (@color) ->
```

```
frog = new Frog('pink')
alert(frog.color)
```




```
class Frog
    sayHello: -> alert('quak')

class Kermit extends Frog
    sayHello: -> alert('hello')

kermit = new Kermit();
kermit.sayHello();
```

```
var pig = {  
  name: 'Piggy'  
  showName: function() {  
    alert(this.name);  
  }  
}
```

```
pig.showName()
```



```
var pig = {  
  name: 'Piggy'  
  showName: function() {  
    alert(this.name);  
  }  
}
```

```
document.querySelector('button').onclick = pig.showName;
```



```
pig =  
  name: 'Piggy'  
  showName: -> alert(this.name);
```

```
document.querySelector('button').onclick = pig.showName;
```

```
pig =  
  name: 'Piggy'  
  showName: => alert(this.name);
```

```
document.querySelector('button').onclick = pig.showName;
```



```
if (element !== null) {  
    # do something  
}
```



```
if (typeof element !== 'undefined' && element !== null) {  
    # do something  
}
```

```
if element?  
  # do something
```




```
foo = 42;
```

```
bar = -> alert('bar');
```

```
var foo, bar;
```

```
foo = 42;
```

```
bar = -> alert('bar');
```



```
(( ) ->  
  var foo, bar;  
  
  foo = 42;  
  
  bar = -> alert('bar');  
)();
```

Syntactic Sugar
und andere Goodies

```
fooBar = (x) -> {  
  if (x == 'foo') {  
    alert 'foo'  
  } else {  
    alert 'bar'  
  }  
}
```

```
fooBar = (x) -> {  
  if (x == 'foo') {  
    alert 'foo'  
  } else {  
    alert 'bar'  
  }  
}
```



```
fooBar = (x) -> {  
    if (x == 'foo') {  
        alert 'foo'  
    } else {  
        alert 'bar'  
    }  
}
```

```
fooBar = (x) ->  
    if (x == 'foo')  
        alert 'foo'  
    else  
        alert 'bar'
```



„Everything is an expression.“

– coffeescript.org




```
fooBar = if x == 'foo' then 'foo' else 'bar'
```

```
cubic = for i in [0..3]
  i * i * i
```

```
# cubic is [0, 1, 8, 27]
```



```
switch state
  when tired
    drinkCoffee()
  when hungry
    eatPizza()
  else code()
```

```
switch state
  when tired, stuck
    drinkCoffee()
  when hungry
    eatPizza()
  else code()
```

```
drink = (liquid = 'coffee') ->  
  alert 'Drinking #{liquid}'
```



```
if min < x < max then alert 'passt schon'
```

```
if element?  
  # do something
```



`x ?= 'default'`

`x = y ? 'default'`




```
street = person.address?.street
```

```
person.notify?()
```



```
numbers = [0, 1, 2, 3, 4]
```

```
slice1 = numbers[1..3]      # [1, 2]
```

```
slice2 = numbers[1..3]      # [1, 2, 3]
```



```
numbers = [0, 1, 2, 3, 4]
```

```
numbers[1..3] = [-1]      # [0, -1, 4]
```

CoffeeScript

JavaScript

`is``===``isnt``!==``not``!``and``&&``or``||``yes, on``true``no, off``false`

```
getNames = () ->  
  ['Snow', 'Jon']
```

```
[lastName, firstName] = getNames()
```



```
getSalutation = () ->  
  title: 'Lord Commander of the Night's Watch'  
  lastName: 'Snow'  
  firstName: 'Jon'
```

```
{lastName, firstName} = getSalutation()
```

Installation und Nutzung



```
> npm install -g coffee-script
```




```
> coffee
```

```
> coffee --compile my_file.coffee
```



```
> npm install --save-dev grunt-contrib-coffee
```



```
grunt.loadNpmTasks('grunt-contrib-coffee');

coffee: {
  compile: {
    files: {
      './public/my_file.js': './src/my_file.coffee'
    }
  }
}
```



```
> npm install --save-dev gulp-coffee
```





```
var coffee = require('gulp-coffee');

gulp.task('coffee', function() {
  gulp.src('./src/my_file.coffee')
    .pipe(coffee().on('error', gutil.log))
    .pipe(gulp.dest('./public'))
})
```



```
<plugin>  
  <groupId>com.theoryinpractise</groupId>  
  <artifactId>coffee-maven-plugin</artifactId>  
  <version>1.4.7</version>  
</plugin>
```



```
apply plugin: 'coffeescript-base'

task compileCoffee(type: CoffeeScriptCompile) {
    source fileTree('src/main/coffee')
    destinationDir file('build/js')
}
```



```
<dependency>  
  <groupId>com.netopyr.coffee4java</groupId>  
  <artifactId>coffee4java</artifactId>  
  <version>1.0.0</version>  
</dependency>
```




```
ScriptEngineManager manager = new ScriptEngineManager();  
  
ScriptEngine engine =  
    manager.getEngineByName("CoffeeScript");  
  
engine.eval("print 'Hello World!'");
```

CoffeeScript: das bessere JavaScript

