

ブログ記事詳細ページを作ろう

最後のレッスンです。ここでは、各ブログ記事の詳細を表示するページを作成し、トップページからリンクで飛べるようにしましょう。

トップページにリンクを作る

まずは、トップページに表示されている各記事に「記事を読む」というリンクを追加しましょう。ユーザーがこのリンクをクリックした時に、その記事の詳細ページに飛ぶようにします。



リンクを作るので、aタグを追加します。以下のように追加してください。

django_blog/blogs/templates/blogs/index.html

```
<!DOCTYPE html>
<html>
<head>
  <title>ブログ管理サイト</title>
</head>
<body>
  <h1 style="text-align: center;">My Blog</h1>
  <div style="width: 70%;margin: 0px auto;">
    <hr />
    {% for blog in blogs %}
    <div>
      <h3>{{ blog.title }}</h3>
      <div>{{ blog.text | truncatechars:100 }}</div>
      <div style="text-align: right;">
        <a href="{% url 'blogs:detail' blog_id=blog.id %}">記事を読む</a>
      </div>
    </div>
  </div>
</body>
</html>
```

```
</div>
</div>
<hr />
{% endfor %}
</div>
</body>
</html>
```

aタグのhrefの中身に注目してください。hrefの中には通常リンクを記述しますが、見慣れない書き方をしていますね。でも、{% %}が何を意味するかはわかりますよね。そうです、テンプレートタグです。つまりここではDjango特有の書き方でURLを指定していることになります。

具体的にそれぞれのコードが何を意味しているのかを説明します。まず、`url` `'blogs:detail'`の部分では、このリンクがblogsアプリケーション内のurls.pyで「detail」と名前をつけたURLにリンクすることを示しています。urls.pyには、まだこのURLを作成していないのであとで作ります。

`blog_id=blog.id`では、変数`blog_id`に各ブログインスタンスのidである数字を代入しています。（`blog.title`で各ブログのタイトルを取得しているのと同様に、`blog.id`で各ブログのidが取得できるので、それを変数に代入しています。）こうすることでid情報（数字）をurls.pyに渡すことができます。

各記事の詳細ページを作る

トップページに、記事詳細ページへのリンクを作ることができたので、いよいよ記事詳細ページを作っていきます。

新しいHTMLファイルを表示させるために必要なことは覚えていますか。「アプリケーションで機能を作ろう」のレッスンで学習しましたね。以下3つが必要になります。

- URLを設定する（urls.py）
- Viewを作る（views.py）
- HTMLファイルを作る

では、URLを作成しましょう。urls.pyファイルを開いて、このようにpathを追加してください。

django_blog/blogs/urls.py

```
from django.urls import path
from . import views

app_name = 'blogs'
urlpatterns = [
    path('', views.index, name='index'),
    path('detail/<int:blog_id>/', views.detail, name='detail'),
]
```

1 目目のpathよりちょっとだけ複雑になってますね。

第一引数の `'detail/<int:blog_id>/'` ではURLを指定しています。 `int` は、ここに何かしらの数字が入ることを示しています。トップページからのリンクで、変数 `blog_id` には数字が代入された状態で情報を受け取っているの、ここにはその数字が自動的に入力されることとなります。また、ここに入った数字は変数 `blog_id` に代入されてViewに渡されます。これで、URL `http://127.0.0.1:8000/detail/1` のページでは `id=1` の記事の詳細ページを表示する準備をしています。ただし、まだViewやHTMLファイルを作っていないので、現段階でこのURLにアクセスしてもエラーが出ます。

第二引数は、1つ上のpathと同様のことをしています。このURLでは、`views.py` ファイルの `detail` 関数が処理されることを意味しています。

第三引数は、このURLに `detail` という名前をつけています。名前をつけることで、`記事を読む` のリンクをクリックした時にこのpathに飛ぶようになります。

Viewを作る

いま作ったpathでは、`views` ファイルの `detail` 関数で記事詳細ページを表示する処理がされるように設定しましたので、`detail` 関数を作りましょう。

`django_blog/blogs/views.py`

```
from django.shortcuts import render
from .models import Blog

...

def detail(request, blog_id):
    blog = Blog.objects.get(id=blog_id)
    return render(request, 'blogs/detail.html', {'blog': blog})
```

`index` 関数と違って、`request` 以外にも `blog_id` という引数がありますね。先ほど設定したpathの第一引数の、変数 `blog_id` を受け取っています。おさらいですが、この変数には数字が代入されています。

`blog = Blog.objects.get(id=blog_id)` では、`id` が `blog_id` の数字と一致する `Blog` インスタンスをデータベースから取得して、変数「`blog`」に代入しています。

`render` 関数で、変数「`blog`」を `detail.html` に渡しています。

これで、Viewは完成です！それでは最後に、記事詳細ページ（`detail.html`）を作りましょう。

Templateを作る

トップページ(`index.html`)を作ったときと同様、`templates/blogs` の中にhtmlファイルを作ってください。今回作るのは、`detail.html` というファイルですよね。

作ったらそのファイルを開いてください。このファイルには何を表示させたらよいでしょうか。

記事の詳細ページですので、記事が持つ情報を全て表示させましょう。Blogインスタンスは「タイトル、本文、作成日、更新日」のフィールドを持っています。「インスタンス.フィールド名」で情報を表示させていきます。先ほど作ったViewで変数blogにインスタンスを代入してますので、「blog.フィールド名」で表示できます。

django_blog/blogs/templates/blogs/detail.html

```
{{ blog.title }}
{{ blog.text }}
{{ blog.created_datetime }}
{{ blog.updated_datetime }}
```

これで、ページの原型ができました。ただ、このままだとこのページからトップページに戻ることができないので、リンクを作ってください。

リンクの貼り方はさっき勉強しましたよね。urls.pyでトップページにはindexという名前をつけているので、以下のようなaタグを書けばOKです。

django_blog/blogs/templates/blogs/detail.html

```
{{ blog.title }}
{{ blog.text }}
{{ blog.created_datetime }}
{{ blog.updated_datetime }}
<a href="{% url 'blogs:index' %}">トップページに戻る</a>
```

最後にちょっとだけデザインを整えましょう。

django_blog/blogs/templates/blogs/detail.html

```
<!DOCTYPE html>
<html>
<head>
  <title>ブログアプリ</title>
</head>
<body>
  <h1 style="text-align: center;">{{ blog.title }}</h1>
  <div style="width: 70%;margin: 0px auto;">
    <div>
      <div style="margin: 60px 20px;">
        {{ blog.text }}
      </div>
      <div style="text-align: center; margin-top: 50px">
        <a href="{% url 'blogs:index' %}">トップページに戻る</a>
      </div>
    </div>
  </div>
</body>
</html>
```

Congratulations !

これでブログ管理サイトの完成です！本当におつかれさまでした！！

今回作ったのは非常に簡単なサイトですが、このチュートリアルでDjango開発の基礎をほとんど抑えることができています。あとは少しずつ応用を利かせていくことでもっと高度なアプリケーションを作ることができるようになります。

チュートリアルが終わったらぜひ次にしてほしいことがあります。それは「オリジナルのサイトを自分で1から作ってみる」ことです。難しいものを作る必要はありません。このチュートリアルとほぼ同じもので大丈夫です。例えば、好きなスポーツチームの選手を一覧で表示→選手ごとの紹介ページを作る、とかなんでもOKです！

1からつくすることで、自分がまだきちんと理解できていないところが明白になり、とても良い復習になるはずです。

DjangoBrothersでは、他のチュートリアルも用意していますので復習が終わったらこちらにも挑戦してみてくださいね！