

5.– 8. September 2011
in Nürnberg



Herbstcampus

Wissenstransfer
par excellence

Kaffee zum Dessert?

CoffeeScript – nur die guten Dinge aus JavaScript

Andreas Schubert

MATHEMA Software GmbH

„It’s just JavaScript“

"Just the good parts from
JavaScript"

<http://jashkenas.github.com/coffee-script/>

<http://pragprog.com/book/tbcoffee/coffeescript>

"Probleme" mit JavaScript

- JavaScript ist eine funktionale Sprache
- JavaScript ist eine objektorientierte Sprache, basierend auf Prototypen
- JavaScript ist sehr dynamisch und deutlich näher an Lisp als an C/Java, aber hat eine C/Java ähnliche Syntax
- Selbst der Name ist verwirrend, da es nichts mit Java zu tun hat

Appetithäppchen...



JavaScript:

```
if (typeof elvis !== "undefined" && elvis !== null) {  
    alert("I knew it!");  
}
```


JavaScript:

```
if (typeof elvis !== "undefined" && elvis !== null) {  
    alert("I knew it!");  
}
```

CoffeeScript:

```
alert "I knew it!" if elvis?
```



JavaScript:

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

JavaScript:

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

CoffeeScript:

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

CoffeScript-Source

installation & usage

installation & usage

```
$ brew install npm

$ npm install coffee-script

$ coffee
coffee> console.log "Hello World"
Hello World

$ coffee -e "... "

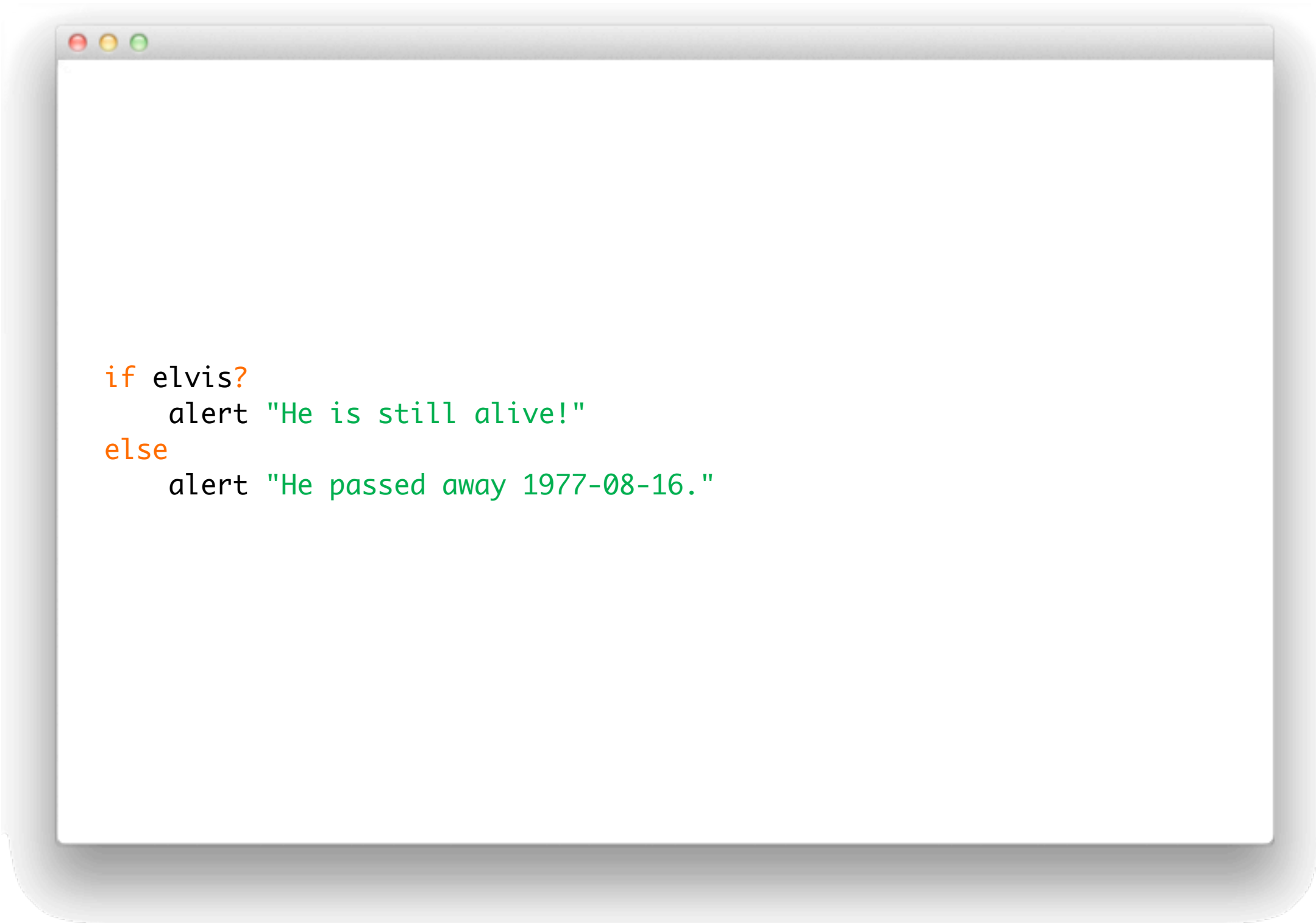
$ coffee -c ...

$ coffee -o ...

$ coffee -w ...
```


"Hello Python" - whitespace

"Hello Python" - whitespace



```
if elvis?  
    alert "He is still alive!"  
else  
    alert "He passed away 1977-08-16."
```

functions

functions

```
square = (x) -> x*x  
  
cube = (x) -> square(x)*x  
  
area = (x,y) -> x*y  
  
fill = (container, liquid="coffee") ->  
    "Filling the #{container} with #{liquid}..."  
  
noop = ->
```

arrays & objects

arrays & objects

```
song = ["do", "re", "mi", "fa", "so"]

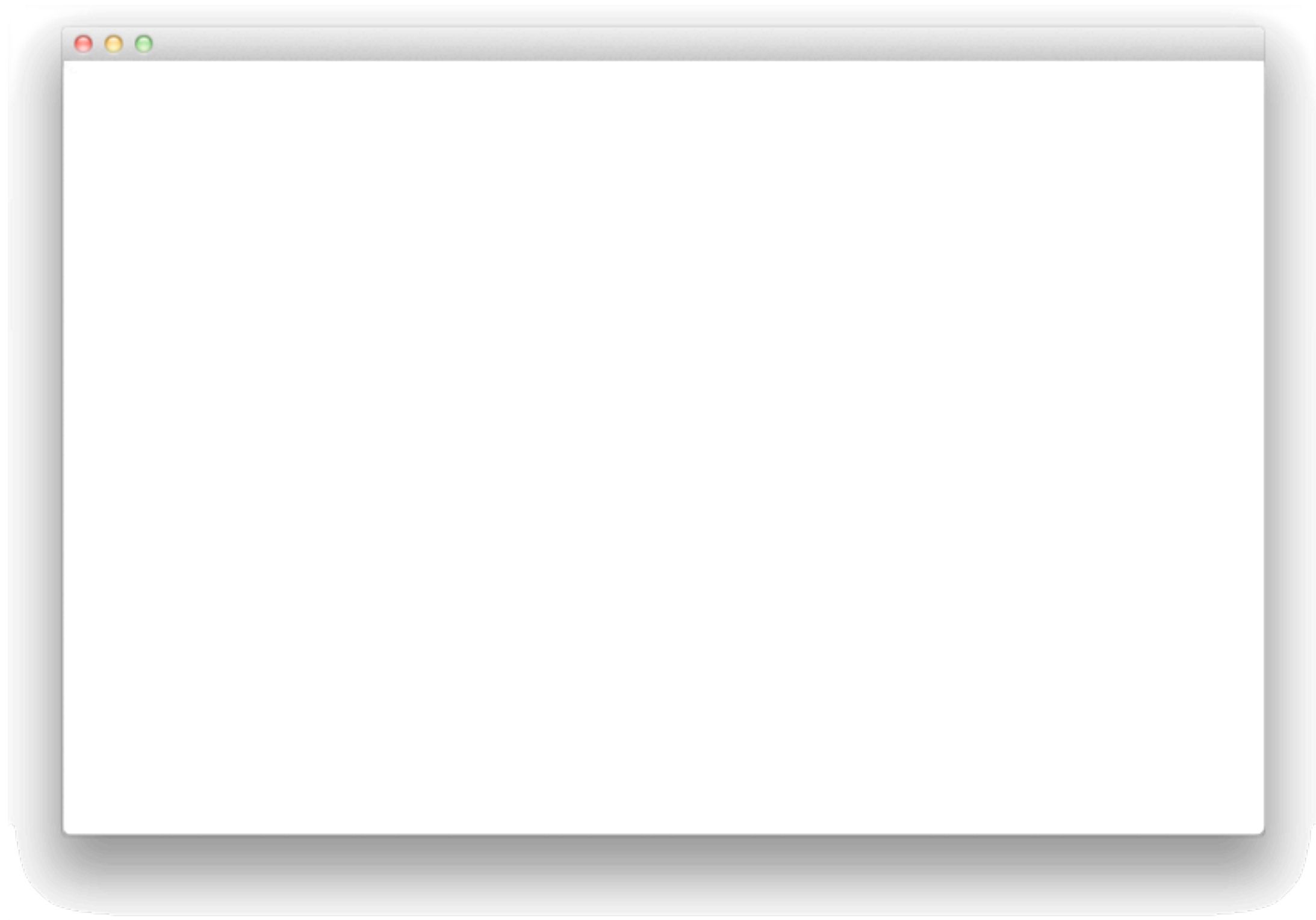
singers = {Jagger: "Rock", Elvis: "Roll"}

bitlist = [
  1, 0, 1
  0, 0, 1
  1, 1, 0
]

kids =
  brother:
    name: "Max"
    age: 11
  sister:
    name: "Ida"
    age: 9
```

variable safety & lexical scope

variable safety & lexical scope



variable safety & lexical scope

CoffeeScript:

```
outer = 1
changeNumbers = ->
  inner = -1
  outer = 10
inner = changeNumbers()
```


variable safety & lexical scope

CoffeeScript:

```
outer = 1
changeNumbers = ->
  inner = -1
  outer = 10
inner = changeNumbers()
```

JavaScript:

```
var changeNumbers, inner, outer;
outer = 1;
changeNumbers = function() {
  var inner;
  inner = -1;
  return outer = 10;
};
inner = changeNumbers();
```

aliases

CoffeeScript	JavaScript
<code>==</code>	<code>===</code>
<code>!=</code>	<code>!==</code>
<code>is</code>	<code>===</code>
<code>isnt</code>	<code>!==</code>
<code>yes</code>	<code>true</code>
<code>on</code>	<code>true</code>
<code>no</code>	<code>false</code>
<code>off</code>	<code>false</code>
<code>and</code>	<code>&&</code>
<code>or</code>	<code> </code>
<code>not</code>	<code>!</code>
<code>unless</code>	<code>if !</code>
<code>@property</code>	<code>this.property</code>

if, else, unless & conditional assignment

if, else, unless & conditional assignment

```
mood = greatlyImproved if singing

if happy and knowsIt
  clapsHands()
  chaChaCha()
else
  showIt()

date = if friday then sue else jill

options or= defaults
```

splats...

splats...

```
gold = silver = rest = "unknown"

awardMedals = (first, second, others...) ->
  gold = first
  silver = second
  rest = others

contenders = [
  "Michael Phelps"
  "Liu Xiang"
  "Yao Ming"
  "Allyson Felix"
  "Shawn Johnson"
  "Roman Sebrle"
  "Guo Jingjing"
  "Tyson Gay"
  "Asafa Powell"
  "Usain Bolt"
]

awardMedals contenders...

alert "Gold: " + gold
alert "Silver: " + silver
alert "The Field: " + rest
```

loops & comprehensions

loops & comprehensions

```
countdown = (num for num in [10..1])

shortNames = (name for name in list when name.length < 5)

evens = (x for x in [0..10] by 2)

yearsOld = max: 10, ida: 9, tim: 11
ages = for child, age of yearsOld
  "#{child} is #{age}"

if this.studyingEconomics
  buy() while supply > demand
  sell() until supply > demand

num = 6
lyrics = while num -= 1
  "#{num} little monkeys, jumping on the bed.
  One fell out and bumped his head."
```


slicing & splicing

slicing & splicing

```
numbers = [0, 1, 2, 3, 4, 5, 6, 7, 8, 9]
```

```
alert numbers[3..6]
```

```
alert numbers[3...6]
```

```
numbers[3..6] = [-3, -4, -5, -6]
```

```
alert numbers
```

(nearly) everything is an expression

(nearly) everything is an expression

```
grade = (student) ->
  if student.excellentWork
    "A+"
  else if student.okayStuff
    if student.triedHard then "B" else "B-"
  else
    "C"
```

```
globals = (name for name of window)[0...10]
```

the existential operator

the existential operator

```
alert "I knew it!" if elvis?  
  
speed ?= 75  
  
footprints = yeti ? "bear"  
  
zip = lottery.drawWinner?().address?.zipcode
```

classes, inheritance & super

classes, inheritance & super

```
class Animal
  constructor: (@name) ->

  move: (meters) ->
    console.log "#{@name} moved #{meters} meters"

class Snake extends Animal
  move: ->
    console.log "#{@name} is slithering..."
    super 5

class Horse extends Animal
  move: ->
    console.log "#{@name} is galloping..."
    super 45

sam = new Snake "Sammy the python"
tom = new Horse "Tommy the palomina"

sam.move()
tom.move()
```


function binding

function binding

```
Account = (customer, cart) ->  
  @customer = customer  
  @cart = cart  
  
$('.shopping_cart').bind 'click', (event) =>  
  @customer.purchase @cart
```

chained comparisons

chained comparisons

```
cholesterol = 127
```

```
healthy = 200 > cholesterol > 60
```

...und noch ein paar andere
Dinge...

Beispiel(e)

webserver

webserver

```
http = require 'http'

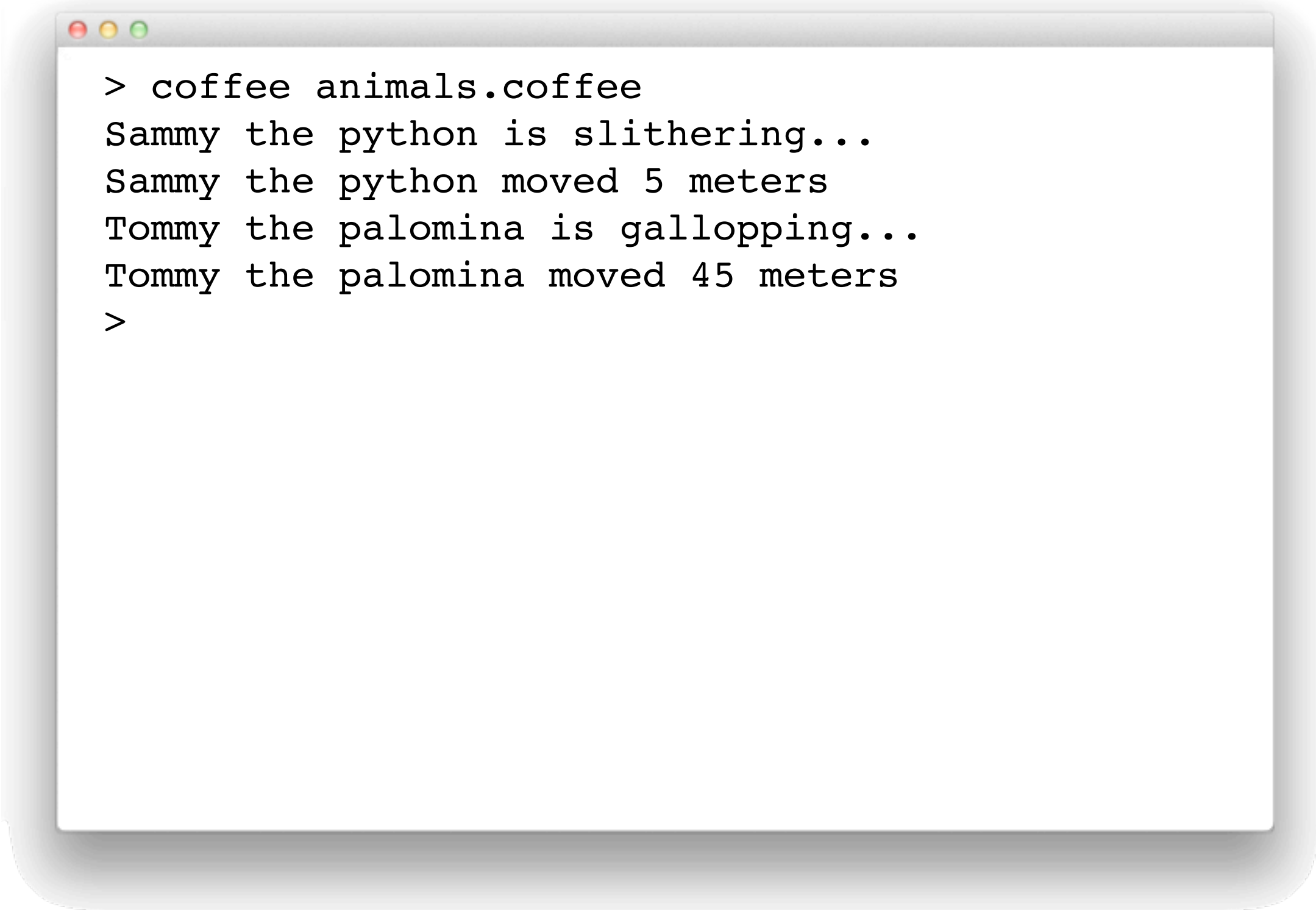
server = http.createServer (req, res) ->
  res.writeHead 200, 'Content-Type': 'text/plain'
  res.write 'Hello, World!'
  res.end()

server.listen 3000

console.log "Server running at http://localhost:3000/"
```


Sammy the Snake & Tommy the Palomina

Sammy the Snake & Tommy the Palomina



```
> coffee animals.coffee  
Sammy the python is slithering...  
Sammy the python moved 5 meters  
Tommy the palomina is gallopping...  
Tommy the palomina moved 45 meters  
>
```

jQuery

jQuery

```
formValues = (elem.value for elem in $(':input'))  
alert formValues
```

5.– 8. September 2011
in Nürnberg



Herbstcampus

Wissenstransfer
par excellence

Vielen Dank!

Andreas Schubert

andreas.schubert@mathema.de
MATHEMA Software GmbH