CoffeeScript is a little language that compiles one-to-one into the equivalent JavaScript, and there is no interpretation at runtime. As one of the successors to JavaScript, CoffeeScript tries its best to output readable, pretty-printed and smooth-running JavaScript code, which works well in every JavaScript runtime.

See also the CoffeeScript website, which has a complete tutorial on CoffeeScript.

```
# CoffeeScript is a hipster language.
# It goes with the trends of many modern languages.
# So comments are like Ruby and Python, they use number symbols.

###
Block comments are like these, and they translate directly to '/ *'s and '* /'s
for the resulting JavaScript code.

You should understand most of JavaScript semantics
before continuing.
###

# Assignment:
number   = 42 #=> var number = 42;
opposite = true #=> var opposite = true;

# Conditions:
number = -42 if opposite #=> if(opposite) { number = -42; }

# Functions:
square = (x) -> x * x #=> var square = function(x) { return x * x; }

fill = (container, liquid = "coffee") ->
  "Filling the #{container} with #{liquid}..."
#=>var fill;
#
#fill = function(container, liquid) {
#  if (liquid == null) {
#    liquid = "coffee";
#  }
#  return "Filling the " + container + " with " + liquid + "...";
#};

# Ranges:
list = [1..5] #=> var list = [1, 2, 3, 4, 5];

# Objects:
math =
  root:   Math.sqrt
  square: square
  cube:   (x) -> x * square x
#=> var math = {
#    "root": Math.sqrt,
#    "square": square,
#    "cube": function(x) { return x * square(x); }
#   };

# Splats:
race = (winner, runners...) ->
```

```
  print winner, runners
#=>race = function() {
#    var runners, winner;
#   winner = arguments[0], runners = 2 <= arguments.length ? __slice.call(arguments, 1) : [];
#    return print(winner, runners);
#  };

# Existence:
alert "I knew it!" if elvis?
#=> if(typeof elvis !== "undefined" && elvis !== null) { alert("I knew it!"); }

# Array comprehensions:
cubes = (math.cube num for num in list)
#=>cubes = (function() {
#     var _i, _len, _results;
#     _results = [];
#   for (_i = 0, _len = list.length; _i < _len; _i++) {
#         num = list[_i];
#         _results.push(math.cube(num));
#     }
#     return _results;
# })();

foods = ['broccoli', 'spinach', 'chocolate']
eat food for food in foods when food isnt 'chocolate'
#=>foods = ['broccoli', 'spinach', 'chocolate'];
#
#for (_k = 0, _len2 = foods.length; _k < _len2; _k++) {
#  food = foods[_k];
#  if (food !== 'chocolate') {
#    eat(food);
#  }
#}
```

Additional resources

- Smooth CoffeeScript
- CoffeeScript Ristretto