

What is a Lambda?

Asked 16 years, 2 months ago Modified 5 years, 10 months ago

Viewed 19k times



95



Could someone provide a good description of what a Lambda is? We have a tag for them and they're on the secrets of C# question, but I have yet to find a good definition and explanation of what they are in the first place.



language-agnostic

lambda

computer-science

terminology

Share

Improve this question

Follow

edited Jan 31, 2013 at 13:04



hippietrail

16.9k ● 21 ● 109 ● 173

asked Sep 29, 2008 at 18:56



Fred

2,733 ● 3 ● 28 ● 31

1 How about [Wikipedia's lambda calculus article](#) for a start?
Then [Wikipedia's functional programming article](#) as a
followup. – [Thorsten79](#) Sep 29, 2008 at 18:58

2 possible duplicate of [What is a lambda \(function\)?](#) – [nawfal](#)
Jul 4, 2014 at 6:04

lambda expression explained [here](#) beautifully. – [Jameer Mulani](#)
Dec 4, 2017 at 16:35

7 Answers

Sorted by:

Highest score (default)



Closures, lambdas, and anonymous functions are not necessarily the same thing.

135



An anonymous function is any function that doesn't have (or, at least, need) its own name.



A closure is a function that can access variables that were in its lexical scope when it was declared, even after they have fallen *out* of scope. Anonymous functions do not necessarily have to be closures, but they are in most languages and become rather less useful when they aren't.

A lambda is.. not quite so well defined as far as computer science goes. A lot of languages don't even use the term; instead they will just call them closures or anon functions or invent their own terminology. In LISP, a lambda is just an anonymous function. In Python, a lambda is an anonymous function specifically limited to a single

expression; anything more, and you need a named function. Lambdas are closures in both languages.

Share Improve this answer

answered Sep 29, 2008 at 19:12

Follow



Eevee

48.5k ● 11 ● 98 ● 127

the term might be intended to imply lambda calculus
en.wikipedia.org/wiki/Lambda_calculus – Steven A. Lowe
Nov 18, 2008 at 4:47

Wow I wish I had more than one upvote to give this question.
I've seen so many technical answers bogged down in
terminology, but this one is concise and makes the point
immediately. – Matthew Stopa May 8, 2011 at 22:47

-
- 5 A region where some variable can exist (i.e., a scope)
defined by the structure of your code (i.e., lexically). For
example, in most languages, a function defines a lexical
scope; any variable declared inside the function is
inaccessible outside of it. – Eevee Dec 3, 2012 at 3:24
-



19

Also called closures or anonymous functions.. I found the
best description [here](#). Basically, inline block of code that
can be passed as an argument to a function.



Share Improve this answer

edited Sep 29, 2008 at 19:09

Follow



answered Sep 29, 2008 at 19:03



Gulzar Nazim

52.2k ● 26 ● 130 ● 170



18



"Lambda" refers to the [Lambda Calculus](#) or to a specific lambda expression. Lambda calculus is basically a branch of logic and mathematics that deals with functions, and is the basis of [functional programming languages](#).

~ William Riley-Land

Share Improve this answer

edited Aug 10, 2009 at 20:05

Follow

answered Sep 29, 2008 at 19:00



wprl

25.4k ● 11 ● 57 ● 70



8



It's just an anonymous function declared inline, most typically assigned to a delegate when you don't want to write a full-fledged function.

In languages like lisp/scheme, they're often passed around quite liberally as function parameters, but the idiom in C# typically finds lambdas used only for lazy evaluation of functions, as in linq, or for making event-handling code a bit terser.

Share Improve this answer

edited Sep 29, 2008 at 19:04

Follow

answered Sep 29, 2008 at 18:59



JasonTrue

19.6k ● 5 ● 37 ● 61



5



There's not really such a thing as 'a lambda' in programming. It depends on the language, etc.

In short, normally a language that 'has lambdas' uses the term for anonymous functions or, in some cases, closures. Like so, in Ruby:



```
f = lambda { return "this is a function with no name" }  
puts f.call
```

Share Improve this answer

answered Sep 29, 2008 at 19:04

Follow



rfunduk

30.4k ● 5 ● 60 ● 55



4



In response to the previous answers:

- The important thing about anonymous functions is not that they don't require a name.

- Closures are a separate concept.

- A gigantic wikipedia article is not making this any clearer.



Here's my answer in 3 parts:

1. A lambda is a function which is also an expression.

This is the important thing.

2. Many languages which implement so-called "lambdas" add some *syntactic sugar* to make writing these short functions easier and faster, but this is not required.

3. Some languages may require that a lambda has *no side effects*. That would be a more pure lambda in the functional sense.

When a function is an expression, it's a "first class citizen" within the language. I can do all the important things with it:

```
x = lambda(){ return "Hello World"; }

doit( 1, 2, lambda(a,b){ return a > b; }, 3 )

x = (lambda(a){ return a+1; }) + 5  // type error,
not syntax error

(lambda(a,b){ print(a); log(b); })( 1, 2 )  // ( )
is valid operator here
```

Share Improve this answer

answered Aug 17, 2011 at 7:42

Follow



cmeub

517 ● 2 ● 11



Clipped from wikipedia:

http://en.wikipedia.org/wiki/Lambda#Lambda.2C_the_wor
[d](#)

1



In programming languages such as Lisp and Python, lambda is an operator used to denote anonymous functions or closures, following lambda calculus usage.



Share Improve this answer

answered Sep 29, 2008 at 18:59

Follow



Mesh

6,402 ● 5 ● 36 ● 55

I guess you meant this line: In programming languages such as Lisp and Python, lambda is an operator used to denote anonymous functions or closures, following lambda calculus usage. I'd already scanned the article but missed that.

– [Fred](#) Sep 29, 2008 at 19:03
