Writing More Views

View Function

A view function/class, or view for short, is a Python function that takes a web request and returns a web response.

```
from django.http import HttpResponse

def index(request):
   return HttpResponse("<h1>Welcome to my homepage!</h1>")
```

HttpResponse

`HttpResponse` is used to provide an inbound HTTP request to a Django web application with a text response.

```
HttpResponse("Response here")
```

urls.py

URL routes are configured in **urls.py** by calling the `path()` function to the `urlpatterns` list.

```
urlpatterns = [
  path(""home/"", home_view),
  path(""about/"", about_me_view),
]
```

Dynamic URLs

Dynamic URLs can be created by using angle brackets (< , >) to capture named parameters inside of a `path()` function.

```
# URLconfig
urlpatterns = [
  path("profile/<str:name>",
profile_view),
]

# views.py
def profile_view(request, name):
  return HttpResponse(f"Hey there! My name
is {name}")
```

render() Function

The `render()` functions take three arguments- the request variable, the template path, and optionally a dictionary object to pass data to the template.

```
render(request, "template.html", { "data":
your_data })
```

404 Exception

An `Http404` exception can be raised inside a view function and returns a built-in 404 page provided by Django.

```
from django.http import Http404

def blog_view(request, slug):
  try:
    entry = Blog.objects.get(slug=slug)
  except Blog.DoesNotExist:
    raise Http404()
  return render(request, "blog/post.html",
{"blog": entry})
```

Dynamic URLs in Templates

The `name` parameter can be provided inside the `path()` function in order to dynamically render it within the route's template. The route will be accessible via the named parameter that's passed on.

```
<a href="{% url 'name_of_path' %}">Contact Me</a>
```