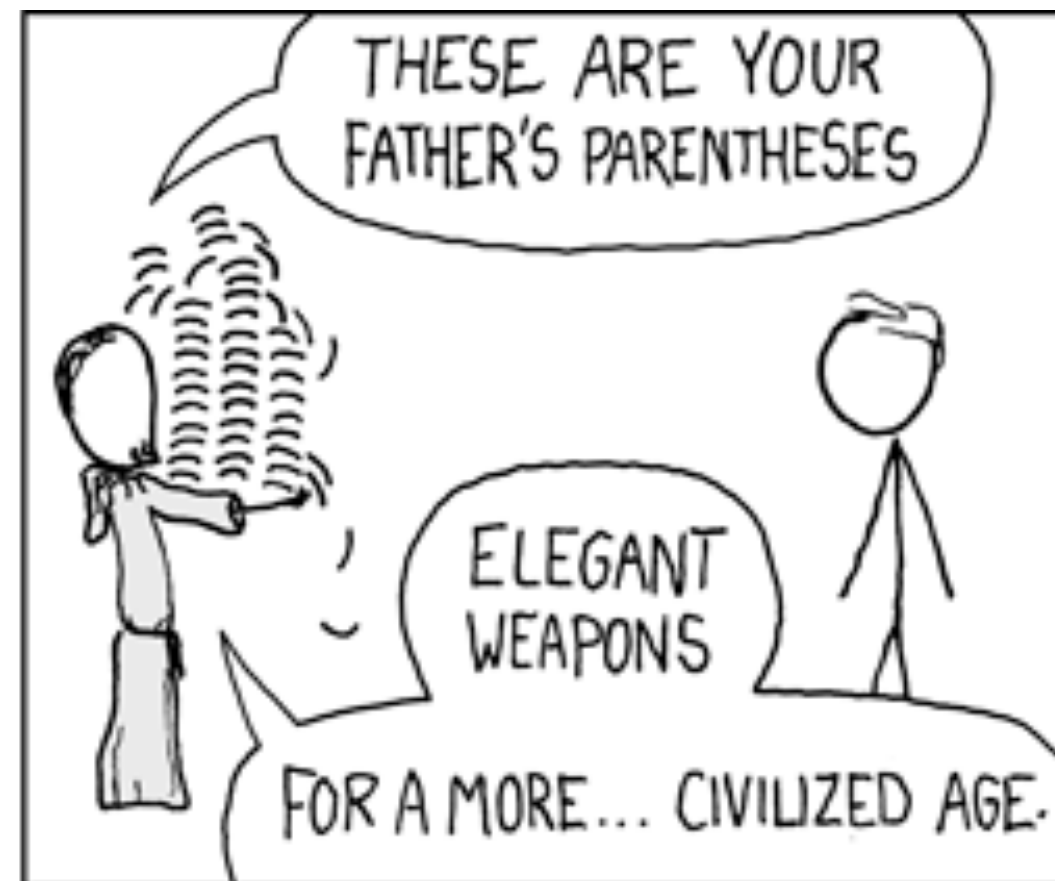
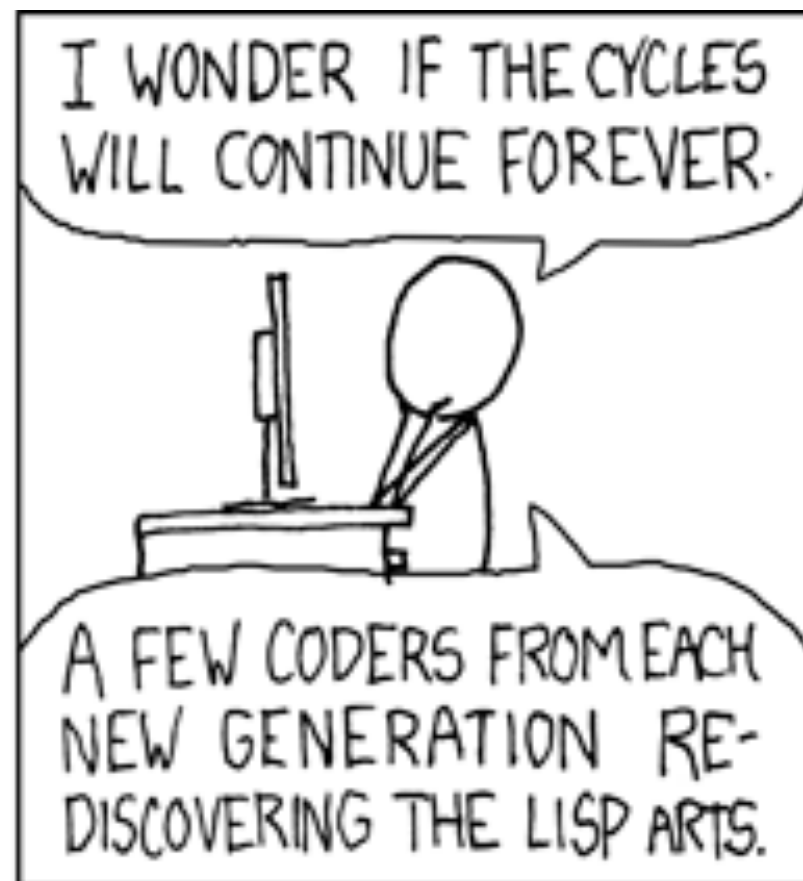




<http://xkcd.com/297/>

Lisp is beautiful!

Jfokus 2013
Alf Kristian Støyle

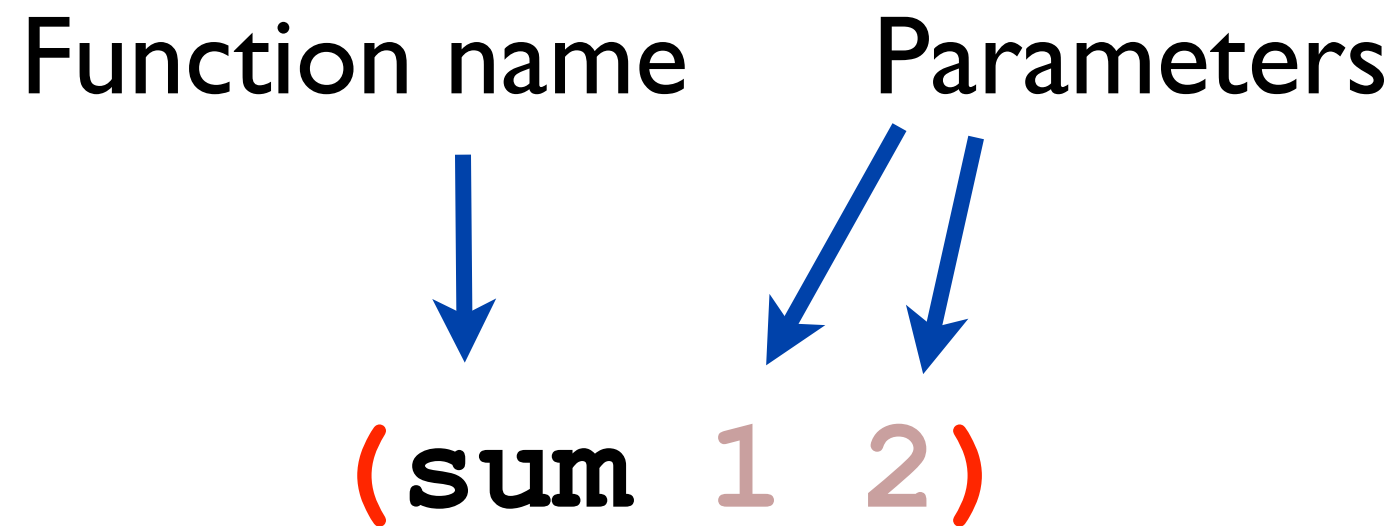
knowit

sum (1 , 2) ;

sum (1 , 2)

(sum 1 2)

Prefix notation



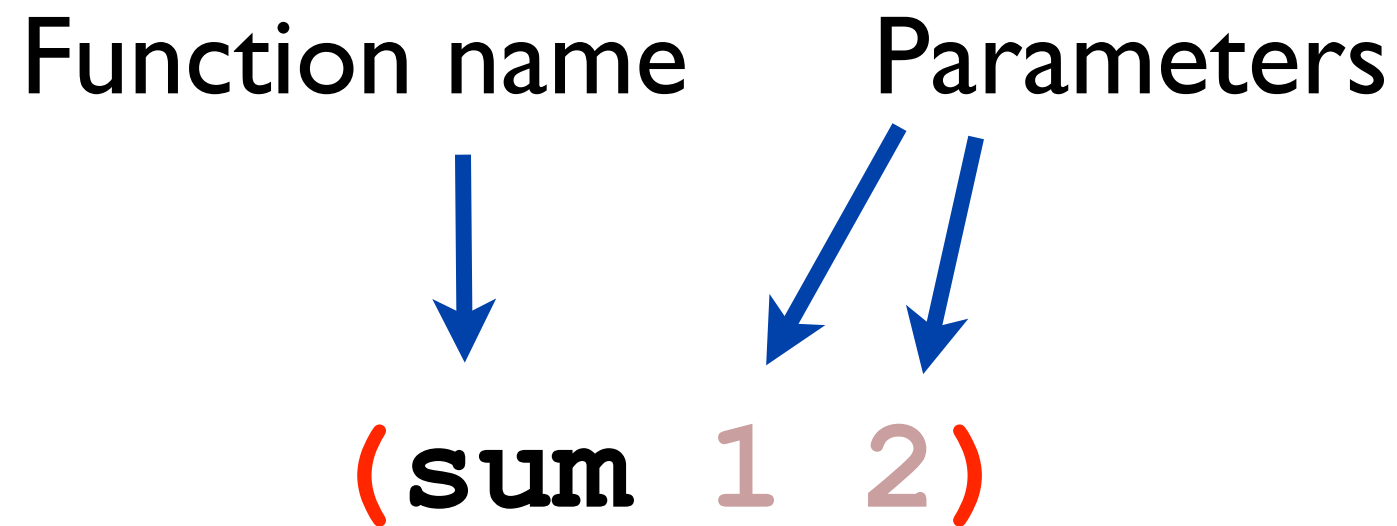
```
Methods.callStatic();  
methods.callInstance();
```

```
callStatic();  
callInstance();
```

```
this.callInstance();  
methods.callStatic();
```

```
Methods.this.callInstance();  
Methods.this.callStatic();
```


Prefix notation



(sum 1 2)

=> 3

(sum 1 2 3)

=> 6

(sum)

=> 0

(+ 1 2)

=> 3

1 + 2

(+ 1 2 3)

=> 6

1 + 2 + 3

(+)

=> 0

$$1 + (2 * 3)$$

1 + (2 * 3)

(+ 1 (* 2 3))

$$1 + (2 * 3) / 2 - 4$$

$$(- (+ 1 (/ (* 2 3) 2)) 4)$$

$$((1 + ((2 * 3) / 2)) - 4)$$

Lists

```
' (3 2 1)  
; => (3 2 1)
```

```
(+ 3 2 1)  
; => 6
```

```
(defn iterate  
  [f x] (cons x (lazy-seq (iterate f (f x) ) ) ) )
```


Lists

```
' (3 2 1)  
; => (3 2 1)
```

```
(+ 3 2 1)  
; => 6
```

```
(iterate inc 0)
```

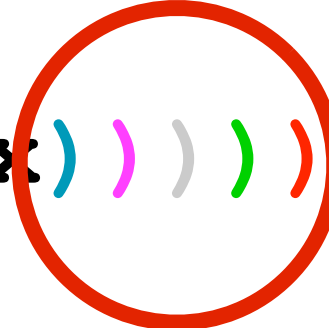
```
(defn iterate  
  [f x] (cons x (lazy-seq (iterate f (f x))))))
```


Lists

```
' (3 2 1)  
; => (3 2 1)
```

```
(+ 3 2 1)  
; => 6
```

```
(defn iterate  
  [f x] (cons x (lazy-seq (iterate f (f x) ) ) ) )
```



Macros

```
(infix (1 + 2))  
; => 3
```

```
(defmacro infix [form]  
  ` (~ (second form) ~ (first form) ~@ (nnext form) ) )
```

Macros

```
(macroexpand '(infix (1 + 2)))  
; => (+ 1 2)
```

```
(defmacro infix [form]  
  `(~(second form) ~ (first form) ~@ (nnext form) ) )
```

```
(defn iterate
  [f x] (cons x (lazy-seq (iterate f (f x)))))
```

Abstract syntax tree

```
(defn iterate
  [f x]
    (cons x
          (lazy-seq
            (iterate f
                    (f x) ) ) ) )
```

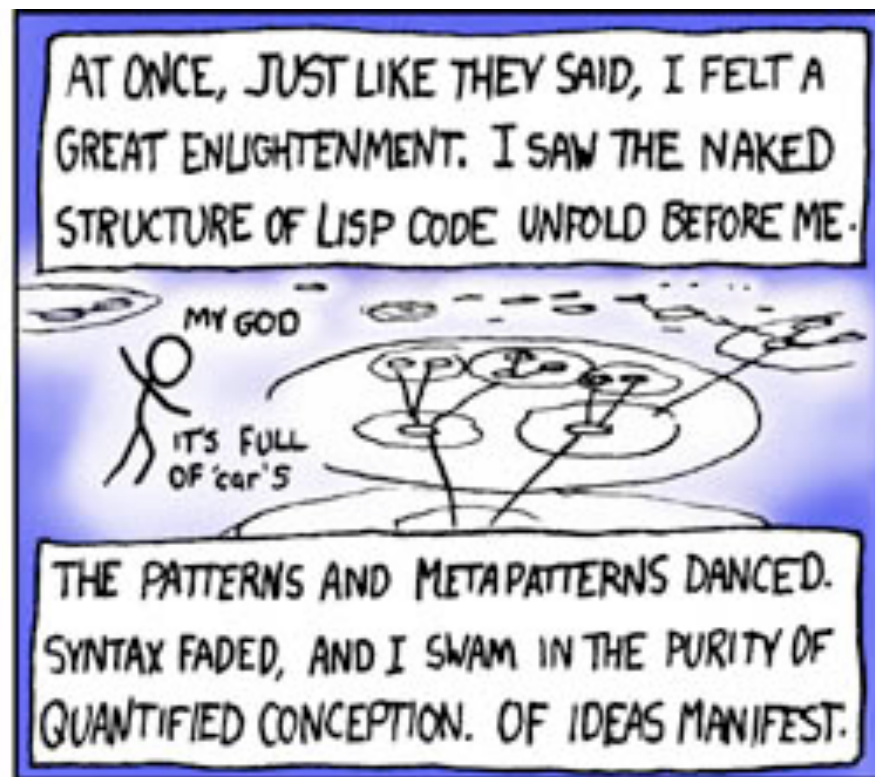
Clojure



Q & A

aks@knowit.no

@stoyle



TRULY, THIS WAS THE LANGUAGE FROM WHICH THE GODS WROUGHT THE UNIVERSE.



<http://xkcd.com/224/>