

Usage

-c, --compile
-i, --interactive
-o, --output [DIR]
-p, --print
-e, --eval
--nodejs

most commonly used parameters from
coffeescript.org

Functions

#Last expression value is return value
fill = (container, liquid = "coffee") ->
"Filling the #{container} with #{liquid}..."
internal variables override outer ones
message = 'Outer'
getMessage = -> message
overrideMessage = -> message = 'Inner'
overrideMessage() == 'Inner'
supports splats
returnAllArgs = (allargs...) -> allargs
returnAllArgs('first', 'second', 'third') ==
['first', 'second', 'third']
returnAllButFirst = (firstArg, rest...) -> rest
returnAllButFirst('first', 'second', 'third') ==
['second', 'third']
destructuring assignment
weatherReport = (location) -> [location, 22,
'Mostly sunny']
[city, temperature, forecast] = weatherReport 'London'
city == 'London'
temperature == 22

Higher Order

2 in [1..3] == true
#JavaScript-style filter
even = (a) -> a % 2 == 0
[1..6].filter even == [2, 4, 6]
CoffeeScript-style filter
odds = (x for x in [1..6] when not even x)
CoffeeScript-style map
twice = (a) -> a * 2

Higher Order (cont)

doubles = (twice x for x in [1..6])

Objects

```
meglomaniac = {}
beforeEach ->
  meglomaniac =
    mastermind: 'The Monarch'
    henchwoman: 'Dr Girlfriend'
    theBomb: true

# ? existence operator
meglomaniac.theBomb? == true
meglomaniac.theDetonator? == false
# properties can be added and deleted
meglomaniac.mastermind2 = 'Agent Smith'
delete meglomaniac.mastermind
# prototype to add to all projects
Circle = (radius) -> @radius = radius
'@' = 'this context' In CoffeeScript
```

Inheritance

running example from Muppets
class Muppet
 constructor: (@age, @hobby) ->
 answerNanny: -> "Everything's cool!"
class SwedishChef extends Muppet
 constructor: (age, hobby, @mood) ->
 super(age, hobby)
 cook: -> 'Mmmm soup!'
@swedishChef = new SwedishChef 3,
'cooking', 'chillin'
@swedishChef.cook() == 'Mmmm soup!'
#base object
@swedishChef.answerNanny() == "Everyt-
hing's cool!"
#instances to override class methods
gonzo = new Muppet 3, 'daredevil
performer'
gonzo.answerNanny = -> 'Hehehe!'

Hello World

```
echo "console.log 'Hello World'" > hello.c-  
offee  
coffee hello.coffee
```

Arrays

```
[1..5] == [1, 2, 3, 4, 5]  
[1...5] == [1, 2, 3, 4] # extra dot  
[3..1] == [3, 2, 1]  
fourNumberArray = [1, 2, 3, 4]  
fourNumberArray.push(5, 6)  
fourNumberArray == [1, 2, 3, 4, 5, 6]  
[1..10][3..5] == [4, 5, 6] # range slicing  
"my string"[0..1] == "my" # string slicing  
#iterate with hasOwnProperty check  
for own key, value of object  
copyOfArray = array.slice()
```

Array Reduction

Javascript style
total = (i, a) -> i + a
reduction = [1..3].reduce total == 6
Coffeescript-style
total = 0
sum = (a) -> total = total + a
sum x for x in [1..3]
total == 6

MDN Documentation:
<http://tinyurl.com/arreduce>

Credits

This cheat sheet is (mostly) based on work
of sleepyfox, which can be found here:
[https://github.com/sleepyfox/coffeescript--
koans](https://github.com/sleepyfox/coffeescript--koans)
CoffeeScript @github:
<http://jashkenas.github.com/coffee-script/>

Misc

Official Site: <http://coffeescript.org/>
Book: [http://pragprog.com/book/tbcoffee/co-
ffeescript](http://pragprog.com/book/tbcoffee/coffeescript)
Smooth CoffeeScript: [http://tiny.cc/smo-
othcs](http://tiny.cc/smo-othcs)
Jitter (compiler after modifying files):
<https://github.com/TrevorBurnham/jitter>
Codeschool Course:
[http://www.codeschool.com/courses/coffe-
escript](http://www.codeschool.com/courses/coffee-script)



By **Dimitrios Mistrionis**
(dimitrios)
cheatography.com/dimitrios/

Published 23rd December, 2011.
Last updated 12th May, 2016.
Page 1 of 2.

Sponsored by **ApolloPad.com**
Everyone has a novel in them. Finish
Yours!
<https://apollopadd.com>

Logo



C

By **Dimitrios Mistrionis**
(dimitrios)
cheatography.com/dimitrios/

Published 23rd December, 2011.
Last updated 12th May, 2016.
Page 2 of 2.

Sponsored by **ApolloPad.com**
Everyone has a novel in them. Finish
Yours!
<https://apollopadd.com>