

File: haha/testsdasesss.rb

```
1 # -*- coding: utf-8 -*-
2 require "prawn"
3
4 class Certificate
5
6   def initialize(aluno = nil)
7
8     Prawn::Document.generate("#{aluno}.pdf", :page_size => "A5", :page_layout => :landscape, :background => "cert_bg.resized.jpg") do
9       fill_color "ffffff" #"40464e"
10      text "Ruby Metaprogramming", :size => 40, :style => :bold, :align => :center
11
12      move_down 30
13      text "Certificado", :size => 24, :align => :center, :style => :bold
14
15      move_down 30
16      text "Certificamos que <b>#{aluno}</b> participou...", :inline_format => true
17
18      move_down 15
19      text "São Paulo, #{Time.now.strftime("%d/%m/%Y")}."
20
21      move_down 30
22      # pdf.font Rails.root.join("fonts/custom.ttf")
23      text "howto", :size => 24
24
25      move_up 5
26      font "Helvetica"
27      text "http://howtocode.com.br", :size => 10
28    end
29  end
30 end
31
32 end
33
34 =begin
35 alunos = ["123 da Silva 4", "Adoniran Barbosa", "Pedro Paulo Pereira"]
36
37 alunos.each do |aluno|
38   Certificate.new(aluno)
39 end
40 =end
41
42 File.open("nomes").each { |line| Certificate.new(line) }
43
44
```

File: haha/teste2.rb

```
1 #!/usr/bin/ruby
2 #
3 # code2pdf
4 # Created by Lucas Caton at 2011, may 31
5 #
6
7 require 'optparse'
8 require 'yaml'
9 require 'rubygems'
10 require 'prawn'
11 require 'code2pdf/convert_to_pdf'
12
13 options = {}
14
15 optparse = OptionParser.new do |opts|
16   opts.banner = "Usage: code2pdf <project path> <blacklist file>\n\nYou can use flags as such:"
17
18   opts.on('-h', '--help', 'Display this screen') do
19     puts opts
20     exit
21   end
22
23   if ARGV.size < 2
24     puts opts
25     exit
26   end
27 end
28
29 begin
30   optparse.parse!
31 rescue OptionParser::InvalidOption => e
32   puts e
33   exit
34 end
35
36 PATH = ARGV[0].gsub(/\$/ , '')
37
38 # Check if path exists
39 if !File.exists?(PATH) || FileTest.file?(PATH)
40   puts "#{PATH} path does not exist or it is not a directory."
41   exit
42 end
43
44 BLACK_LIST_YML_FILE = ARGV[1]
45
46 # Check if blacklist file exists
47 if !File.exists?(BLACK_LIST_YML_FILE) || FileTest.directory?(BLACK_LIST_YML_FILE)
48   puts "#{BLACK_LIST_YML_FILE} file does not exist or it is not a regular file."
49   exit
50 end
51
52 BLACK_LIST = YAML.load File.read(BLACK_LIST_YML_FILE)
53
54 # Check if blacklist file has directories and files keys
55 if !BLACK_LIST.keys.include?(:directories) || !BLACK_LIST.keys.include?(:files)
56   puts "#{BLACK_LIST_YML_FILE} file has no directories and files keys."
57   exit
58 end
59
60 BLACK_LIST_DIRECTORIES = BLACK_LIST[:directories].map {|directory| "#{PATH}/#{directory}"}
61 BLACK_LIST_FILES = BLACK_LIST[:files]
62
```

```

63 @code_files = []
64
65 # Read code files from path
66 def read_code_files_from_path(path)
67   Dir.foreach path do |it|
68     path_and_name = "#{path}/#{it}"
69     if FileTest.directory?(path_and_name) && it != '.' && it != '..'
70       read_code_files_from_path path_and_name unless BLACK_LIST_DIRECTORIES.include? path_and_name
71     elsif FileTest.file?(path_and_name) && !BLACK_LIST_FILES.include?(it)
72       puts "Processing => #{path_and_name}"
73       file = File.open(path_and_name, 'r')
74       file_content = ''
75       line_number = 1
76       file.each_line do |line|
77         file_content << line.gsub(/</, '&lt;'); line.gsub(/>/, "<color rgb='AAAAAA'>#{line_number}</color> "); line_number += 1
78       end
79       file.close
80       @code_files << [path_and_name, file_content]
81     end
82   end
83 end
84
85 read_code_files_from_path PATH
86
87 # Convert to PDF
88 puts "\n\nConverting to PDF..."
89 filename = PATH.gsub(/(\.|\//), '_')
90 pdf = ConvertToPDF.new "#{filename}.pdf", @code_files
91 pdf.save

```