Ruby On Rails course

CRUD, Forms & Bootstrap

REST

Create method

```ruby
class UsersController < ApplicationController
 def create
 end
end
```

Read method

```
class UsersController < ApplicationController

 def show

 end

end
```

Update method

```
class UsersController < ApplicationController
 def update
 end
end
```

Delete method

```
class UsersController < ApplicationController
 def destroy
 end
end
```

```ruby
class UsersController < ApplicationController
 def create
 end
 def show
 end
 def update
 end
 def destroy
 end
end
```

Index method

```ruby
class UsersController < ApplicationController
 def index
 end
end
```

New method

```
class UsersController < ApplicationController

 def new

 end

end
```

Edit method

```
class UsersController < ApplicationController
 def edit
 end
end
```

7 Methods → 7 Routes

Index, New, Create, Show, Edit, Update, Destroy

```
   get 'users'             => 'users#index'
   get 'users/new'         => 'users#new'
  post 'users'             => 'users#create'
   get 'users/:id'         => 'users#show'
   get 'users/:id/edit'    => 'users#edit'
   put 'users/:id'         => 'users#update'
delete 'users/:id'         => 'users#destroy'
```

```
resources :users
```

resources :users

```
   Prefix  Verb    URI Pattern               Controller#Action

    users  GET     /users                    users#index
           POST    /users                    users#create
 new_user  GET     /users/new                users#new
edit_user  GET     /users/:id/edit           users#edit
     user  GET     /users/:id                users#show
           PATCH   /users/:id                users#update
           PUT     /users/:id                users#update
           DELETE  /users/:id                users#destroy
```

Nested resources

```
get 'subreddits/:id', to: ‘subreddits#show', as: :subreddit

get 'subreddits/:subreddit_id/posts/:id', to: ‘posts#show’,
as: :subreddit_post
```

```
resources :subreddits, only: [:show] do
  resources :posts, only: [:show]
end
```

Index method

app/controllers/users_controller.rb

```
class UsersController < ApplicationController

 def index
   @users = User.all
 end

end
```

app/views/users/index.html.erb

```
<table>
  <thead>
    <tr>
      <td>Name</td>
    </tr>
  </thead>
  <tbody>
    <% @users.each do |user| %>
      <tr>
        <td><%= prettify_name(user.name) %></td>
      </tr>
    <% end %>
  </tbody>
</table>
```

Helpers

app/helpers/users_helper.rb

```ruby
module UsersHelper

 def prettify_name(msg)
  ['awesome', 'beautiful', 'sexy'].sample + " #{msg}”
 end

end
```

Forms

Two types of form helpers

- form_tag (basic form)
- form_for (binding to an object)

form_tag

```
<%= form_tag(‘/contact’) do %>
  <%= text_field_tag :subject %>
   <%= text_field_tag :email %>
   <%= text_field_tag :content %>
   <%= submit_tag :send %>
<% end %>
```

More: password_field_tag, text_area_tag, hidden_field_tag…

form_for

```
<% @contact = Contact.new %>
<%= form_for @contact do |f| %>
  <%= f.text_field :subject %>
   <%= f.text_field :email %>
   <%= f.text_field :content %>
   <%= f.submit :send %>
<% end %>
```

New method

app/controllers/users_controller.rb

```ruby
class UsersController < ApplicationController

 def new
   @user = User.new
 end

end
```

app/views/users/new.html.erb

```
<%= form_for @user do |f| %>
  <%= f.text_field :name %>
   <%= f.text_field :email %>
   <%= f.submit :send %>
<% end %>
```

Create method

```
class UsersController < ApplicationController

 def create
   @user = User.new(user_params)

   if @user.save
     redirect_to users_path
   else
     render :new
   end
 end

 def user_params
   params.require(:user).permit(:name, :email)
 end

end
```

redirect_to examples

- redirect_to ‘http://infnum.co'
- redirect_to ‘http://www.google.com/favicon.ico'
- redirect_to users_path
- redirect_to edit_user_path(1)
- redirect_to :back
- ...
- Notice: redirect loop

render examples

- render :new - (will search the new.html.erb view)
- render json: @users
- render xml: @users
- ...

app/views/users/new.html.erb

```
<%= form_for @user do |f| %>
  <%= f.text_field :name %>
   <%= f.text_field :email %>
   <%= f.submit :send %>
   <% f.object.errors.full_messages.each do |error| %>
     <%= error %>
   <% end %>
 <% end %>
```

Edit method

app/views/users/index.html.erb

```
<table>
  <thead>
    <tr>
      <td>Name</td>
    </tr>
  </thead>
  <tbody>
    <% @users.each do |user| %>
      <tr>
        <td><%= prettify_name(user.name) %></td>
        <td><%= link_to ‘Edit’, edit_user_path(user) %></td>
      </tr>
    <% end %>
  </tbody>
</table>
```

app/controllers/users_controller.rb

```
class UsersController < ApplicationController

 def edit
   @user = User.find(params[:id])
 end

end
```

app/views/users/edit.html.erb

```
<%= form_for @user do |f| %>
  <%= f.text_field :name %>
   <%= f.text_field :email %>
   <%= f.submit :send %>
   <% f.object.errors.full_messages.each do |error| %>
     <%= error %>
   <% end %>
 <% end %>
```

Update method

```
class UsersController < ApplicationController

 def update
   @user = User.find(params[:id])

   if @user.update(user_params)
     redirect_to users_path
   else
     render :edit
   end
 end

 def user_params
   params.require(:user).permit(:name, :email)
 end

end
```

app/views/users/new.html.erb

```
<h1>New user</h1>

<%= form_for @user do |f| %>

  <%= f.text_field :name %>

  <%= f.text_field :email %>

  <%= f.submit :send %>

<% end %>
```

app/views/users/edit.html.erb

```
<h1>Edit user</h1>

<%= form_for @user do |f| %>

  <%= f.text_field :name %>

  <%= f.text_field :email %>

  <%= f.submit :send %>

<% end %>
```

Partials

Simple view shared among many pages

app/views/users/_form.html.erb

```
<%= form_for @user do |f| %>
  <%= f.text_field :name %>
   <%= f.text_field :email %>
   <%= f.submit :send %>
   <% f.object.errors.full_messages.each do |error| %>
     <%= error %>
   <% end %>
 <% end %>
```

app/views/users/new.html.erb

```
<h1>New user</h1>

<%= render ‘form’ %>
```

app/views/users/edit.html.erb

```
<h1>Edit user</h1>

<%= render ‘form’ %>
```

Destroy method

app/views/users/index.html.erb

```
<table>
 <thead>
  <tr>
   <td>Name</td>
  </tr>
 </thead>
 <tbody>
  <% @users.each do |user| %>
   <tr>
    <td><%= prettify_name(user.name) %></td>
    <td><%= link_to ‘Edit’, edit_user_path(user) %></td>
    <td><%= link_to ‘Delete’, user_path(user), method: :delete %></td>
   </tr>
  <% end %>
 </tbody>
</table>
```

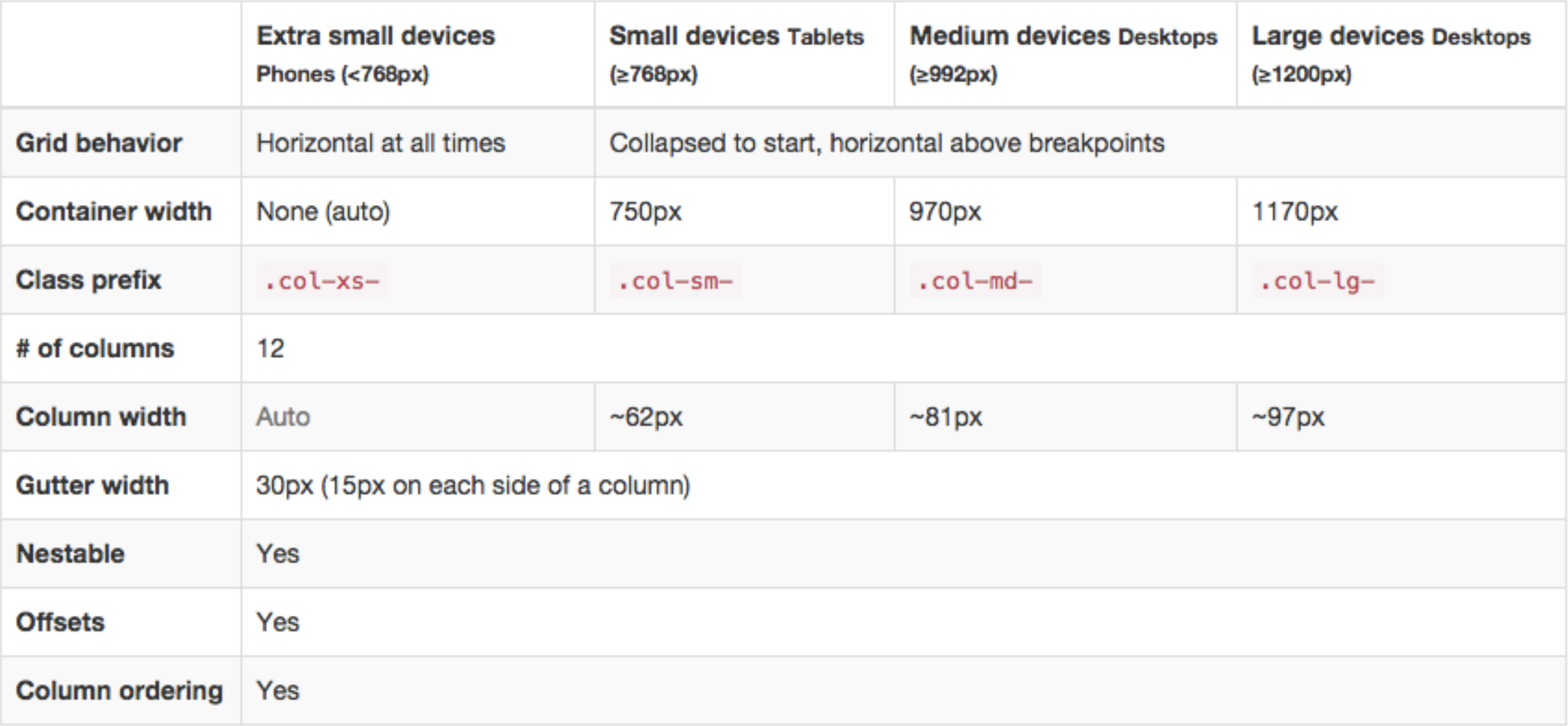
app/controllers/users_controller.rb

```ruby
class UsersController < ApplicationController

 def destroy
   User.find(params[:id]).destroy

   redirect_to users_path
 end

end
```

Bootstrap

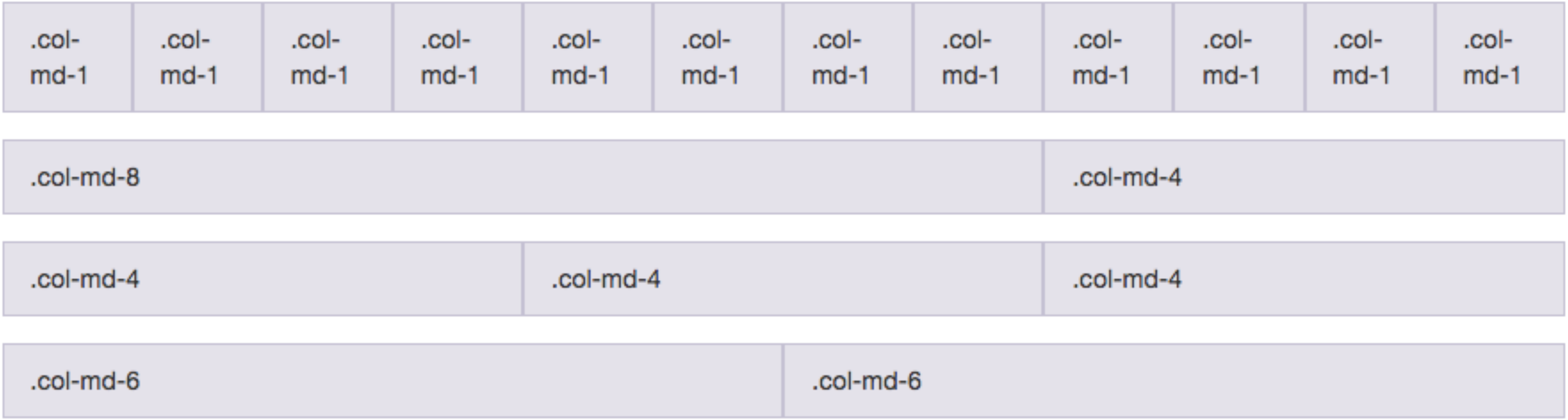
Bootstrap is the most popular H™L, CSS, and JS framework for developing responsive, mobile first projects on the web.

DEVICES



	Extra small devices Phones (<768px)	**Small devices** Tablets (≥768px)	**Medium devices** Desktops (≥992px)	**Large devices** Desktops (≥1200px)
Grid behavior	Horizontal at all times	Collapsed to start, horizontal above breakpoints		
Container width	None (auto)	750px	970px	1170px
Class prefix	`.col-xs-`	`.col-sm-`	`.col-md-`	`.col-lg-`
# of columns	12			
Column width	Auto	~62px	~81px	~97px
Gutter width	30px (15px on each side of a column)			
Nestable	Yes			
Offsets	Yes			
Column ordering	Yes			

GRID



<table>
<tr><td>.col-md-1</td><td>.col-md-1</td><td>.col-md-1</td><td>.col-md-1</td><td>.col-md-1</td><td>.col-md-1</td><td>.col-md-1</td><td>.col-md-1</td><td>.col-md-1</td><td>.col-md-1</td><td>.col-md-1</td><td>.col-md-1</td></tr>
<tr><td colspan="8">.col-md-8</td><td colspan="4">.col-md-4</td></tr>
<tr><td colspan="4">.col-md-4</td><td colspan="4">.col-md-4</td><td colspan="4">.col-md-4</td></tr>
<tr><td colspan="6">.col-md-6</td><td colspan="6">.col-md-6</td></tr>
</table>

AND MORE

- Forms
- Tables
- Buttons
- Labels
- Icons
- Navigation
- …

Bootstrap as a gem

Add to Gemfile

- gem ‘bootstrap-sass’

Run

- bundle install

Restart application

app/views/layouts/application.html.erb

```
<!DOCTYPE html>
<html>
<head>
  <title>Academy</title>
  <%= stylesheet_link_tag    'application', media: 'all', 'data-turbolinks-track' => true %>
  <%= javascript_include_tag 'application', 'data-turbolinks-track' => true %>
  <%= csrf_meta_tags %>
</head>
<body>

<%= yield %> /* yield identifies a section where content from the view should be inserted */

</body>
</html>
```

Assets

`app/assets/javascripts/application.js`

- `//= require bootstrap`

`app/assets/stylesheets/application.scss`

- `@import ‘bootstrap’;`

Assets pipeline

The asset pipeline provides a framework to concatenate and minify or compress JavaScript and CSS assets. It also adds the ability to write these assets in other languages and pre-processors such as CoffeeScript, Sass and ERB.

Assets pipeline

- gem ‘sprocket-rails’

- Precompile
- Concatenate
- Minify
- Uglify

HW4 Design

https://projects.invisionapp.com/share/7W3IJXKPA#/screens/88472787?maintainScrollPosition=false

Student academy promo video