

What is the lambda calculus

Compare the following concrete three sentences (i add brackets to separate them):

(John gives Mary a kiss)

(Fred gives Richie a banana)

(Alfred gives Jeff a hint)

You can generalize those sentences with a **function** (another word is **expression**):

(X gives Y a Z)

This function (or expression) contains the variables X, Y and Z. To make that clear, lets write those variables in front, so that we know what is a variable and what is not:

(X Y Z (X gives Y a Z))

Lets also add the word 'function' at the beginning, to make clear that this is a function:

(function X Y Z (X gives Y a Z))

This function does not do much. We have to fill the variables with concrete values to get concrete sentences out of it. Lets write the concrete values behind the function:

((function X Y Z (X gives Y a Z)) John Mary kiss)

When we execute (or evaluate) this function we get:

(John gives Mary a kiss)

Now we do it with the other two concrete sentences from the start:

((function X Y Z (X gives Y a Z)) Fred Richie banana)

... results in ...

(Fred gives Richie a banana)

... and ...

((function X Y Z (X gives Y a Z)) Alfred Jeff hint)

... results in ...

(Alfred gives Jeff a hint)

Notice that the inventor, Alonzo Church, called it lambda calculus (he learned greek at school) so today people do not write ...

(function X Y Z (X gives Y a Z))

What is the lambda calculus (cont)

... but instead ...

(lambda X Y Z (X gives Y a Z))

Thats all what you have to know to use the lambda calculus in practice.

What is Scheme

Scheme is a successor of Lisp, both programming languages. Lisp was the first language that implemented the lambda calculus.

The way they talk about Scheme is wrong.

Some People talk about Scheme as if it were the theory of relativity. They throw lots of Math Spam at their readers (Math is easy too btw., but it has a short and therefore bizarre syntax). If you see docs like this and you do not like it, do not complain, instead leave immediately. You will find docs explaining it the easy way.

Scheme sourcecode formatting

Instead of ...

```
(define (hello name)
  ....(string-append "Hello " name)
)
```

.. or ...

```
(
  ....define
  ....(hello name)
  ....(string-append "Hello " name)
)
```

... you will see ...

```
(define (hello name)
  ....(string-append "Hello " name))
```

... which is obviously hard to read (but maybe easier to code). If you see code formatted like this and you do not like it, copy it into your editor and format it the way you want it.

What is a REPL

"REPL" is a shortcut for "read-eval-print loop". You have a window where you enter a line of code. Then you press enter. Then the program **reads** the code you entered. Then it **evaluates** it. Then it **prints** the result (if there is one) to the next line. Now the game starts at the beginning, you enter code, you press enter, etc. Therefore it is called a **loop**.

You usually enter your code at lines starting with '>', the results of an evaluation are printed on lines that do not start with '>'

Example repl session:

```
> 5
5
> "Hello World"
"Hello World"
> (define (hello name) (string-append "Hello " name))
> (hello "John")
"Hello John"
>
```

the repl is a practical thing to do quick tests. Scheme was the first programming language where environments provided it and it has been adopted by many other languages, for example Python.

Downloading Racket and starting the repl

Download racket from here:

<http://racket-lang.org/download/>

install it, then start the Racket.exe in the installation folder. A repl opens. You can quit this repl by typing ...

```
> (exit)
```

What is an expression

An expression is something that evaluates to something which is simpler or identical but usually not more complex.



By [deleted]

cheatography.com/deleted-916/

Not published yet.

Last updated 15th April, 2014.

Page 1 of 2.

Sponsored by **Readable.com**

Measure your website readability!

<https://readable.com>

Simple Expressions

Simple expressions evaluate to themselves, like Numbers ...

```
> 5
```

```
5
```

... and strings (a string is a list of chars surrounded by "):

```
>"hello"
```

```
"hello"
```

Complex expressions

Complex expressions are surrounded with brackets and evaluate to something simpler:

```
> (substring "the boy out of the country" 4 7)
```

```
"boy"
```

this expression is a call of the "substring" function, which has three parameters, a string and two numbers. it returns a string.

C

By **[deleted]**

cheatography.com/deleted-916/

Not published yet.

Last updated 15th April, 2014.

Page 2 of 2.

Sponsored by **Readable.com**

Measure your website readability!

<https://readable.com>