② 静的ファイルの追加

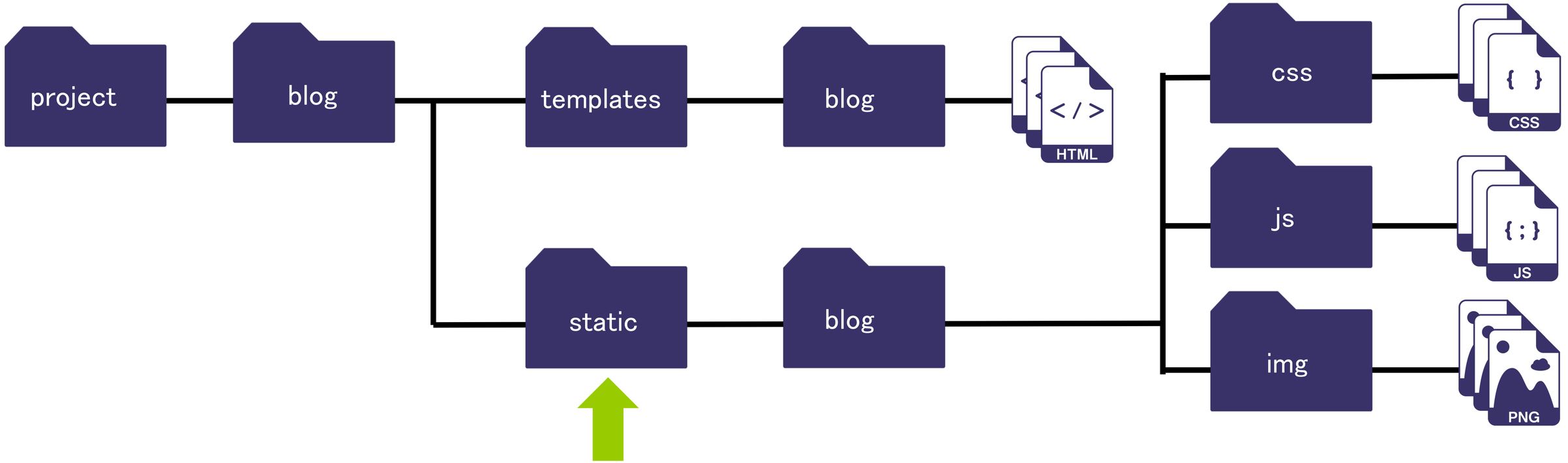
ここではまず、静的ファイルを追加する方法を学習しましょう。

静的なファイル

- これまでに、Djangoのテンプレートエンジンを用いて、テンプレートから動的にHTMLを生成する手順を学習しました
- 一方で、以下のようなファイルは必ずしも動的に生成する必要はありません
  - 画像ファイル
  - CSSファイル
  - JavaScriptファイル
- このような「静的なファイル」として、予め用意したファイルを、リクエストに応じてそのまま返す方が効率的です

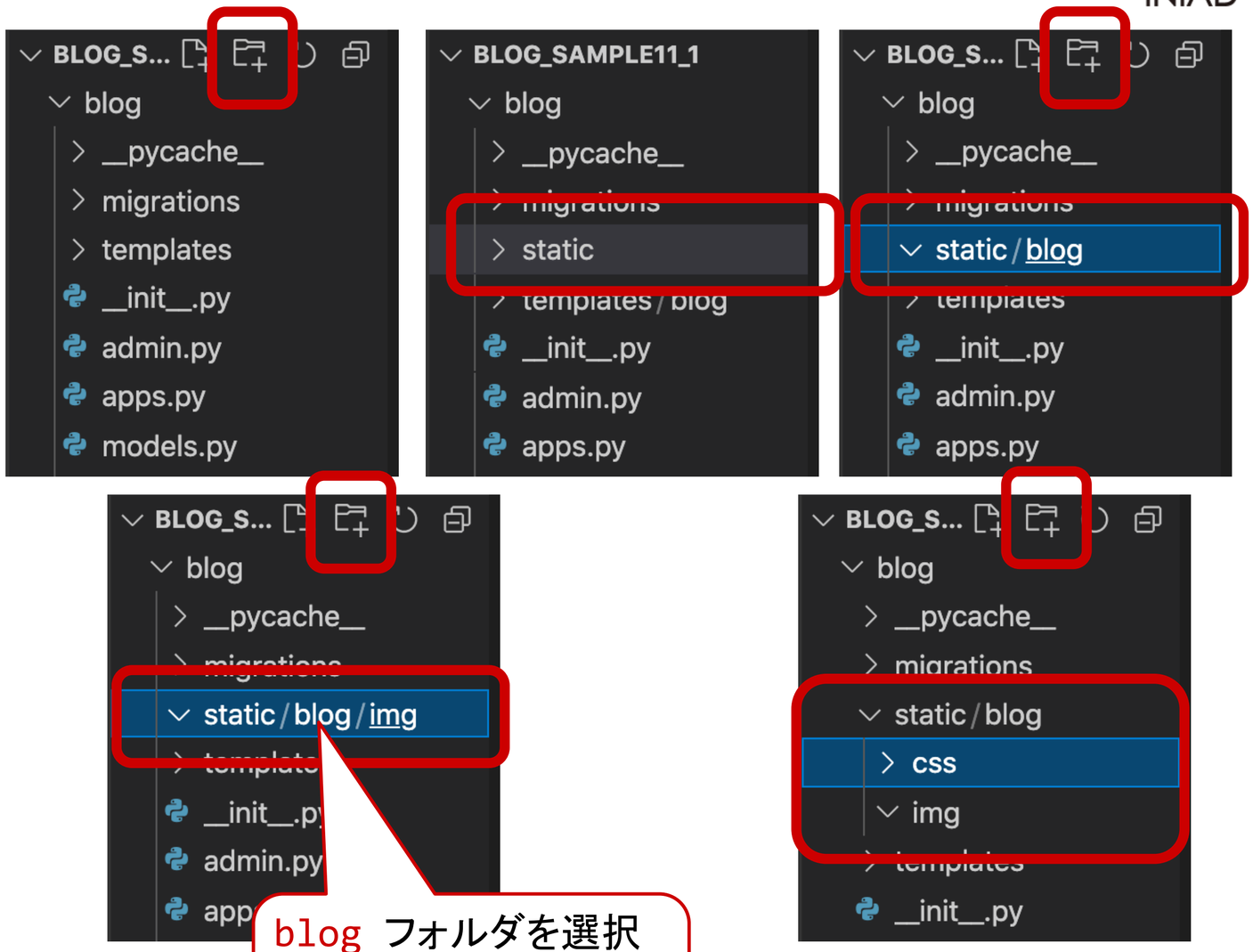
Djangoにおける静的ファイル

- Djangoでは、動的にファイルを生成する仕組みだけでなく、静的にファイルを返す仕組みも用意されています
- 各アプリケーション直下の `static` ディレクトリに配置する必要があります



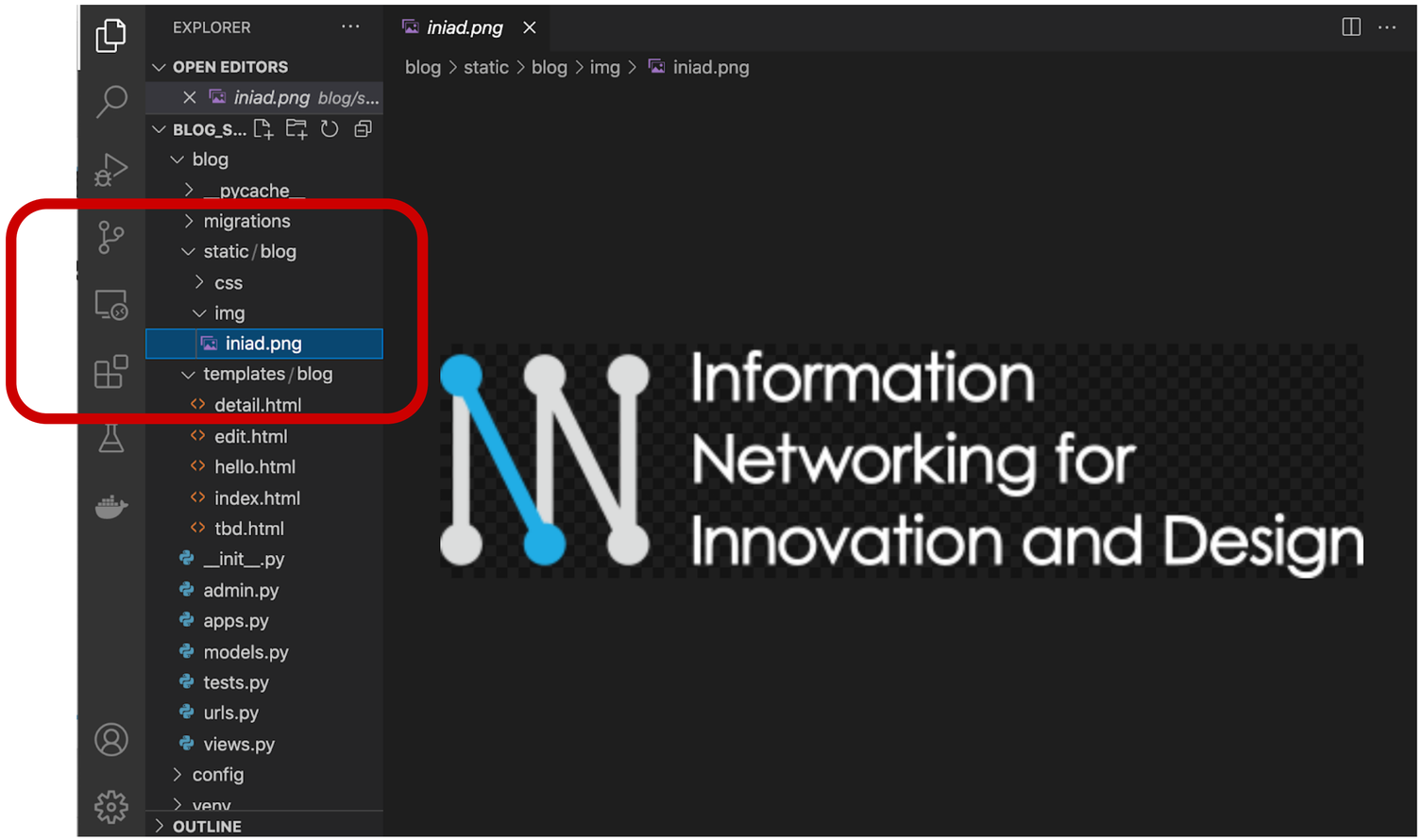
フォルダを作成する

1. blog フォルダに static フォルダを作る
2. static フォルダ（1.で作ったもの）の中に blog フォルダを作る
3. blog フォルダ（2.で作ったもの）に img フォルダを作成する
4. blog フォルダ（2.で作ったもの）に css フォルダを作成する



画像ファイルを保存する

- MOOCsのリンクから、`iniad.png` という画像をダウンロードし、`blog/static/blog/img` フォルダに移動します



(再掲)テンプレート言語：その他のタグ

- URL
  - `{% url '名前' 変数,変数... %}`
    URLディスパッチャ内の、指定した名前(`name=`で指定した値)に対応するパターンに置き換える
    - 変数をパターンに代入する
  - 例）
    - `{% url 'detail' article.id %}`
      : `'detail'`という名前のURLパターンに置き換える
      → /数字

- その他、サンプルで利用しているタグ
  - `{% load static %}`
    静的ファイルを読み込めるようにする
  - `{% static 'パス' %}`
    静的ファイルのURLに置き換える
  - `{% csrf_token %}`
    フォーム内で、CSRF対策のためのトークンを出力する

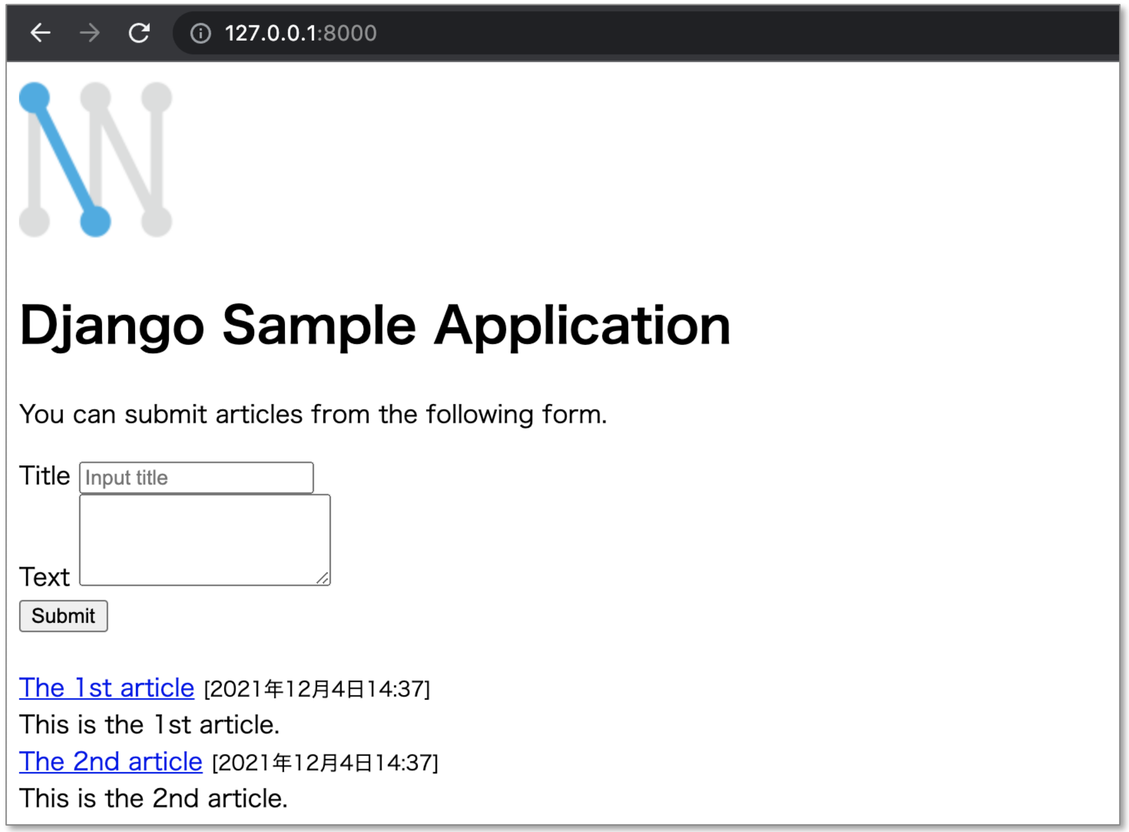
index.html を修正する

- ためしに、一覧表示画面に画像が表示されるようにしてみます
- blog/templates/blog/index.html を以下のように編集します

```
{% load static %}
<!DOCTYPE html>
<html>
  <head>
    <meta charset="UTF-8">
  </head>
  <body>
    <div id="header">
      <img src="{% static 'blog/img/iniad.png' %}" alt="Logo">
    </div>
    <h1>Django Sample Application</h1>
    <p>You can submit articles from the following form.</p>
    <form action="{% url 'index' %}" method="post">
      {% csrf_token %}
```

動作確認

- Djangoサーバを起動し、一覧表示画面を確認しましょう
  - ここまでの手順が正しく完了していれば、画像が表示されるはずです！



http://127.0.0.1:8000/