

Cours Ruby on Rails

Installation de Ruby on Rails :

Pour commencer avec Rails, vous devez d'abord installer Ruby sur votre système. Ensuite, vous pouvez installer Rails en utilisant la commande suivante :

```
bash
```

```
gem install rails
```

Création d'une nouvelle application Rails :

Pour créer une nouvelle application Rails, utilisez la commande suivante :

```
bash
```

```
rails new NomDeVotreApplication
```

Structure d'une application Rails :

- **app/controllers** : Contient les contrôleurs qui gèrent la logique des requêtes HTTP.
- **app/models** : Contient les modèles qui représentent les données de l'application.
- **app/views** : Contient les vues qui définissent la présentation des données.
- **config** : Contient les fichiers de configuration de l'application.
- **db** : Contient les migrations de base de données.
- **public** : Contient les fichiers statiques (CSS, JavaScript, images).
- **test** : Contient les tests automatisés.

Création d'un modèle, d'une migration et d'une base de données :

Pour créer un modèle, une migration associée, et mettre à jour la base de données, utilisez les commandes suivantes :

```
bash
```

```
rails generate model NomDuModele attribut1:type attribut2:type  
rails db:migrate
```

Création d'un contrôleur et de vues :

Pour créer un contrôleur avec des actions et des vues associées, utilisez les commandes suivantes :

```
bash
```

```
rails generate controller NomDuControleur action1 action2
```

Routes dans config/routes.rb :

Définissez les routes pour rediriger les requêtes vers les actions du contrôleur. Par exemple :

```
ruby
```

```
Rails.application.routes.draw do
  get 'welcome/index'
  root 'welcome#index'
end
```

Utilisation de vues avec HTML et ERB :

Dans les fichiers de vue (dans le répertoire `app/views`), vous pouvez utiliser des fichiers HTML avec l'inclusion de code Ruby à l'aide d'ERB (Embedded Ruby).

```
html

<!-- app/views/welcome/index.html.erb -->
<h1>Bienvenue sur mon site Rails</h1>
<p><%= @message %></p>
```

Utilisation du contrôleur pour gérer les requêtes :

```
ruby

# app/controllers/welcome_controller.rb
class WelcomeController < ApplicationController
  def index
    @message = "Ceci est un message provenant du contrôleur."
  end
end
```

Utilisation des modèles pour interagir avec la base de données :

```
ruby

# app/models/article.rb
class Article < ApplicationRecord
end
```

Utilisation de formulaires :

```
html

<!-- app/views/articles/new.html.erb -->
<%= form_with scope: :article, url: articles_path, local: true do |form| %>
  <%= form.label :title %>
  <%= form.text_field :title %>

  <%= form.label :content %>
  <%= form.text_area :content %>

  <%= form.submit %>
<% end %>
```

Contrôleurs pour traiter les formulaires :

```
ruby

# app/controllers/articles_controller.rb
class ArticlesController < ApplicationController
  def new
    @article = Article.new
  end
end
```

```

end

def create
  @article = Article.new(article_params)
  if @article.save
    redirect_to @article
  else
    render 'new'
  end
end

private

def article_params
  params.require(:article).permit(:title, :content)
end
end

```

Utilisation des partials pour réutiliser des vues :

Les partials sont des fichiers partiels réutilisables qui peuvent être inclus dans d'autres vues.

```

ruby

<!-- app/views/shared/_header.html.erb -->
<h1>Mon Super Site</h1>

ruby

<!-- app/views/layouts/application.html.erb -->
<!DOCTYPE html>
<html>
  <head>
    <title>Mon Super Site</title>
  </head>
  <body>
    <%= render 'shared/header' %>
    <%= yield %>
  </body>
</html>

```

Utilisation des relations entre modèles :

```

ruby

# app/models/user.rb
class User < ApplicationRecord
  has_many :articles
end

# app/models/article.rb
class Article < ApplicationRecord
  belongs_to :user
end

```

Déploiement de l'application :

Pour déployer une application Rails, vous pouvez utiliser des services tels que Heroku ou déployer sur des serveurs tels que Nginx ou Apache avec Passenger.

