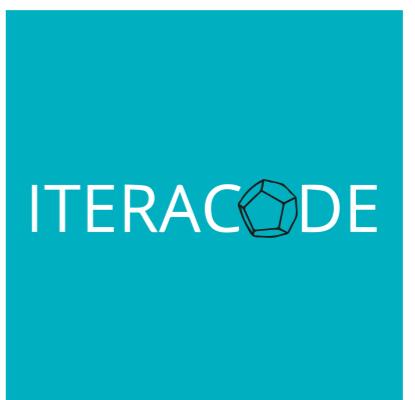




Pourquoi je trouve Clojure cool ?

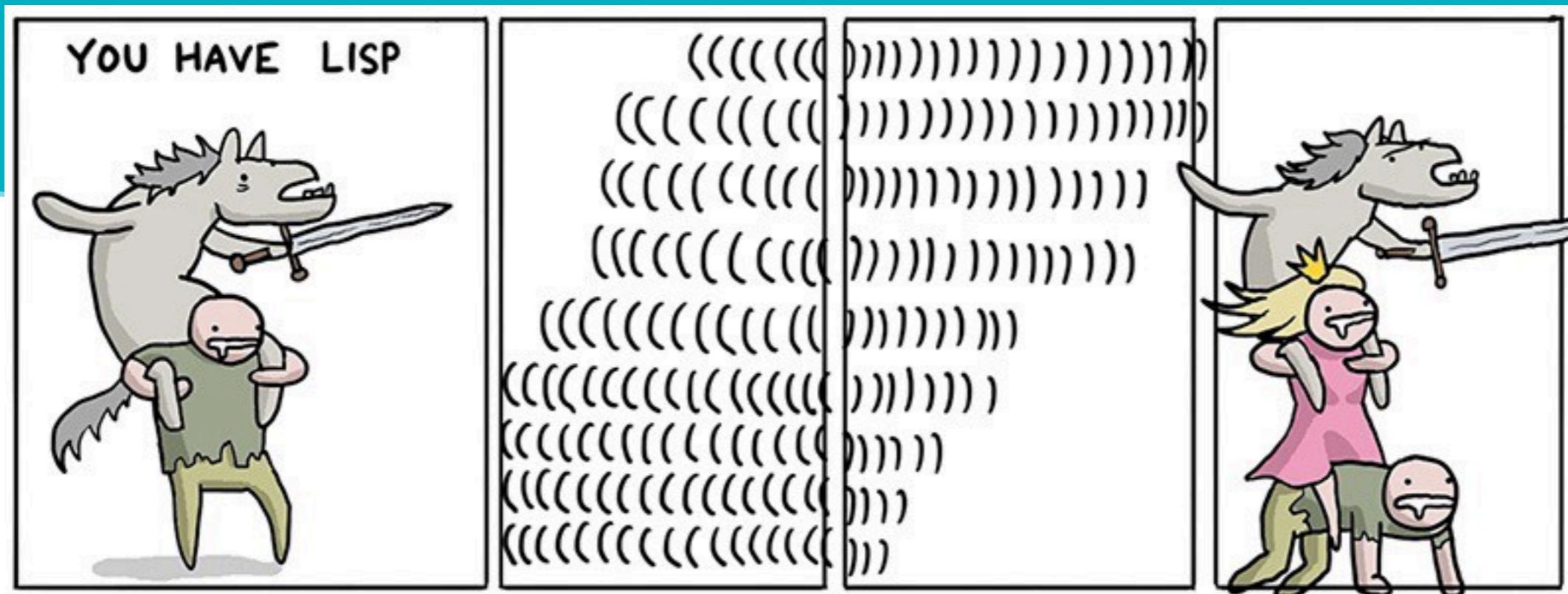
Charles Fourdrignier - Lead Dev



Expérience

- Un peu d'Advent of Code
- « Testeur d'API REST » pour la Nuit de l'Info
- Scrapeur de la bibliothèque d'Amiens
- Analyse de CSV
- Computer Vision (OpenCV)

C'est quoi Clojure ?



Né en 2007



Né en 2007



(lisp)



JVM

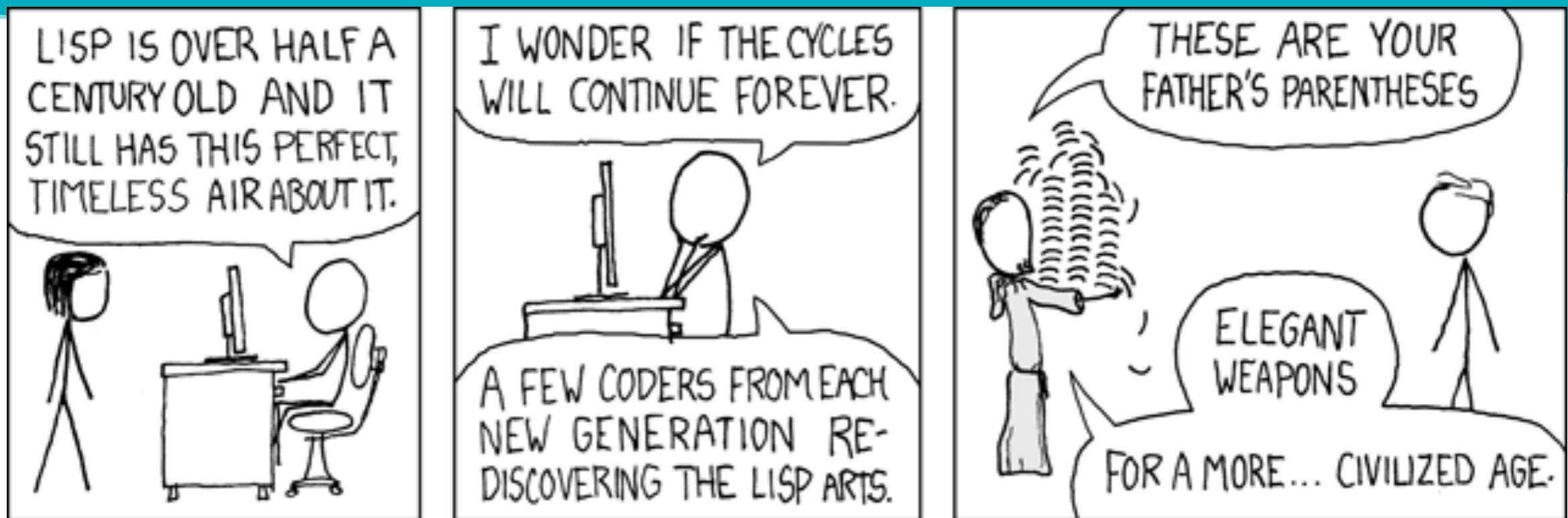
LISP 101

calc(1, 2)

LISP 101

calc(1, 2) (calc 1 2)

Pourquoi c'est cool ?



Notation préfixée

(+ 1 2 5 4 7)

Notation préfixée

(+ 1 2 5 4 7)

(<= 0 x y 255)

Immutabilité

```
{  
  :name "Twilight Starla Sparkle"  
  :image "images/twilight.jpg"  
  :meta {  
    :prices {  
      :price 3300  
      :premiumPrice 3900  
    }  
  }  
}
```

```
{  
  :name "Twilight Starla Sparkle"  
  :image "images/twilight.jpg"  
  :meta {  
    :prices {  
      :price 3301  
      :premiumPrice 3900  
    }  
  }  
}
```

Immutabilité

```
{ :name "Twilight Starla Sparkle" :image "images/twilight.jpg" :meta { :prices { :price 3300 :premiumPrice 3900 } } } { :name "Twilight Starla Sparkle" :image "images/twilight.jpg" :meta { :prices { :price 3301 :premiumPrice 3900 } } }
```

(update-in poney [:meta :prices :price] inc)

Problème de crypto

(frequencies text)

Problème de crypto

(frequencies text)

{\a 3, \c 2, \d 1, ...}

Lazy Infinite Sequences

```
[0 1 1 2 3 5 8 13 21 ...]
```

Construction

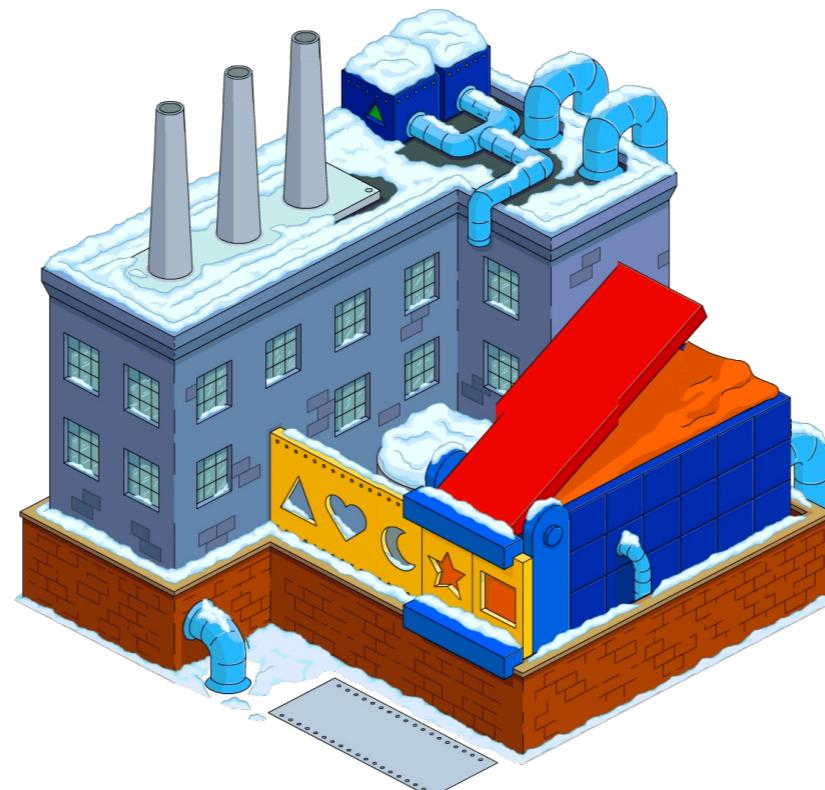
(iterate inc 0)

Construction

(iterate inc 0)

[0 1 2 3 4 5 ...]

Le monde réel



start-cycle



n cycles



Solution

```
(defn next-cycle [cycle] ...)  
  
(->>  
  (iterate next-cycle start-cycle)  
  (take n))
```

Et le dimanche ?

```
(defn next-cycle [cycle] ...)  
(defn working-day? [cycle] ...)  
  
(->>  
  (iterate next-cycle start-cycle)  
  (take n))
```

Et le dimanche ?

```
(defn next-cycle [cycle] ...)  
(defn working-day? [cycle] ...)  
  
(->>  
  (iterate next-cycle start-cycle)  
  (filter working-day?)  
  (take n))
```

Un nouveau problème

```
(defn next-cycle [cycle] ...)  
(defn working-day? [cycle] ...)  
  
(->>  
  (iterate next-cycle start-cycle)  
  (filter working-day?)  
  (take n)  
  )
```

Un nouveau problème

```
(defn next-cycle [cycle] ...)  
(defn working-day? [cycle] ...)  
  
(->>  
  (iterate next-cycle start-cycle)  
  (filter working-day?)  
  (drop n)  
  (first) )
```

Les boucles FOR ?

```
( for [x [0 1 2 3 4 5]]  
      (* x 2))
```

Les boucles FOR ?

```
( for [x [0 1 2 3 4 5]
      (* x 2))
```

```
[0 2 4 6 8 10]
```

Les boucles FOR ?

```
(for [x [0 1 2 3 4 5]  
      ]  
    (* x y) )
```

Les boucles FOR ?

```
(for [x [0 1 2 3 4 5]
      y [3 2 7 8 9 2]
      ]
  (* x y) )
```

Les boucles FOR ?

```
(for [x [0 1 2 3 4 5]
      y [3 2 7 8 9 2]
      ]
  (* x y))
```

[0 0 0 0 ... 10]

Les boucles FOR ?

```
( for [x [0 1 2 3 4 5]
      y [3 2 7 8 9 2]
          ]
      (* x y) )
```

Les boucles FOR ?

```
(for [x [0 1 2 3 4 5]
      y [3 2 7 8 9 2]
:when (< x y) ]
  (* x y))
```

Les boucles FOR ?

```
(for [x [0 1 2 3 4 5]
      y [3 2 7 8 9 2]
      :when (< x y) ]
      (* x y))
```

```
[0 0 0 0 0 0 3 2 7 8 9 2 6 14 16 18 21
24 27 28 32 36 35 40 45]
```

Ils n'ont pas pu venir..

REPL driven development

get default value

tree-seq

clojure.spec

partition

reductions

A vos questions !



ITERACODE

Studio d'innovation digitales, en mode Agile.

Charles Fourdrignier - Lead Développeur

charles@iteracode.fr

iteracode.fr - 09 81 36 43 87