

*Gfdnavi*の実装



実装の細かなことを話されても、
どうせよく分からないだろうし、
興味もわかない



Rails? それって難しそうだし
データベースって馴染無いし
使うにはマウスだけでいいし
開発は開発する人に任せればいいし



そんな人にもなんとなくでも分かるように
Rails の “MVC” を
Gfdnavi の描画機能を例にとり
できるだけ分かりやすく (?) 解説



MVC

⇒ M: Model

- パラメータや条件を受け取って、必要なものを返すもの
- Railsでは大抵の場合、データベース(のラッパー)

⇒ V: View

- 見た目を作るもの
- Railsではhtmlを生成するもの

⇒ C: Controller

- 作業の流れをコントロールするもの



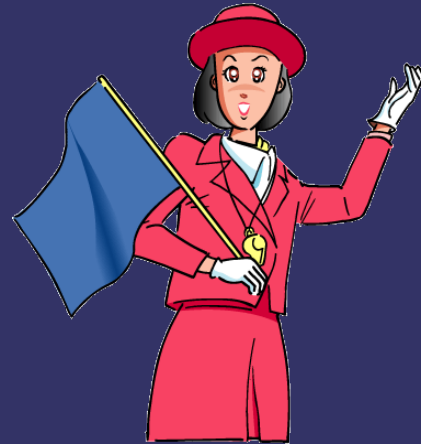
Gfdnavi描画機能におけるMVC

- ⇒ Model
 - 描画パラメータを受け取って、Vizshotを返す
- ⇒ View
 - 描画パラメータを設定するためのフォームの生成
 - 結果の図の表示
- ⇒ Controller
 - それらの手順をコントロールする





User



Controller
app/controllers



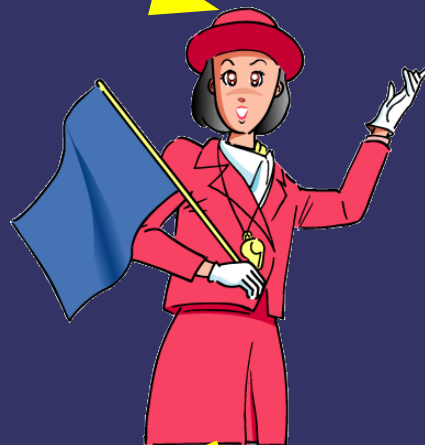
View
app/views

Model
app/models





変数選択



フォーム表示

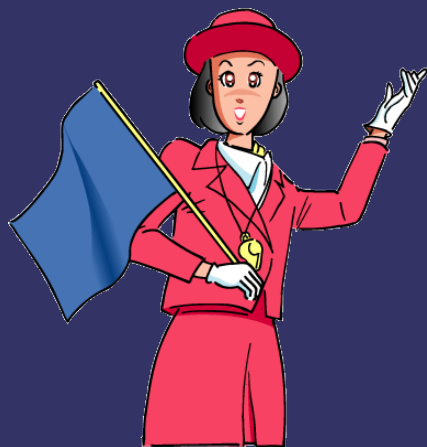


パラメータ設定ページ
作成指示



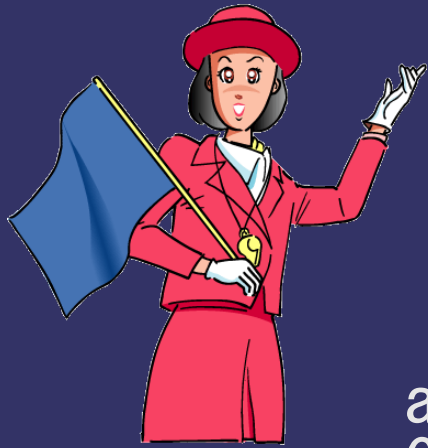


Analysis機能のページに移動する
<http://host:port/path/analysis/index>



上記URIをもとに、
Analysis Controller の
index アクションが
呼び出される





アクション名と同じ名前の
メソッドが呼ばれる

```
app/controller/analysis_controller.rb  
Class AnalysisController < ApplicationController  
  def index  
    @variables = all_variables() #解析用変数リスト  
  end  
end
```



アクション用のメソッドが終了すると、
自動的に View に、
Analysis の index 用のページの作成が
指示される





アクション名と同じ名前の テンプレートが使われる

```
gfdnavi/app/view/analysis/index.rhtml  
<div id=" variables" >変数リスト</div>  
<div id=" action_form" >  
  <div id=" axes" >軸設定つまみ</div>  
  <div id=" options" >オプション設定フォーム</div>  
</div>
```



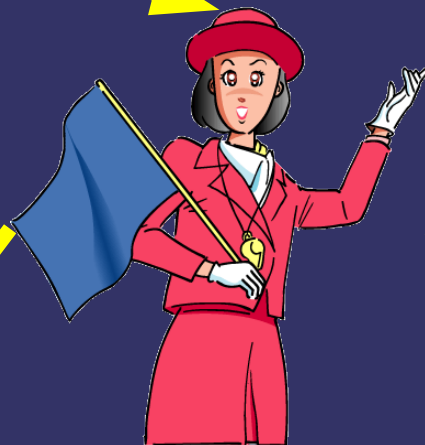
描画パラメータ設定
drawボタンクリック



図表示



図の表示指示



パラメータ

Vizshot

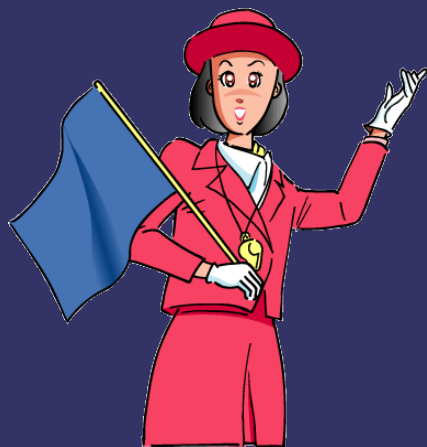




描画用パラメータを設定し、
draw ボタンをクリック

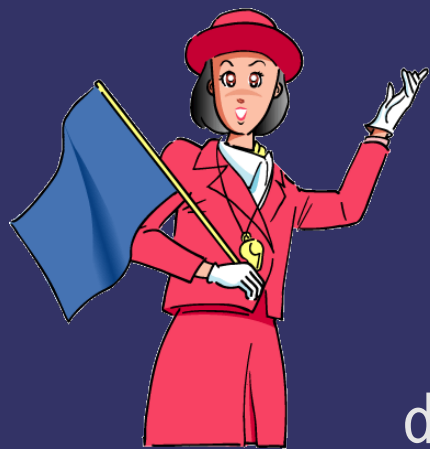
Ajax により、

`http://host:port/path/analysis/execut`
が呼び出される



Analysis Controller の
execute アクションが
呼び出される





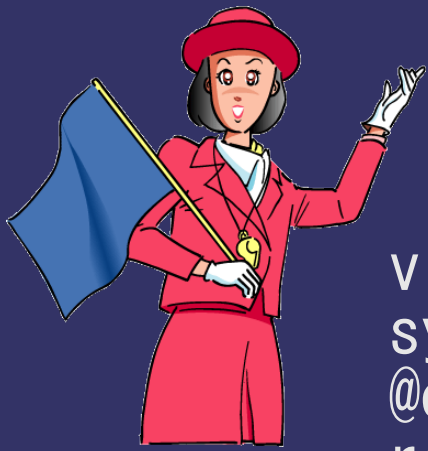
execute メソッドが呼ばれる

```
def execute  
  analysis = Analysis.new( params[:analysis] )  
  vizshot = analysis.get_vizshots( "temp.rb" )
```



パラメータをもとに、
Analysis Model が作られる
vizshot を作成し、
コントローラに返す





Vizshot から得られたruby コードを実行し、
図を作成する

```
vizshot.dump_code( nil, true, :image_dump=>true )  
system( "ruby temp.rb" )  
@diagram = [id, path, diagram_id, saved?, vizshot]  
render(:update) { |page|  
  page.replace_html :diagrams,  
                    render(:partial=>" diagram" )  
}  
end
```



render メソッドを呼出し、
View にページの作成を指示





ページの diagrams という id を持った
エレメントを置き換える
_diagram テンプレートを使用

```
gfdnavi/app/view/analysis/_diagram.rhtml
```

```
<%  
id, path, diagram_id, saved, vizshot = @diagram  
%>
```

```
<table id=" <%=id%>_table"><tr>  
  <td id=" <%=id%>" ><%= image_tag(path) %></td>  
</tr></table>
```



まとめ

- ⇒ MVCを理解すれば全体の流れが分かる
 - 変更したいときに、変更すべき箇所が分かる
 - Model: データの管理
 - View: 見た目を作る
 - Controller: 作業の流れを制御

