



# The Assets Pipeline

# What are assets?

CSS  
JS  
images  
videos  
fonts  
...

# Organizing your assets

app/assets  
lib/assets  
vendor/assets



# What does the pipeline do?

concatenating  
minifying/compressing  
preprocessing



# What does the pipeline do?

concatenating  
minifying/compressing  
preprocessing



public/assets

# Caching and fingerprinting

Browsers cache assets to speed things up

`application.js`



# Caching and fingerprinting

Browsers cache assets to speed things up

~~application.js~~



# Caching and fingerprinting

Browsers cache assets to speed things up

~~application.js~~

application-908e25f4bf641868d8683022a5b62f54.js



# Caching and fingerprinting

Browsers cache assets to speed things up

~~application.js~~

application-908e25f4bf641868d8683022a5b62f54.js

```
<%= javascript_include_tag "application" %>
```

# Manifest files

Used to let the pipeline know what files to include



# Manifest files

Used to let the pipeline know what files to include

`app/assets/javascripts/application.js`

`app/assets/stylesheets/application.css.scss`



# Manifest files

These files contain directives to specify what to compile



# Manifest files

These files contain directives to specify what to compile

`require_self`



# Manifest files

These files contain directives to specify what to compile

`require_self`

`require`



# Manifest files

These files contain directives to specify what to compile

`require_self`

`require`

`require_tree`

# Cool, but...

What about other types of assets?





# Cool, but...

What about other types of assets?

```
<%= image_tag("image.jpg") %>
```

# Cool, but...

What about other types of assets?

```
<%= image_tag("image.jpg") %>
```



```

```

# Cool, but...

What about other types of assets?

```
<%= image_tag("image.jpg") %>
```



```

```

```

```

# Using ERB

Adding the .erb extension to your css files



# Using ERB

Adding the .erb extension to your css files  
`application.css.erb`



# Using ERB

Adding the .erb extension to your css files

`application.css.erb`

lets you use helpers like:



# Using ERB

Adding the .erb extension to your css files

`application.css.erb`

lets you use helpers like:

```
.class { background-image: url(<%= asset_path 'image.png' %>) }
```



# Using ERB

The same works for your javascript





# Using ERB

The same works for your javascript  
`application.js.erb`



# Using ERB

The same works for your javascript

`application.js.erb`

lets you use helpers like:



# Using ERB

The same works for your javascript

`application.js.erb`

lets you use helpers like:

```
$('#logo').attr({ src: "<%= asset_path('logo.png') %>" });
```

# Development vs Production

In development assets are served as separate files in the order they are specified in the manifest file.

# Development vs Production

In development assets are served as separate files in the order they are specified in the manifest file.

```
<script src="/assets/core.js?body=1"></script>  
<script src="/assets/projects.js?body=1"></script>  
<script src="/assets/tickets.js?body=1"></script>
```

# Development vs Production

In production by default Rails assumes assets have been precompiled and will be served as static assets by your web server.



# Development vs Production

In production by default Rails assumes assets have been precompiled and will be served as static assets by your web server.

```
<script src="/assets/  
application-908e25f4bf641868d8683022a5b62f54.js"></script>
```

```
<link href="/assets/  
application-4dd5b109ee3439da54f5bdfd78a80473.css" media="screen"  
rel="stylesheet" />
```



# Precompiling assets

To precompile the assets





# Precompiling assets

To precompile the assets

```
rake assets:precompile
```

