

rubyでする Web Scraping
Hpricot と mechanize そして
scRUBYt

朴 芝印

Park Ji-in

自己紹介

- 名前 : 朴 芝印
- 職業 : 大学性
 - ヨンセー大学 四年生
- 専攻 : 科学工学
- 趣味のRuby & C プログラマ
- Rubykr - forum

目次

- Web Scrapingは何？
- 簡単なもの
- Hpricotの使い方
- Mechanizeの使い方
- scRUBYtの使い方
- その他
- 質問

Rubyでやっていること

- 自分の為のプログラムを作る
 - 0-gameの自動回避Bot
 - ActiveX代替りのダウンローダ
 - LDAP管理スクリプト

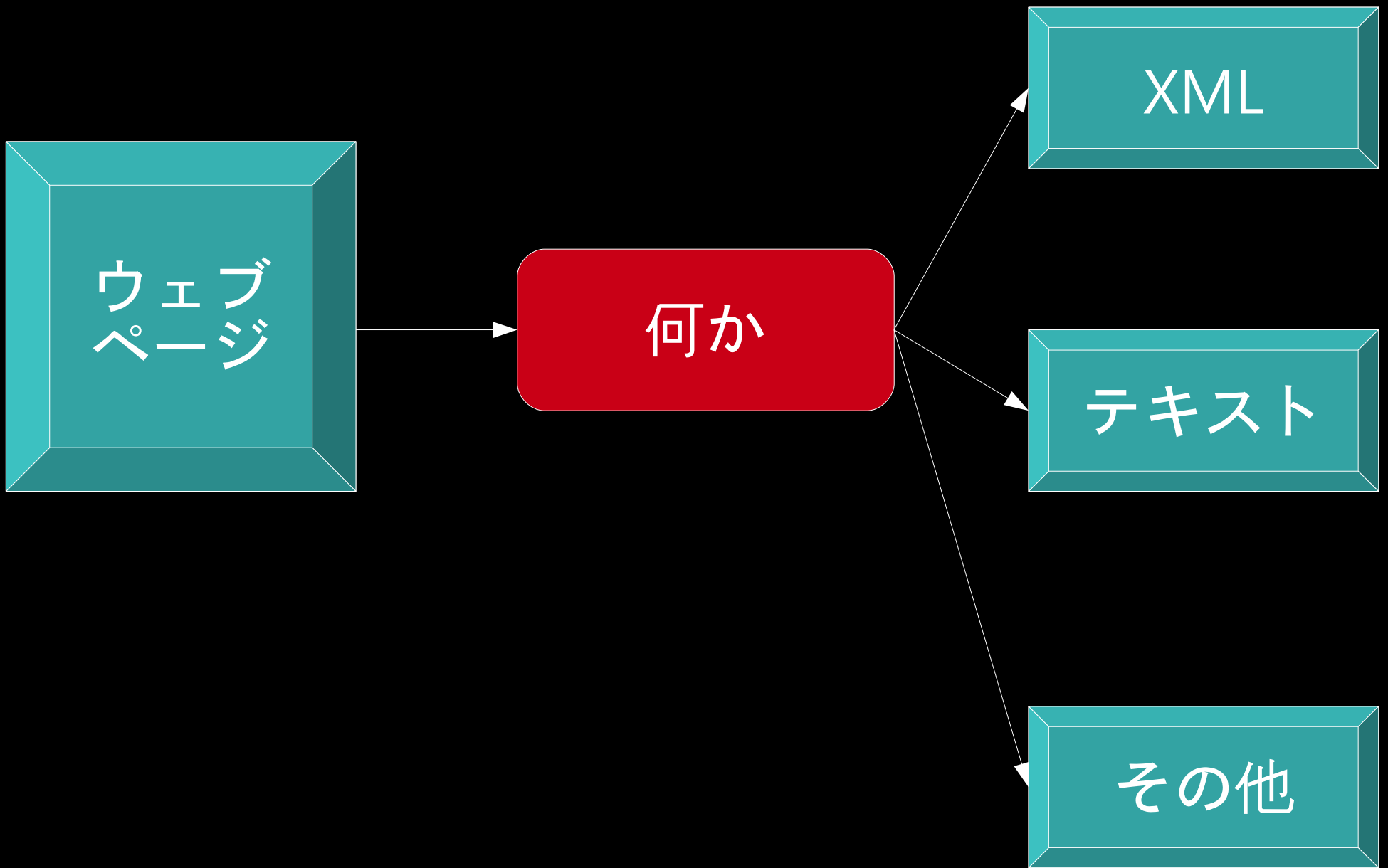
その他

- Programming Ruby 韓国語版翻訳

Web Scrapingて何？

- Web scraping generically describes any of various means to extract content from a website over HTTP for the purpose of transforming that content into another format suitable for use in another context.
- Web scrapingとは別のコンテキストでの使用のため、別のフォーマットに変換させるためにHTTPを通じてウェブサイトから内容を得ることだ。

- From Wikipedia



作ってみよう

簡単な方法

- open-url
- 文字列処理
- 正規表現
- read / puts / gets

google-search1.rb

```
require 'open-uri'
require 'cgi'

page = open('http://www.google.com/search?q=ruby')
html = page.read
results = html.scan(/<h2 class=r><a href="(.*?)" [^>]*>(.*?)</a>/).map {|x|
  { :link => x[0], :title => CGI.unescapeHTML(x[1].gsub(/<.*?>/, "")) }
}

results.each {|r|
  puts "#{r[:title]} - #{r[:link]}"
}
```

IRB example



面倒臭くない？

google-search2.rb

```
require 'open-uri'
require 'hpricot'
require 'cgi'

page = open('http://www.google.com/search?q=ruby')
html = page.read
doc = Hpricot(html)
results = (doc/"//div/h2/a").map {|a|
  { :title => CGI.unescapeHTML(a.inner_text), :link => a.attributes['href'] }
}

results.each {|r|
  puts "#{r[:title]} :: [#{r[:link]}]"
}
```

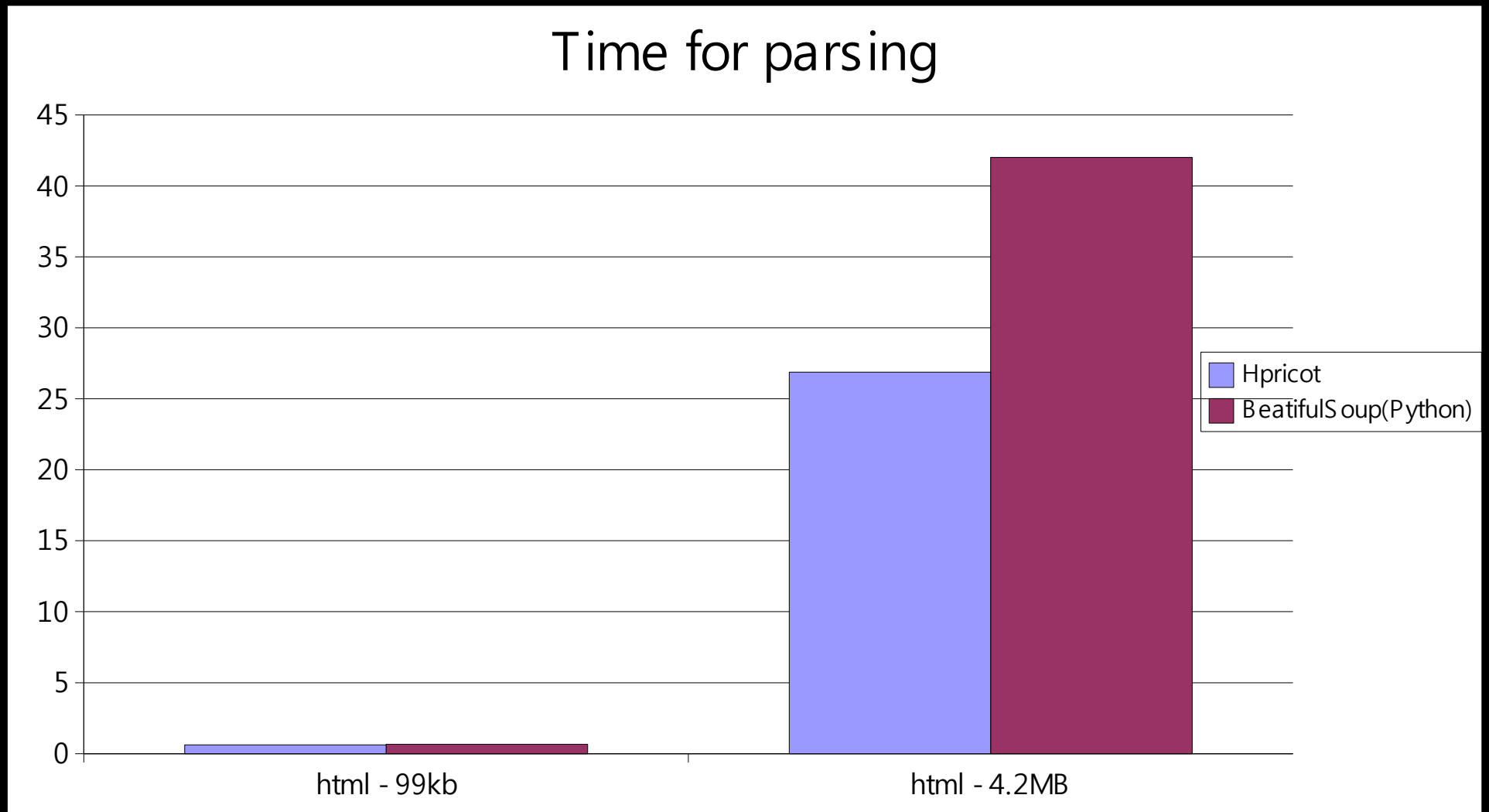
IRB example



Hpricotとは?

- why the lucky stiff 製作
- スキャナはRageIで生成
- JQueryと互換
- 早い!

Hpricotの速度



Hpricotの使い方

- XPath
- CSS Selector

XPath for Hpricot

- XML Path Language
- XML ツリー構造を表現
- ex) `doc.search("/html/body/div/a")`
- ex) `doc/"//div[@class='content']"`
- 詳しくは Hpricot ドキュメント
<http://code.whytheluckystiff.net/hpricot/wiki/SupportedXPathExpressions>

IRB example



Firefox - firebugを利用

- Firebug : Firefox 拡張
- Inspect という機能

Google™ [Adv Pre](#)

[New!](#) [View and manage your web history](#)

Web [Blogs](#) [Code](#) [Groups](#) [Video](#) Results 1 - 10 of about 88,800,000 for **ruby** [[definition](#)]. (0.04 seconds)

[Ruby Programming Language](#)
 Interpreted, dynamically typed, pure object-oriented, scripting language for fast, easy programming, from Japan. Simple, straightforward, extensible.
www.ruby-lang.org/ - 13k - 6 Jun 2007 - [Cached](#) - [Similar pages](#) - [Note this](#)
[Ruby Home Page - Download Ruby](#) - www.ruby-lang.org/en/20020102.html
[Downloads](#) - www.ruby-lang.org/en/downloads/
[Documentation](#) - www.ruby-lang.org/en/documentation/

Inspect Edit | a.l < h2.r < div.g < div < div#res < body

Console	HTML	CSS	Script	DOM	Net	Options	Style	Layout	DOM	Options
<pre> <div> <div class='g'> <h2 class='r'> </h2> </div> <table collapsing='0' colloading='0' border='0'> </pre>						<pre> a:link, .w, a:w:link, .w a:link, .q:visited, .q:link, .q:active, .q { color: #0000CC; } </pre>				



ruby

[Code](#) [Groups](#) [Video](#)

Programming Language

... dynamically typed, pure object-oriented ...
... from Japan. Simple, straightforward ...
... g.org/ - 13k - 6 Jun 2007 - [Cached](#) ...
... [Home Page](#) - [Download Ruby](#) - [www](#) ...
... [ads](#) - [www.ruby-lang.org/en/downlo](#) ...
... [entation](#) - [www.ruby-lang.org/en/do](#)

Object Edit | a.l < h2.r <

HTML CSS Script DOM N

```
<div>  
  <div class='g'>  
    <h2 class='r'>  
      <a class='l' href='h  
    </h2>  
  <table cellpadding='0' cellspacing='0' border='0'>
```

- Copy HTML
- Copy innerHTML
- Copy XPath**
- Log Events
- Scroll Into View
- New Attribute...
- Edit Attribute "href"...
- Delete Attribute "href"
- Edit HTML...
- Delete Element
- Inspect in DOM Tab

S

[View and manage yo](#)

by [[definition](#)]. (0.0

Search

Layout DOM

```
.q {  
  color: #0000CC;  
}
```

CSS selector

- XPath より短く、読みやすい。
- ex) `doc.search('#menu').inner_html`
 - `<div id='menu'>`
- ex) `doc/'div.sidebar'`
 - `<div class="sidebar">`

IRB example



HTTPを操る

- 簡単な物は `open-uri`
- 複雑な場合 `Net::HTTP`
- もっと楽な方法は？

Mechanize

Mechanize

- PerlのWWW::Mechanizeからの影響
- Michael Newman & Aaron Paterson 製作
- 内部では Net::HTTP理用している
- Hpricotを使用する

Mechanizeの動作

- HTTP request
- 返事により redirectなどの処理
- Cookieがあったらそれを処理
- HTMLをパーズング

yahoo-login.rb

```
require 'mechanize'
require 'config'

agent = WWW::Mechanize.new
page = agent.get('https://login.yahoo.co.jp/config/login')
puts "Cookie (before login)"
puts agent.cookies.to_s
login_form = page.forms.with.name('login_form').first
login_form.login = $ID
login_form.passwd = $PASSWD
agent.submit(login_form)

puts
puts "Cookie (after login)"
puts agent.cookies.to_s
```

scRUBYt

scRUBYt

- Web Extractionのためのライブラリ
- MechanizeとHpricotを利用する
- DSL 記法
- 今も活発に開発中

google-search3.rb

```
google_data = Scrubyt::Extractor.define do
  #Perform the action(s)
  fetch 'http://www.google.com/ncr'
  fill_textfield 'q', 'ruby'
  submit
  #Construct the wrapper
  link "Ruby Programming Language" do
    url "href", :type => :attribute

  end
  next_page "Next", :limit => 2
end

google_data.to_xml.write($stdout, 1)
google_data.export(__FILE__)
```

export?

- サンプル文字列をXPathに変換
- Ruby2Rubyのおかげ

Web scrapingの応用

- 2ch 掲示版に連続書き込む
- サーバに無理なrequestをしてさせる
- 掲示版ずっと同じ書き込みをする

良い大人は真似し
ちゃいけません

Web scrapingの応用

- rssがないサイトのrssをつくるスクリプト
- サーチエンジンのcrawler
- railsアプリケーションの最終テスト

ライブラリのインストール

- `gem install hpricot mechanize scrubyt -y`

質 問