CoffeeScript

CoffeeScript is a programming language that compiles to JavaScript. It adds syntactic sugar inspired by Ruby, Python and Haskell in an effort to enhance JavaScript's brevity and readability.[4] Specific additional features include list comprehension and destructuring assignment.

CoffeeScript support is included in Ruby on Rails version 3.1[5] and Play Framework.[6] In 2011, Brendan Eich referenced CoffeeScript as an influence on his thoughts about the future of JavaScript.[7][8]

CoffeeScript

Paradigm	Multi-paradigm: prototype-based, functional, imperative, scripting
Designed by	Jeremy Ashkenas
Developer	Jeremy Ashkenas
First appeared	13 December 2009
Stable release	2.5.1[1] / 31 January 2020
Typing discipline	dynamic, implicit
OS	Cross-platform
License	MIT License
Filename extensions	.coffee, .litcoffee
Website	coffeescript.org (https://coffeescript.org/)

Influenced by

Haskell, JavaScript, Perl, Python,[2] Ruby, YAML[3]

Influenced

MoonScript, LiveScript, JavaScript

Contents

History
Syntax
Development and distribution
Latest additions
Extensions
Adoption
See also
References
Further reading
External links

History

On December 13, 2009, Jeremy Ashkenas made the first Git commit of CoffeeScript with the comment: "initial commit of the mystery language."[9] The compiler was written in Ruby. On December 24, he made the first tagged and documented release, 0.1.0. On February 21, 2010, he committed version 0.5, which replaced the Ruby compiler with a self-hosting version in pure CoffeeScript. By that time the project had attracted several other contributors on GitHub, and was receiving over 300 page hits per day.

On December 24, 2010, Ashkenas announced the release of stable 1.0.0 to Hacker News, the site where the project was announced for the first time.[10][11]

On September 18, 2017, version 2.0.0 was introduced,[12] which "aims to bring CoffeeScript into the modern JavaScript era, closing gaps in compatibility with JavaScript while preserving the clean syntax that is CoffeeScript's hallmark."

Syntax

Almost everything is an expression in CoffeeScript, for example `if`, `switch` and `for` expressions (which have no return value in JavaScript) return a value. As in Perl, these control statements also have postfix versions; for example, `if` can also be written in `consequent if condition` form.

Many unnecessary parentheses and braces can be omitted; for example, blocks of code can be denoted by indentation instead of braces, function calls are implicit, and object literals are often detected automatically.

To compute the body mass index, one may do (here in JavaScript):

```
const mass = 72
const height = 1.78
const BMI = mass / height ** 2
if (18.5 < BMI && BMI < 25) { alert('You are healthy!') }
```

With CoffeeScript the interval is directly described:

```
mass = 72
height = 1.78
BMI = mass / height**2
alert 'You are healthy!' if 18.5 < BMI < 25
```

To compute the greatest common divisor of two integers with the euclidean algorithm, in JavaScript one usually needs a *while* loop:

```
gcd = (x, y) => {
  do {
    z = x % y
    x = y
    y = z
  } while (y !== 0)
  return x
}
```

Whereas in CoffeeScript one can use `until` and destructuring assignment[13] instead:

```
gcd = (x, y) ->
  [x, y] = [y, x%y] until y is 0
  x
```

Any *for* loop can be replaced by a list comprehension; so that to compute the squares of the positive odd numbers smaller than ten (i.e. numbers whose remainder modulo 2 is 1), one can do:

```
alert n*n for n in [1..10] when n%2 is 1
```

Alternatively, there is:

```
alert n*n for n in [1..10] by 2
```

A linear search can be implemented with a one-liner using the when keyword:

```
names = ["Ivan", "Joanna", "Nikolay", "Mihaela"]
linearSearch = (searchName) -> alert(name) for name in names when name is searchName
```

The `for ... in` syntax allows looping over arrays while the `for ... of` syntax allows looping over objects.

The `?` keyword quickly checks if a variable is `null` or `undefined` :

```
personCheck = ->
  if not person? then alert("No person") else alert("Have person")
person = null
personCheck()
person = "Ivan"
personCheck()
```

This would alert "No person" if the variable is `null` or `undefined` and "Have person" if there is something there.

A common JavaScript snippet using the jQuery library is:

```
$(document).ready(function() {
  // Initialization code goes here
})
```

Or even just:

```
$(function() {
  // Initialization code goes here
})
```

In CoffeeScript, the `function` keyword is replaced by the `->` symbol, and indentation is used instead of curly braces, as in other off-side rule languages such as Python and Haskell. Also, parentheses can usually be omitted, using indentation level instead to denote a function or block. Thus, the CoffeeScript equivalent of the snippet above is:

```
$(document).ready ->
  # Initialization code goes here
```

Or just:

```
$ ->
  # Initialization code goes here
```

Ruby-style string interpolation is included in CoffeeScript. Double-quoted strings allow for interpolated values, using #{ ... }, and single-quoted strings are literal.[14]

```
author = "Wittgenstein"
quote  = "A picture is a fact. -- #{ author }"

sentence = "#{ 22 / 7 } is a decent approximation of π"
```

CoffeeScript has been criticized for its unusual scoping rules.[15][16] In particular, it completely disallows variable shadowing which makes reasoning about code more difficult and error-prone in some basic programming patterns established by and taken for granted since procedural programming principles were defined.

For example, with the following code snippet in JavaScript one does not have to look outside the `{}`-block to know for sure that no possible `foo` variable in the outer scope can be incidentally overridden:

```
  // ...
  function baz() {
    var foo = "bar"
    console.log(`foo = ${foo}`)
  }
  // ...
}
```

In CoffeeScript there is no way to tell if the scope of a variable is limited to a block or not without looking outside the block.

Development and distribution

The CoffeeScript compiler has been self-hosting since version 0.5 and is available as a Node.js utility; however, the core compiler does not rely on Node.js and can be run in any JavaScript environment.[17] One alternative to the Node.js utility is the Coffee Maven Plugin, a plugin for the Apache Maven build system. The plugin uses the Rhino JavaScript engine written in Java.

The official site at CoffeeScript.org has a "Try CoffeeScript" button in the menu bar; clicking it opens a modal window in which users can enter CoffeeScript, see the JavaScript output, and run it directly in the browser. The js2coffee[18] site provides bi-directional translation.

Latest additions

- Source maps allow users to de-bug their CoffeeScript code directly, supporting CoffeeScript tracebacks on run time errors.
- CoffeeScript supports a form of Literate Programming, using the `.coffee.md` or `.litcoffee` file extension. This allows CoffeeScript source code to be written in Markdown. The compiler will treat any indented blocks (Markdown's way of indicating source code) as code, and ignore the rest as comments.

Extensions

Iced CoffeeScript is a superset of CoffeeScript which adds two new keywords: `await` and `defer`. These additions simplify asynchronous control flow, making the code to look more like a procedural programming language, eliminating the call-back chain. It can be used on the server side and in the browser.[19]

Adoption

On September 13, 2012, Dropbox announced that their browser-side code base has been rewritten from JavaScript to CoffeeScript,[20] however it has been migrated to TypeScript in 2017.[21]

GitHub's internal style guide once said "write new JS in CoffeeScript", and while it no longer does, all the advice in the style guide references how to write good CoffeeScript,[22] and their Atom text editor is also written in the language.[23]

Pixel Game Maker MV makes uses of CoffeeScript as part of its game development environment.[24]

See also

- Haxe
- Nim (programming language)
- Amber Smalltalk
- Clojure
- Dart (programming language)
- Kotlin (programming language)
- LiveScript
- Opa (programming language)
- Elm (programming language)
- TypeScript
- PureScript

References

1. "Release 2.5.1" (https://github.com/jashkenas/coffeescript/releases/tag/2.5.1). 31 January 2020. Retrieved 1 February 2020.
2. https://coffeescript.org/ "CoffeeScript borrows chained comparisons from Python"
3. Heller, Martin (18 October 2011). "Turn up your nose at Dart and smell the CoffeeScript" (https://www.infoworld.com/article/2078452/turn-up-your-nose-at-dart-and-smell-the-coffeescript.html). *InfoWorld*. Retrieved 2020-07-15.
4. Alex MacCaw (January 2012). *The Little Book on CoffeScript*. O'Reilly Media. ISBN 978-1-4493-2105-5.
5. Josh Peek (April 13, 2011). "Tweet by Rails Core Team Member" (https://twitter.com/joshpeek/status/58184348742074368).
6. "AssetsCoffeeScript - 2.5.x" (https://www.playframework.com/documentation/2.5.x/AssetsCoffeeScript). *www.playframework.com*. Retrieved 2016-10-31.
7. Eich, Brendan. "Harmony of My Dreams (http://brendaneich.com/2011/01/harmony-of-my-dreams/)"
8. Eich, Brendan. "My JSConf.US Presentation (http://brendaneich.com/2011/05/my-jsconf-us-presentation/)"
9. Github. 'initial commit of the mystery language' (https://github.com/jashkenas/coffee-script/commit/8e9d637985d2dc9b44922076ad54ffef7fa8e9c2)
10. Hacker News. CoffeeScript 1.0.0 announcement (https://news.ycombinator.com/item?id=2037801) posted by Jeremy Ashkenas on Dec 24, 2010
11. Hacker News. Original CoffeeScript announcement (https://news.ycombinator.com/item?id=1014080) posted by Jeremy Ashkenas on Dec 24, 2009
12. coffeescript.org Announcing CoffeeScript 2 (http://coffeescript.org/announcing-coffeescript-2/)
13. CoffeeScript calls this "pattern matching", which is a non-standard use of that term.
14. "Official CoffeeScript Page" (http://coffeescript.org). Retrieved 20 November 2013.

15. "The Problem with Implicit Scoping in CoffeeScript" (http://lucumr.pocoo.org/2011/12/22/implicit-scoping-in-coffeescript/). Retrieved 2018-10-13.
16. "CoffeeScript's Scoping is Madness" (https://donatstudios.com/CoffeeScript-Madness). Retrieved 2018-10-13.
17. CoffeeScript (https://jashkenas.github.com/coffee-script/#installation) Archived (https://web.archive.org/web/20120427060308/http://jashkenas.github.com/coffee-script/) 2012-04-27 at the Wayback Machine. Jashkenas.github.com. Retrieved on 2013-07-21.
18. Sta Cruz, Rico. "js2coffee" (http://js2.coffee). Retrieved 11 May 2014.
19. "Official IcedCoffeeScript website" (http://maxtaco.github.io/coffee-script/).
20. Wheeler, Dan; Mahkovec, Ziga; Varenhorst, Chris (13 September 2012). "Dropbox dives into CoffeeScript" (https://tech.dropbox.com/?p=361). Retrieved 11 May 2013.
21. Goldstein, David (13 May 2020). "The Great CoffeeScript to Typescript Migration of 2017" (https://dropbox.tech/frontend/the-great-coffeescript-to-typescript-migration-of-2017). *Dropbox.Tech*. Retrieved 30 June 2020.
22. "JavaScript · Styleguide · GitHub" (https://web.archive.org/web/20130815075924/https://github.com/styleguide/javascript). Github.com. Archived from the original (https://github.com/styleguide/javascript) on 2013-08-15. Retrieved 2015-11-30.
23. Atom source code (https://github.com/atom/atom). github.com. Retrieved on 2015-07-22.
24. Cullen, Daniel. "PIXEL GAME MAKER MV (PC)" (https://www.christcenteredgamer.com/reviews/pc-mac/7520-pixel-game-maker-mv-pc). *Christ Centered Gaming*. Retrieved 15 January 2021.

Further reading

- Lee, Patrick (May 14, 2014). *CoffeeScript in Action* (First ed.). Manning Publications. p. 432. ISBN 978-1617290626.
- Grosenbach, Geoffrey (May 12, 2011). "Meet CoffeeScript" (First ed.). PeepCode.
- Bates, Mark (May 31, 2012). *Programming in CoffeeScript* (First ed.). Addison-Wesley. p. 350. ISBN 978-0-321-82010-5.
- MacCaw, Alex (January 31, 2012). *The Little Book on CoffeeScript* (First ed.). O'Reilly Media. p. 62. ISBN 978-1449321055.
- Burnham, Trevor (August 3, 2011). *CoffeeScript: Accelerated JavaScript Development* (https://archive.org/details/isbn_9781934356784/page/138) (First ed.). Pragmatic Bookshelf. p. 138 (https://archive.org/details/isbn_9781934356784/page/138). ISBN 978-1934356784.

External links

- Official website (https://coffeescript.org/)