Chaining Operator in Climb

Christopher Chedeau

LRDE
Laboratoire de Recherche et Développement d'EPITA

July 3, 2011

http://lrde.epita.fr/

Christopher Chedeau

Christopher Chedeau

jQuery

- ▸ Old browsers are full of compatibility issues

jQuery[1] is a [...] JavaScript Library that simplifies HTML document traversing

---

[1] http://jquery.com/

Christopher Chedeau

Method Chaining

- Intuitive API

- Heavily rely on Method Chaining

  ```
  $("a")
    .addClass("external")
    .attr("target", "_blank")
    .click(function () { /* analytics call */ })
  ```

Method Chaining in Javascript

```
function Object() {}
Object.prototype = {
  methodA: function () {
    // do something
    return this;
  },
  methodB: function () {
    // do something
    return this;
  }
};

var obj = new Object();
obj.methodA().methodB();
```

It works in any Object Oriented language:

- C++
- Java
- PHP
- C#
- ...

Christopher Chedeau

Extended Chaining

- ## HTML Document Traversing

```
$("li")              // li: List Item
  .has("ul")         // ul: Unordered List
  .eq(1)
  .parent()
```

Christopher Chedeau

Extended Chaining

- ## HTML Document Traversing

```
$("li")                    // li: List Item
  .has("ul")               // ul: Unordered List
  .eq(1)
  .parent()
```

- ## SQL Query

```
SQL("select")
  .field("*")
  .table("users")
  .limit(10)
```

Christopher Chedeau

Extended Chaining

- HTML Document Traversing

```
$("li")                    // li: List Item
  .has("ul")               // ul: Unordered List
  .eq(1)
  .parent()
```

- SQL Query

```
SQL("select")
  .field("*")
  .table("users")
  .limit(10)
```

- Asynchronous Tasks

```
$("<div/>")
  .fetch("navigation.html") // Asynchronous
  .addClass("column")
  .appendTo("#side")
```

Image processing

▸ Image processing chains can be expressed through method chaining

Image("lena.jpg")
  .togray()
  .otsu()
  .opening(8_connectivity)
  .save("lena_opening.png")

Christopher Chedeau

Straight Common Lisp Implementation

- Common Lisp methods do not belong to classes

```
(let ((img (load "lena.jpg")))
 (setf img (togray img))
 (setf img (otsu img))
 (setf img (opening img (8-connectivity)))
 (save img "lena_opening.png"))
```

Christopher Chedeau

Straight Common Lisp Implementation

- Common Lisp methods do not belong to classes

```
(let ((img (load "lena.jpg")))
  (setf img (togray img))
  (setf img (otsu img))
  (setf img (opening img (8-connectivity)))
  (save img "lena_opening.png"))
```

- The temporary variable is written 8 times

Christopher Chedeau

Without Temporary Variable

```
(save
 (opening
   (otsu
     (togray
       (load "lena.jpg")
     )
   )
   (8−connectivity))
 "lena_opening.png")
```

Christopher Chedeau

Without Temporary Variable

```
(save
 (opening
  (otsu
   (togray
    (load "lena.jpg")
    )
   )
  )
 (8−connectivity))
"lena_opening.png")
```

▶ Functions executed last are written first

Without Temporary Variable

```
(save
 (opening
  (otsu
   (togray
    (load "lena.jpg")
    )
   )
  )
 (8−connectivity))
"lena_opening.png")
```

▶ Functions executed last are written first

▶ Arguments are far away from function name

Christopher Chedeau

Introduction of Dollar $ Macro

Dollar $ Macro

```
($ (load "lena.jpg")
   (togray)
   (otsu)
   (opening 8−connectivity)
   (save "lena_opening.png"))
```

Method Chaining

```
Image("lena.jpg")
   .togray()
   .otsu()
   .opening(8_connectivity)
   .save("lena_opening.png")
```

- It takes a sequence of actions

Christopher Chedeau

Introduction of Dollar $ Macro

Dollar $ Macro

Method Chaining

```
($ (load "lena.jpg")
   (togray ☐ )
   (otsu)
   (opening 8−connectivity)
   (save "lena_opening.png"))
```

```
Image("lena.jpg")
   .togray()
   .otsu()
   .opening(8_connectivity)
   .save("lena_opening.png")
```

- ▸ It takes a sequence of actions
- ▸ The result of each action is passed as first argument on the next
- ▸ Rewritten either with or without temporary variable
- ▸ Can express all the method chaining features

Christopher Chedeau

Modifiers

- Not everything fits in the chaining pattern

  ($ (action)
     (print "Something")
     (action))

Christopher Chedeau

Modifiers

- ▶ Not everything fits in the chaining pattern

  ($ (action)
    '(print "Something")
    (action))

- ▶ We introduce the quote ' modifier

Christopher Chedeau

Modifiers

- Normal behavior

(before)
(action ☐ arg)
(after ☐)

Modifiers

- Normal behavior

  (before)
  (action ☐ arg)
  (after ☐)

- Quote ' Modifier

  (before) ⟶ $1, $2 ...
  '(action arg $1)
  (after ☐)
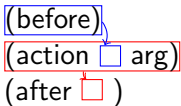
Modifiers
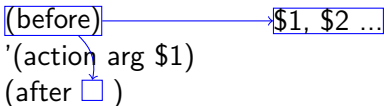
- Normal behavior

  (before)
  (action ☐ arg)
  (after ☐)
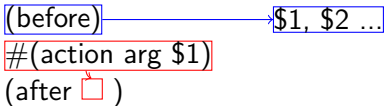
- Quote ' Modifier

  (before) ⟶ $1, $2 ...
  '(action arg $1)
  (after ☐)

- Sharp # Modifier

  (before) ⟶ $1, $2 ...
  #(action arg $1)
  (after ☐)

Grouped Actions

```
Image("lena.png")
  .preprocess()

  .background()
    .fill(red)

  .foreground()
    .fill(green)
```

Christopher Chedeau

Grouped Actions

```
Image("lena.png")              [lena]
  .preprocess()                [lena, proc]

  .background()                [lena, proc, background]
    .fill(red)                 [lena, proc, background]
  .end()                       [lena, proc]

  .foreground()                [lena, proc, foreground]
    .fill(green)               [lena, proc, foreground]
  .end()                       [lena, proc]
```

- ▸ Feels more like a hack
- ▸ Hard to know which methods push to the stack
- ▸ Elements in the stack cannot be garbage collected

Christopher Chedeau

Introduction of // Macro

```
($ (load "lena.png")
   (preprocess)

   (//
      ((background)
       (fill red))

      ((foreground)
       (fill green))))
```

Christopher Chedeau

Introduction of // Macro

```
($ (load "lena.png")
   (preprocess)

   (//
      ((background □ )
       (fill red))

      ((foreground □ )
       (fill green))))
```

▶ Explicit dispatch of the value

Introduction of // Macro

```
($ (load "lena.png")
   (preprocess)

   (//
      ((background)
       (fill red))

      ((foreground)
       (fill green)))

   (combine □ □ ))
```

- ▸ Explicit dispatch of the value
- ▸ Last value of branch can be collected

Example

Otsu

Sauvola 1

Sauvola 5

Otsu - Sauvola 1

Otsu - Sauvola 5

Sauvola 1 - 5

Christopher Chedeau

Example

($ (load "lenagray.png")

Christopher Chedeau

Example

```
($ (load "lenagray.png")
   (//
```

Christopher Chedeau

Example

```
($ (load "lenagray.png")
   (//
    ('(timer-start)
```

Christopher Chedeau

Example

```
($ (load "lenagray.png")
   (//
    ('(timer-start) (otsu)
```

Christopher Chedeau

Example

```
($ (load "lenagray.png")
   (//
    '(timer-start) (otsu)
    '(timer-print "Otsu")
```

Christopher Chedeau

Example

```
($ (load "lenagray.png")
    (//
      ('(timer-start) (otsu)
       '(timer-print "Otsu") (save "otsu.png")) ; $1
```

Christopher Chedeau

Example

```
($ (load "lenagray.png")
   (//
     ('(timer−start) (otsu)
      '(timer−print "Otsu") (save "otsu.png"))   ; $1
     ('(timer−start) (sauvola (box2d 1))
      '(timer−print "Sauvola 1") (save "sauvola1.png"))   ; $2
```

Christopher Chedeau

Example

```
($ (load "lenagray.png")
   (//
    ('(timer−start) (otsu)
     '(timer−print "Otsu") (save "otsu.png")) ; $1
    ('(timer−start) (sauvola (box2d 1))
     '(timer−print "Sauvola 1") (save "sauvola1.png")) ; $2
    ('(timer−start) (sauvola (box2d 5))
     '(timer−print "Sauvola 5") (save "sauvola5.png")))) ; $3
```

Christopher Chedeau

Example

```
($ (load "lenagray.png")
   (//
    ('(timer-start) (otsu)
     '(timer-print "Otsu") (save "otsu.png")) ; $1
    ('(timer-start) (sauvola (box2d 1))
     '(timer-print "Sauvola 1") (save "sauvola1.png")) ; $2
    ('(timer-start) (sauvola (box2d 5))
     '(timer-print "Sauvola 5") (save "sauvola5.png"))) ; $3
   (//
    (#(diff $1 $2)
```

Christopher Chedeau

Example

```
($ (load "lenagray.png")
   (//
    ('(timer-start) (otsu)
     '(timer-print "Otsu") (save "otsu.png")) ; $1
    ('(timer-start) (sauvola (box2d 1))
     '(timer-print "Sauvola 1") (save "sauvola1.png")) ; $2
    ('(timer-start) (sauvola (box2d 5))
     '(timer-print "Sauvola 5") (save "sauvola5.png"))) ; $3
   (//
    (#(diff $1 $2) (save "diff1.png"))
```

Christopher Chedeau

Example

```
($ (load "lenagray.png")
   (//
    ('(timer−start) (otsu)
     '(timer−print "Otsu") (save "otsu.png")) ; $1
    ('(timer−start) (sauvola (box2d 1))
     '(timer−print "Sauvola 1") (save "sauvola1.png")) ; $2
    ('(timer−start) (sauvola (box2d 5))
     '(timer−print "Sauvola 5") (save "sauvola5.png"))) ; $3
   (//
    (#(diff $1 $2) (save "diff1.png"))
    (#(diff $1 $3) (save "diff2.png"))
    (#(diff $2 $3) (save "diff3.png"))))
```

Christopher Chedeau

Example

```
($ (load "lenagray.png")
   (//
    ('(timer-start) (otsu)
     '(timer-print "Otsu") (save "otsu.png")) ; $1
    ('(timer-start) (sauvola (box2d 1))
     '(timer-print "Sauvola 1") (save "sauvola1.png")) ; $2
    ('(timer-start) (sauvola (box2d 5))
     '(timer-print "Sauvola 5") (save "sauvola5.png"))) ; $3
   (//
    (#(diff $1 $2) (save "diff1.png"))
    (#(diff $1 $3) (save "diff2.png"))
    (#(diff $2 $3) (save "diff3.png"))))

; Otsu: 101ms
; Sauvola 1: 263ms
; Sauvola 5: 3407ms
```

Christopher Chedeau

Conclusion

- ▶ Chaining Operator
  - • Ease of use
  - • Entire API has to be rethought
- ▶ Related Work
  - • Implementation of two Component Tree algorithms ([NC04], [BGL+07])
  - • Overhaul of Value representation
  - • Rationalization of Site Set interface
- ▶ Next Steps
  - • New design patterns from dynamic languages
  - • Explore image processing in the browser

Questions ?

Christopher Chedeau

Bibliography

📄 Christophe Berger, Thierry Géraud, Roland Levillain, Nicolas Widynski, Anthony Baillard, and Emmanuel Bertin, *Effective component tree computation with application to pattern recognition in astronomical imaging*, Proceedings of the IEEE International Conference on Image Processing (ICIP) (San Antonio, TX, USA), vol. 4, September 2007, pp. IV–41–IV–44.

📄 L. Najman and M. Couprie, *Quasi-linear algorithm for the component tree*, IS&T/SPIE Symposium on Electronic Imaging, In Vision Geometry XII, 2004, pp. 18–22.

Christopher Chedeau