Rails Tutorial Chapter 13 summary by Kevin M

==========//CHAPTER THIRTEEN==========

Text vs String - Text is essentially the same, but doesn't have the same character limit (255) as String.
There is no performance difference in production.

Setting up the posts
rails generate model Micropost content:text user:references
user:references refers to the fact that it's a user-based resource.
It'll generate the model file with a line with belongs_to :user
In the migration, we add an index for user_id and created_at, since we'll be listing them in reverse date order.
add_index :microposts, [:user_id, :created_at]

belongs_to/has_many
Instead of microposts.create, microposts.new, it's user.microposts.create and so on-- as long as you specify that posts belongs_to :user, and that user has_many :microposts.
That means, instead of using this to associate a post with a user:

```
@user = users(:michael)
# This code is not idiomatically correct.
@micropost = Micropost.new(content: "Lorem ipsum", user_id: @user.id)
```

we can do this:

```
@user = users(:michael)
@micropost = @user.microposts.build(content: "Lorem ipsum")
```

(build is used here so that it doesnt modify the database)

If we change users has_many to this: ||| has_many :microposts, dependent: :destroy |||, that will make it so that microposts are both dependent upon users, and destroyed when their user is destroyed.

Default Scope
default_scope -> { order(created_at: :desc) }
This line of code (placed in the microposts model file) instructs rails to view posts as ordered from the most recent to the oldest. Look, it uses a lambda function!

Displaying Posts

We use a similar method to getting a lot of users to display on the same page-- that is, with partials!

```
<li id="micropost-<%= micropost.id %>">
  <%= link_to gravatar_for(micropost.user, size: 50), micropost.user %>
  <span class="user"><%= link_to micropost.user.name, micropost.user %></span>
  <span class="content"><%= micropost.content %></span>
  <span class="timestamp">
      Posted <%= time_ago_in_words(micropost.created_at) %> ago.
  </span>
</li>
```

Then, on the users_controller, we added

```
@microposts = @user.microposts.paginate(page: params[:page])
```

to the show method. This makes it so that the posts are displayed on the user's page when you add this div to the user show.html.erb page:

```
<div class="col-md-8">
    <% if @user.microposts.any? %>
      <h3>Microposts (<%= @user.microposts.count %>)</h3>
      <ol class="microposts">
          <%= render @microposts %>
      </ol>
      <%= will_paginate @microposts %>
    <% end %>
</div>
```

```
def feed
  Micropost.where("user_id = ?", id)
end
```

This is the code that we use to display microposts in a little feed, after some modifications. The important takeaway is that user_id = ?, which escapes characters to protect the database. It's just an int, so no big deal, but it's an important thing to know for other stuff.

==========CHAPTER THIRTEEN\\==========