

5장. 폴과 view 함수



calc 앱 생성하기

➤ calc 앱 만들기

1. calc 앱 생성

```
(mysite) C:\projects\polls> django-admin startapp calc
```

2. URL 매핑 설정 – config/urls.py

```
urlpatterns = [  
    (... 생략 ...)  
    path('calc/', include('calc.urls')),  
]
```

calc 앱 생성하기

3. URL 매핑 설정 – calc/urls.py

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

app_name = 'calc'

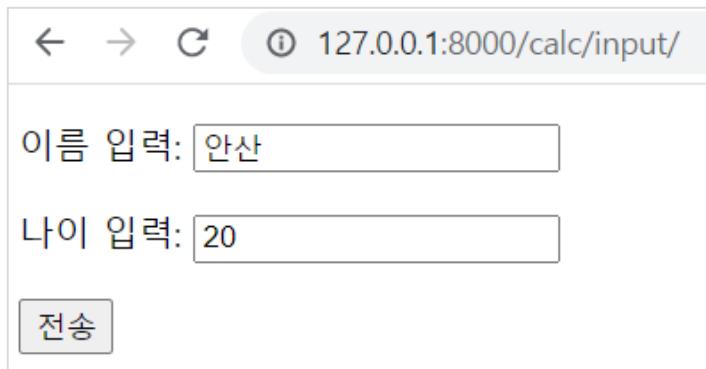
urlpatterns = [
    # 숫자, 문자 넘기기
    path('input/', views.input, name='input'),
    path('output/', views.output, name='output'),

    # 홀수/짝수 판정
    path('input_num/', views.input_num, name='input_num'),
    path('even_odd/', views.even_odd, name='even_odd'),

    # 구구단
    path('input_dan/', views.input_dan, name='input_dan'),
    path('gugudan/', views.gugudan, name='gugudan'),
]
```

폼의 데이터 입 / 출력 제어

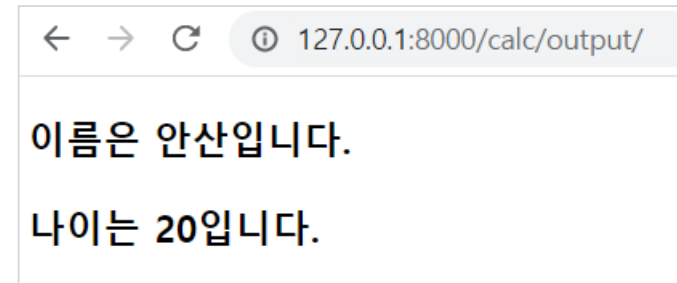
- 데이터 입 / 출력 제어



← → ↻ ⓘ 127.0.0.1:8000/calc/input/

이름 입력:

나이 입력:



← → ↻ ⓘ 127.0.0.1:8000/calc/output/

이름은 안산입니다.

나이는 20입니다.

데이터 넘기고 받기

1. 제어 함수 만들기 – calc/views.py

```
def input(request):  
    # 입력  
    return render(request, 'calc/input.html')  
    #return HttpResponse("입력 페이지입니다.")  
  
def output(request):  
    # 출력  
    name = request.POST['name']  
    age = int(request.POST['age'])  
    context = {'name': name, 'age': age,}  
    return render(request, 'calc/output.html', context)
```

데이터 넘기고 받기

2. 템플릿 html 만들기 – templates/calc/input.html

```
<form action="{% url 'calc:output' %}" method="post">
    {% csrf_token %}
    <p><label>이름 입력: </label>
    <input type="text" name="name"></p>
    <p><label>나이 입력: </label>
    <input type="text" name="age"></p>
    <input type="submit" value="전송">
</form>
```

3. templates/calc/output.html

```
<h3>이름은 {{ name }}입니다.</h3>

<h3>나이는 {{ age }}입니다.</h3>
```

홀수/짝수 판별 프로그램

- 홀수/짝수 판별 프로그램

← → ↻ ⓘ 127.0.0.1:8000/calc/input_num/

숫자 입력:



← → ↻ ⓘ 127.0.0.1:8000/calc/even_odd/

홀수/짝수 판정

결과는 홀수입니다.

홀수/짝수 판별 프로그램

1. 제어 함수 만들기 – calc/views.py

```
def input_num(request):  
    # 숫자 입력  
    return render(request, 'calc/input_num.html')  
  
def even_odd(request):  
    # 홀수/짝수 계산  
    num = int(request.POST['num'])  
    if num % 2 == 1:  
        context = {'result': "홀수입니다."}  
    else:  
        context = {'result': "짝수입니다."}  
    return render(request, 'calc/even_odd.html', context)
```


홀수/짝수 판별 프로그램

2. 템플릿 html 만들기 – templates/calcul/input_num.html

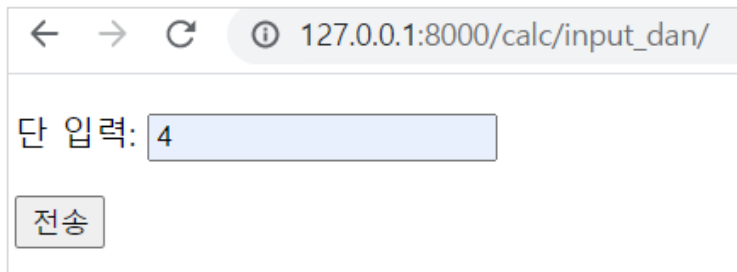
```
<form action="{% url 'calcul:even_odd' %}" method="post">
    {% csrf_token %}
    <p><label>숫자 입력: </label>
    <input type="text" name="num"></p>
    <input type="submit" value="전송">
</form>
```

3. templates/calcul/even_odd.html

```
<h2>홀수/짝수 판정</h2>
{% if result %}
<h3>결과는 {{ result }}</h3>
{% endif %}
```

구구단 프로그램

- 구구단 프로그램



A screenshot of a web browser window. The address bar shows the URL `127.0.0.1:8000/calc/input_dan/`. Below the address bar, there is a text input field labeled "단 입력:" with the number "4" entered. Below the input field is a button labeled "전송" (Submit).



4단

4 x 1 = 4
4 x 2 = 8
4 x 3 = 12
4 x 4 = 16
4 x 5 = 20
4 x 6 = 24
4 x 7 = 28
4 x 8 = 32
4 x 9 = 36

구구단 프로그램

1. 제어 함수 만들기 – calc/views.py

```
def input_dan(request):  
    # 단 입력  
    return render(request, 'calc/input_dan.html')  
  
def gugudan(request):  
    # 구구단 계산  
    dan = int(request.POST['dan'])  
    times = []  
    for i in range(1, 10):  
        times.append(dan * i)  
    context = {'dan': dan, 'times': times}  
    return render(request, 'calc/gugudan.html', context)
```

구구단 프로그램

2. 템플릿 html 만들기 – templates/calcul/input_dan.html

```
<form action="{% url 'calcul:gugudan' %}" method="post">
    {% csrf_token %}
    <p><label>단 입력: </label>
    <input type="text" name="dan"></p>
    <input type="submit" value="전송">
</form>
```

3. templates/calcul/gugudan.html

```
<h3>{{ dan }}단</h3>
{% for val in times %}
    {{ dan }} x {{ forloop.counter }} = {{ val }}<br>
{% endfor %}
```