

Ask Company



리액트와 함께 장고 시작하기 / 장고 Forms

HTML Form

여러분의 파이썬/장고 페이스메이커가 되겠습니다.

언제 쓰나요?

HTML Form (클라이언트 측)

클라이언트에서 사용자에게 입력폼을 제공하고, 이를 서버로 전송코자 할 때

Django Form (서버 측)

클라이언트로부터 전달받은 값들에 대한 유효성 검사를 수행하고, 이를 데이터베이스에 저장하는 등의 처리

HTML Form을 생성하는 기능을 제공합니다. 이를 활용하거나, 인터페이스만 맞춰 직접 HTML Form을 코딩해도 OK.

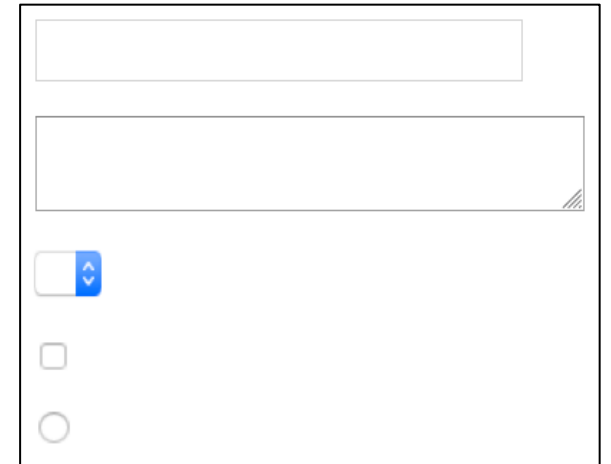
HTML Form

HTML 웹페이지에서는 <form></form> 태그를 통해, 입력폼을 구성하고, submit시에 지정 action URL로 데이터 전송을 시도

ex) 로그인 폼, 댓글 폼

하나의 <form>태그는 하나 이상의 Widget을 가집니다.

```
<form action="" method="POST">  
  <input type="text" />      <!-- 1줄 문자열 입력 -->  
  <textarea></textarea>    <!-- 1줄 이상의 문자열 입력 -->  
  <select></select>        <!-- 항목 중 택일 -->  
  <input type="checkbox" />     <!-- 체크박스 (한 그룹 내 여러 항목을 다수 선택 가능) -->  
  <input type="radio" />    <!-- 라디오박스 (한 그룹 내 여러 항목 중에 하나만 선택 가능) -->  
  그 외 다수 위젯  
</form>
```

A visual representation of an HTML form. It contains a single-line text input field at the top, followed by a multi-line text area. Below the text area is a dropdown menu with a blue arrow icon. At the bottom, there is a checkbox and a radio button.

HTML <form> 태그 필수 속성

action : 요청을 보낼 주소

method : 전송 방식

"GET" : 주로 데이터 조회 요청 시에 사용

"POST" : 파괴적인 액션(생성/수정/삭제)에서 사용

enctype : 인코딩 방식

POST 요청에서만 유효

GET 요청에서는 한 enctype으로 강제됨.

<form>의 enctype

"application/x-www-form-urlencoded" (디폴트)

GET 요청에서는 이 유형이 강제됨.

인자들을 "URL 인코딩"을 수행하여, QueryString 형태로 전달

→ 파일 업로드 불가

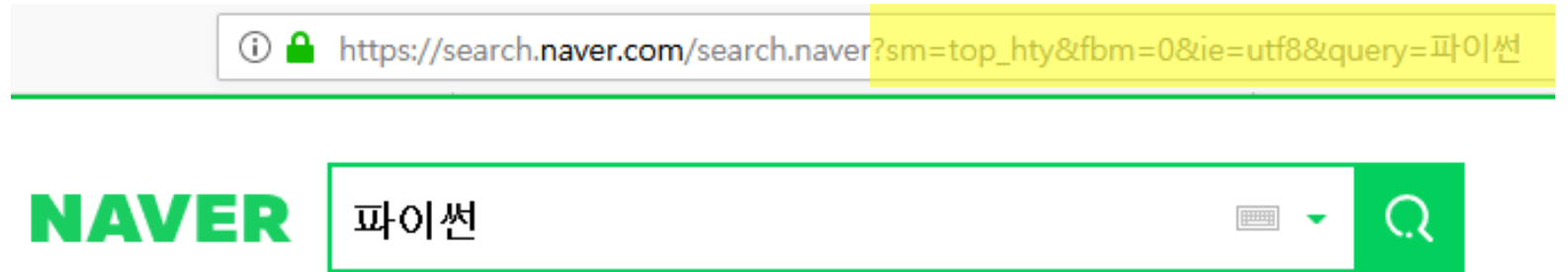
"multipart/form-data"

파일 업로드 가능.

"text/plain"

실제로 사용되지는 않는 유형

공백만 "+"로 변환



url encode

key=value 값의 쌍이 & 문자로 이어진 형태

공백은 +로 인코딩하며, Special 문자들은 ascii 16진수 문자열, UTF8 인코딩 16진수 문자열로 변환

```
from urllib.parse import urlencode
print(urlencode({'key1': 'value1', 'key2': 10, 'name': '방탄소년단'}))
print('방탄소년단'.encode('utf8'))
print(''.join('%{:X}'.format(ch) for ch in '방탄소년단'.encode('utf8')))
```

```
key1=value1&key2=10&name=%EB%B0%A9%ED%83%84%EC%86%8C%EB%85%84%EB%8B%A8
b'\xeb\xba\x9d\xed\x83\x84\xec\x86\x8c\xeb\x85\x84\xeb\x8b\xa8'
%EB%B0%A9%ED%83%84%EC%86%8C%EB%85%84%EB%8B%A8
```

Ask Company

Form Method

Form 요청에서 인자 보내는 2가지 방법

1) 요청 URL 뒤에 ? 를 붙이고, 인자를 실어서 보내기.

x-www-form-urlencoded 인코딩의 값만 실을 수 있습니다.

이 인자를 Query String 인자라고 부릅니다.

GET요청에서 주로 사용하기에 GET 인자라고도 합니다.

POST 요청에서도 요청 URL 뒤에 QueryString을 실을 수 있습니다.

2) 요청 Body에 모든 인코딩의 인자를 실어서 보냅니다.

x-www-form-urlencoded 인코딩의 값도 OK

multipart/form-data 인코딩의 값도 OK → 파일 업로드 가능

2가지 Form Method

GET 방식

엽서에 비유. 물건을 보낼 수 있다.

application/x-www-form-urlencoded 만이 강제되며, 인코딩된 문자열을 QueryString으로 전달

POST 방식

택배에 비유. 다양한 물건을 보낼 수 있다.

다양한 인코딩을 모두 사용 가능하며 (application/x-www-form-urlencoded 및 multipart/form-data), 인코딩된 데이터를 "요청 Body"에 담아서 전달

GET 요청 예시

enctype은 붙여도 의미가 없기에 붙이지 않습니다.

디폴트 : application/x-www-form-urlencoded로 강제

주로 검색폼에서 인자를 넘길 때 씁니다. → 조회 목적

O

```
<form method="GET" action="">
  <input type="text" name="query" />
  <input type="submit" value="검색" />
</form>
```

X

```
<form method="GET" action="">
  <input type="text" name="title" />
  <textarea name="content"></textarea>
  <input type="file" name="photo" />
  <input type="submit" value="저장" />
</form>
```

POST 요청 예시

파일 업로드를 수행한다면, 필히 enctype을 multipart/form-data로 지정 →
오타에 유의.

```
<form method="POST" action="" enctype="multipart/form-data">  
  <input type="text" name="title" />  
  <textarea name="content"></textarea>  
  <input type="file" name="photo" />  
  <input type="submit" value="저장" />  
</form>
```

장고 뷰에서의 인자 접근

request.GET

모든 QueryString 인자 목록

QueryString을 파싱한 QueryDict 객체

GET/POST 요청에서 모두 가능

request.POST

POST 요청에서만 가능.

파일 내역은 제외한 모든 POST인자 목록

"요청 BODY"를 파싱한 QueryDict 객체

request.FILES

POST 요청에서만 가능.

"요청 BODY"에서 파일내역만 파싱한 MultiValueDict 객체

Life is short.
You need Python and Django.

I will be your pacemaker.

