

Maarit Parkkonen (TKMI17SM)

Django -frameworkin käyttöönotto

Django -oppimisprojekti
Ohjelmoinnin erikoiskurssi

2020



**Kaakkois-Suomen
ammattikorkeakoulu**

SISÄLLYS

1	KEHITYSYMPÄRISTÖN ESITTELY	3
2	ASENNUS	5
3	KÄYTTÖÖNOTTO: HELLO WORLD!	6
	LÄHTEET	11
	LIITTEET	

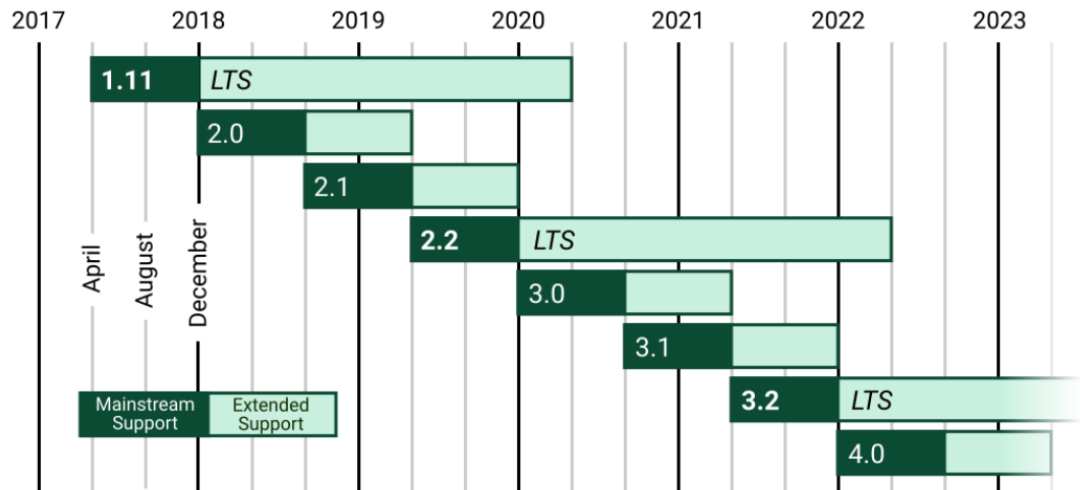
Liite 1. Hello World

1 KEHITYSYMPÄRISTÖN ESITTELY

Django on Pythonilla toteutettu sovelluskehys verkkosovellusten kehitykseen. Sen toteutus käynnistyi vuonna 2003 Lawrence Journal-World sanomalehden sisäisenä projektina mutta avoimeksi lähdekoodiksi projekti julkaistiin heinäkuussa 2005 BSD-lisenssillä. Django 1.0 julkaistiin 2008. Django'n alkuperäisiin kehittäjiin kuuluivat Adrian Holovaty ja Simon Willison sekä myöhemmin Jakob Kaplan-Moss. Django on saanut nimensä jazz-kitaristi Django Reinhardtilta. (Roine, 2010.)

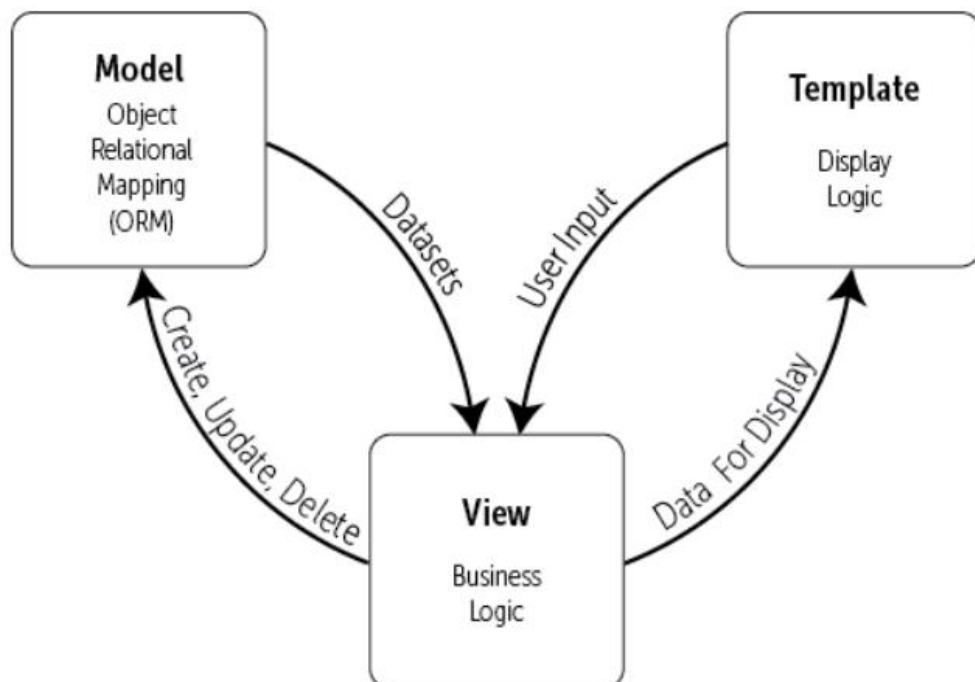
Django'n tavoitteena on yksinkertaistaa ja nopeuttaa tietokantapohjaisten verkkosovellusten kehitystä. Django Bookin kirjoittajan George Nigelin (2020a) mielestä syitä valita Django'n web-kehityksen sovelluskehikseksi ovat:

- Python on helposti opittava ohjelmointikieli, joka pohjautuu luonnollisen kielen tyyppisiin rakenteisiin. Laajasti ammattikäytössä, esimerkiksi Google käyttää sitä monissa sovelluksissaan.
- Valmiit työkalut ja ohjelmistokirjastot yleisimpiin verkkosovellusprosesseihin, kuten ylläpitoon, käyttäjien autentikointiin tai sessioiden hallintaan.
- Ei turhaa tavaraa mukana. Django projektin aloitus luo joukon lähes tyhjiä tiedostoja, pari kansiota ja pari lyhyttä aloitustiedostoa. Jokainen kehittäjä ottaa sovellukseen mukaansa vain sen mitä tarvitsee.
- Sisäänrakennettu käyttöliittymä (admin) sovelluksenhallintaan, sekä tietokannan että käyttäjien osalta.
- Helposti skaalautuva MTV -arkkitehtuurin ansiosta. Mahdollistaa tietokannan, sovelluksen ja median sijoittelun eri palvelimille. Backendin ja Frontendin lisäksi erottelee staattiset tiedostot, kuten kuvat, css, javascript, erilleen sovelluksesta.
- Toimivuus on käytössä testattu. Ollut jo yli 12 vuotta osana isoja globaaleja verkkopalveluja, esim. Instagram ja Pinterest.
- Runsaasti ilmaisia avoimenkoodin ohjelmistopaketteja. Laajin repository: <https://djangopackages.org/>
- Ylläpidosta ja kehitystyöstä vastaava avoin kehitysyhteisö on laaja ja aktiivinen, joka on julkaissut tasaisesti uusia versioita (kuva 1). Se takaa tulevaisuudessakin pitkäaikaisen tuen jatkumisen. LTS (long term support) versioiden kautta Django'n julkaisut pysyvät vakaina.
- Django'sta on kehittäjien toimesta tarjolla hyvät dokumentaatiot ja op-
paat.



Kuva 1. Django tuetut versiot (Django Software Foundation, 2005-2020).

Djangon arkkitehtuurina on MTV (model-template-view) -malli, joka pohjautuu periaatteiltaan MVC (model-view-control) -malliin. Kummassakin mallissa M eli model vastaa tietokannan toiminnallisuudesta. Djangossa se pohjautuu ORM (object-relational mapper) tekniikkaan, joka mahdollistaa relaatiotietokantataulujen riviin muuntamisen ohjelmiston olioiksi. Djangossa templatet vastaavat MVC:n näkymiä. Ne ovat sivupohjamalleja käyttäjille näkyvistä sivuista. Djangossa view hoitaa sovelluksen kontrolloinnin templatien ja modelien välillä, kuten kuvassa 2. (Nigel, 2020b)



Kuva 2. Django MTV -malli. (Nigel, 2020b)

Omasta mielestäni liittäisin tuohon malliin vielä Django URL-ohjaimen, jolla kontrolloidaan sovelluksen reititystä. Djangoissa ainakin aloittelijalle tärkeimmät ymmärrettävät tiedostot minun mukaani ovat:

- **urls.py**: reititys
- **views.py**: funktiot, joilla ohjataan koko sovellusta
- **models.py**: tietokannan mukaiset relaatio-oliot, joiden kautta views.py:n funktiot käsittelevät tietokantaa
- **forms.py**: lomakeoliot, joilla templateihin voidaan todella näppärästi liittää lomakkeita
- **templatet**: Esimerkiksi html-sivuja, joita voi ketjuttaa toistensa sisään ja joille voi tageilla välittää muuttujia.

Lisäksi Django projektinhallintaan kannattaa tutustua. Yksi projekti voisi sisältää useita sovelluksia ja niiden nimiavaruudet kannattaa pitää erillään. (Nämä huomiot lisätty kirjastoautomaattisovelluksen teon jälkeen)

2 ASENNUS

Djangoon pääohjelmointikielenä on siis Python. Olen käyttänyt soveltuvasti Pythonia aiemmilla matematiikan kursseilla. Tuolloin ohjelmointiympäristöksi itseleni asensin Spyderin (<https://www.spyder-ide.org/>), joka on tarkoitettu teolliselle laskennalle. Tiesin siis jotain Pythonin alkeista mutta halusin syventää tietojani, joten aloitin tämän oppimisprojektin lukemalla joululomalla Lappeenrannan teknillisen yliopiston Python 3 -ohjelmointioppaan (<https://lut-pub.lut.fi/bitstream/handle/10024/63381/isbn%209789522149701.pdf?sequence=1&isAllowed=y>).

Kieleen tutustumisen jälkeen vuorossa oli Django asennus (<https://docs.djangoproject.com/en/3.0/intro/install/#install-django>) :

1. Päivitin Microsoft Storesta Pythonin versioon 3.8.1 (<https://installpython3.com/windows/>) Nyt komennolla: `python` käynnistyy versio 3.8.1 IDLE.
2. Päivitin uusimman pip version: `pip install pip`
3. Latasin db-yhteyttä varten: `pip install mysqlclient`.
4. Latasin: `python -m pip install virtualenvwrapper`. Latauksen aikana varoitti, että `virtualenv.exe`, `virtualenv-clone.exe` ja `pbr.exe` ovat ladattu polkuun: `C:\Users\maari\AppData\Local\Packages\PythonSoftwareFoundation.Python.3.8_qbz5n2kfra8p0\LocalCache\local-packages\Python38\Scripts` joka ei kuulu PATH -muuttujiin.

5. Asensin Django: `python -m pip install Django`. Latauksen aikana samasta polusta tuli varoitus `sqlformat.exe` ja `django-admin.exe` kanssa. (kuva 3)
6. Kokeilin latausten toimivuutta. Kysyin django versiota: `python -m django --version` ja sain vastaukseksi 3.0.2 eli asennus vaikutti onnistuneen.

```
C:\Users\maari>python -m pip install virtualenvwrapper
Collecting virtualenvwrapper
  Downloading https://files.pythonhosted.org/packages/c1/6b/2f05d73b2d2410b48b96d3783a0034c26afa34ada5f1178d61191/virtualenvwrapper-4.8.4.tar.gz (334kB)
    | 337kB 819kB/s
Collecting virtualenv
  Downloading https://files.pythonhosted.org/packages/05/f1/2e07e8ca50e8470cc9ad56cf4291f4e041fa732070000a095fe478abf84/virtualenv-16.7.9-py2.py3-none-any.whl (3.4MB)
    | 3.4MB 6.4MB/s
Collecting virtualenv-clone
  Downloading https://files.pythonhosted.org/packages/ba/f8/50c2b7d9c9e0f5ce5b9d9a31f37c988c99acde8ded720b7b8d4011d/virtualenv-clone-0.5.3-py2.py3-none-any.whl
Collecting stevedore
  Downloading https://files.pythonhosted.org/packages/b1/el/f5d8db83f0b03f522f173c03e006c1bfb8343f0232a292ac96aa633b47a/stevedore-1.31.0-py2.py3-none-any.whl (43kB)
    | 51kB 1.7MB/s
Collecting pbr==2.1.0,>=2.0.0
  Downloading https://files.pythonhosted.org/packages/7a/db/a968f7beb9fe06901c1841cb25c9cb666ca1b9a19b11d41bbdf1126fc/pbr-5.4.4-py2.py3-none-any.whl (110kB)
    | 112kB ...
Collecting six>=1.10.0
  Downloading https://files.pythonhosted.org/packages/65/26/32b8464df2a97eddd1b56ed26b2c194606c16fe163c695a992b36c11cdf/six-1.13.0-py2.py3-none-any.whl
Installing collected packages: virtualenv, virtualenv-clone, pbr, six, stevedore, virtualenvwrapper
WARNING: The script virtualenv.exe is installed in 'C:\Users\maari\AppData\Local\Packages\PythonSoftwareFoundation.Python.3.8_qbz5n2kfra8p0\LocalCache\local-packages\Python38\Scripts' which is not on PATH.
Consider adding this directory to PATH or, if you prefer to suppress this warning, use --no-warn-script-location
WARNING: The script virtualenv-clone.exe is installed in 'C:\Users\maari\AppData\Local\Packages\PythonSoftwareFoundation.Python.3.8_qbz5n2kfra8p0\LocalCache\local-packages\Python38\Scripts' which is not on PATH.
Consider adding this directory to PATH or, if you prefer to suppress this warning, use --no-warn-script-location
WARNING: The script pbr.exe is installed in 'C:\Users\maari\AppData\Local\Packages\PythonSoftwareFoundation.Python.3.8_qbz5n2kfra8p0\LocalCache\local-packages\Python38\Scripts' which is not on PATH.
Consider adding this directory to PATH or, if you prefer to suppress this warning, use --no-warn-script-location
Running setup.py install for virtualenvwrapper... done
Successfully installed pbr-5.4.4 six-1.13.0 stevedore-1.31.0 virtualenv-16.7.9 virtualenv-clone-0.5.3 virtualenvwrapper-4.8.4

C:\Users\maari>python -m pip install Django
Collecting Django
  Downloading https://files.pythonhosted.org/packages/55/d1/8ade70e65fa157e1903fe4078305ca53b6819ab212d9fbb5755fc8ea2e/Django-3.0.2-py3-none-any.whl (7.4MB)
    | 7.4MB 3.3MB/s
Collecting pytz
  Downloading https://files.pythonhosted.org/packages/e7/f9/f0b53f88060247251bf481fa6ea62cd825bf1b11a8788e53ce5b7c8ad2/pytz-2019.3-py2.py3-none-any.whl (509kB)
    | 512kB 3.3MB/s
Collecting asgiref<3.2
  Downloading https://files.pythonhosted.org/packages/a5/ch/5a23b605a9753ebc2730c75e10f51c8ab3f012000ba8229fa36ad/asgiref-3.2.3-py2.py3-none-any.whl
Collecting sqlparse<0.2.2
  Downloading https://files.pythonhosted.org/packages/ef/53/900f7d2a54557c6a37886585a91336520e5539e3ae2423ff1102daf43a7/sqlparse-0.3.0-py2.py3-none-any.whl
Installing collected packages: pytz, asgiref, sqlparse, Django
WARNING: The script sqlformat.exe is installed in 'C:\Users\maari\AppData\Local\Packages\PythonSoftwareFoundation.Python.3.8_qbz5n2kfra8p0\LocalCache\local-packages\Python38\Scripts' which is not on PATH.
Consider adding this directory to PATH or, if you prefer to suppress this warning, use --no-warn-script-location
WARNING: The script django-admin.exe is installed in 'C:\Users\maari\AppData\Local\Packages\PythonSoftwareFoundation.Python.3.8_qbz5n2kfra8p0\LocalCache\local-packages\Python38\Scripts' which is not on PATH.
Consider adding this directory to PATH or, if you prefer to suppress this warning, use --no-warn-script-location
Successfully installed Django-3.0.2 asgiref-3.2.3 pytz-2019.3 sqlparse-0.3.0
```

Kuva 3. Asennuksen aikaiset varoitukset.

Testasin asennuksen toimivuutta yrittämällä luoda uuden projektin:

```
django-admin startproject testasta
```

mutta tuloksena vain herja: `django-adminia` ei löydy. Menin siis järjestelmän ympäristömuuttujiin ja lisäsin uuden polun Path -muuttujiin:

```
C:\Users\maari\AppData\Local\Packages\PythonSoftwareFoundation.Python.3.8_qbz5n2kfra8p0\LocalCache\local-packages\Python38\Scripts.
```

Uudestaan projektin luonti ja nyt projektirakenne syntyi. Avasin projektinkansion Codeen. Käynnistin kehityspalvelimen: `python manage.py runserver` ja avasin verkkosivun: <http://localhost:8000/> Näkyviin tuli lopultakin raketti! (lähteenä: <https://docs.djangoproject.com/en/3.0/intro/tutorial01/>)

3 KÄYTTÖÖNOTTO: HELLO WORLD!

Djangon toimivan asennuksen ja testausta -projektin luonnin jälkeen tein projektiin appin `helloWorld`: `python manage.py startapp helloWorld` ja avasin sen kansiorakenteen Codeen. Kirjoitin ensimmäisen näkymäfunktion appin `views.py` -tiedostoon:

```
from django.http import HttpResponse
```

```
def index(request):
    return HttpResponse("<h1> TERVE! <h1>")
```

Loin appin reittejä varten appikansioon urls.py -tiedoston, johon kirjoitin ensimmäisen reitin tiedot:

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

urlpatterns = [
    path('', views.index, name='index'),
]
```

Lisäksi appin reitit tuli kytkeä koko projektin reitteihin projektin omassa urls.py -tiedostossa, jonka urlpatterns:iin lisäsin rivin:

```
path('', include('helloWorld.urls'))
```

Halusin, että Hello World -appi käynnistyy suoraan projektin juuresta, joten määrittelin sen juuripoluksi. Kehityspalvelimen käynnistytksen jälkeen verkkosivulla <http://localhost:8000> näkyikin pelkkä teksti: TERVE!

Halusin seuraavaksi reitin lähettävän vastaukseksi kokonaisen html-sivun, en pelkkää yhtä riviä. Kopioin ja muutin index näkymäfunktion nimeksi terve ja lisäsin muutoksen myös appin reitteihin. Nyt pelkkä tervehdys avautuu <http://localhost:8000/terve>.

Indexin tulisi siis pyytää avattavaksi html -sivun templatea. Loin appikansioon templates -kansion, jonka alle helloWorld- kansion, jonne siis tiedoston index.html. Tämän jälkeen indexin näkymäfunktio muotoon:

```
from django.template import loader

def index(request):
    template = loader.get_template('helloWorld\index.html')
    return HttpResponse(template.render(request))
```

Kehityspalvelimen käynnistytksen jälkeen verkkosivulla näkyi vain virheilmoitusta, jossa herjattiin, ettei index.html tiedostoa löytynyt. Tiedostopolku, josta sovellus tiedostoa haki, oli väärä, joten muutin projektin settings.py tiedostoon TEMPLATES määrittelyyn:

```
'DIRS': [os.path.join(BASE_DIR, 'templates')]
```

Tämä muutos tarkoitti myös samalla, että templeatit löytyvät nyt aina projektin juuresta, joten siirsin koko template -kansion projektin juureen. Nyt, jos olisi useita appeja projektissa, pitäisi olla tarkkana templeatien nimeämisen kanssa, koska juuressa ollessa ne ovat kaikille apeille yhteisiä. Korjasin vielä indexin näkymäfunktion:

```
template = loader.get_template('index.html')
```

Muutosten jälkeen homma toimi. (Idean lähde: <https://djangoforbeginners.com/pages-app/> .)

Perusverkkosivu näytti tylsältä. Halusin siis ottaa Bootstrapin mukaan. Se onnistuikin helposti perinteisesti kopioimalla latauslinkki html -koodiin mukaan ja lisäämällä koodiin bootstrapin tagit. Lisäksi tahdoin oman tyyllisivun. Se vaati staattisten tiedostojen mukaan ottamista. Tein static -kansion projektin juureen ja sen alle oman tyyllisivun. Lisäsin html-sivulle tagin {% load static %} ennen tyyllisivun linkitystä. Myös projektin settings.py tiedosto tarvitsi muok-
kausta staattisten tiedostojen polun osalta:

```
STATICFILES_DIRS = [
    os.path.join(BASE_DIR, "static"), #juuresta
]
```

Näin sain oman tyyllisivun käyttöön.

Seuraava ongelma oli siis saada käyttäjän nimi näkyviin. Tein ensin version, jossa käyttäjä antaa nimensä verkkosivua kutsuessaan eli <http://localhost:8000/Maarit> jolloin tekstiksi tulee Terve, Maarit! tai jos pelkkä <http://localhost:8000> niin tekstinä Terve, tuntematon! Ratkaisin sen näin:

```
def index(request):
    template = loader.get_template('index.html') #juuressa
    context = {
        "henkilo": "tuntematon"
    }
    return HttpResponse(template.render(context, request))

def nimella(request, nimi):
    template = loader.get_template('index.html')
    context = {
        "henkilo": nimi
    }
```



```
return HttpResponse(template.render(context, request))
```

ja lisäksi urls.py reititykseen:

```
path('<str:nimi>/', views.nimella, name='nimella'),
```

eli jos palvelin saa pyynnössä käyttäjän nimen se reitittää näkymäfunktiolle nimella muutoin index kautta. Kumpaankin funktioon on lisätty context muuttujien lähettämistä varten index -templeatille.

Ja lopuksi tahdoin kokeilla lomakkeen käyttöä käyttäjän nimen välityksessä.

Tein appikansioon tiedoston forms.py, johon:

```
from django import forms
```

```
class NimiForm(forms.Form):
    nimi = forms.CharField(label='Nimi', max_length=100)
```

Eli tein input kentistä NimiForm - lomakeolion attribuutteja.

Koodasin myös uuden html-sivun lomake.html, jossa lomake olio otetaan käyttöön välitettynä muuttujana:

```
<form class="mt-4" action="{% url 'lomake' %}" method="POST">
    {% csrf_token %}
    <div class="form-group">
        {{ form }}
        <button type="submit" class="btn btn-primary">
            Lähetä</button>
    </div>
</form>
```

Ja vielä itse lomakkeen näkymäfunktio sekä reitti:

```
from .forms import NimiForm

def lomake(request):
    # jos POST pyyntö
    if request.method == 'POST':
        nimi=request.POST['nimi']
        return HttpResponseRedirect('/')+nimi
    # jos GET pyyntö
    else:
        form = NimiForm()
    return render(request, 'lomake.html', {'form': form})
```

```
path('lomake', views.lomake, name='lomake'),
```

Tämä oli tosi kätevä tapa, kun yksi funktio tarkisti suoraan, onko kyseessä POST vai GET pyyntö.

Nyt enää piti saada index -html:ssä tuntemattomalle käyttäjälle näkyviin nappi: "Anna nimesi" ja jos nimi on annettu, niin nappia ei näytetä. Se toimi näin:

```
{% if tyhja %}
<a class="btn btn-primary" href="lomake" role="button">Kerro
nimesi</a>
{% endif %}
```

jonka vuoksi index- ja nimellä -näköyfunktioiden contexttiin lisäsin vielä *tyhja* -muuttujan.

Koodit liitteenä!

Käytetyt lähteet:

Project&App: <https://docs.djangoproject.com/en/3.0/intro/tutorial01/>

Views: <https://docs.djangoproject.com/en/3.0/intro/tutorial03/>

<https://djangoforbeginners.com/hello-world/>

Urls: <https://docs.djangoproject.com/en/3.0/intro/overview/#design-your-urls>

<https://django.readthedocs.io/en/2.2.x/topics/http/urls.html>

Templates: <https://docs.djangoproject.com/en/3.0/ref/templates/api/>

<https://djangoforbeginners.com/pages-app/>

Forms: <https://docs.djangoproject.com/en/3.0/topics/forms/>

<https://docs.djangoproject.com/en/3.0/ref/forms/fields/>

LÄHTEET

Django Software Foundation. 2005-2020. How to get Django. Saatavilla: <https://www.djangoproject.com/download/#supported-versions> Viitattu 12.1.2020

Nigel, G. 2020a. Beginning Django Tutorial – Lesson 1. Saatavilla: <https://djangobook.com/beginning-django-tutorial-lesson-1/> Viitattu 12.1.2020

Nigel, G. 2020b. Beginning Django Tutorial – Lesson 2. Saatavilla: <https://djangobook.com/beginning-django-tutorial-lesson-2/> Viitattu 12.1.2020

Roine, J. 2010. Django. Pdf -esitys. Saatavilla: https://www.linux.fi/w/images/6/67/Jani_roine-django.pdf Viitattu:12.1.2020

Asennuksen ja käyttöönoton lähteet on suoraan viitattu tekstissä linkeillä niiden käytön yhteydessä.

Hello World

views.py:

```
from django.shortcuts import render
from django.http import HttpResponse
from django.http import HttpResponseRedirect
from django.template import loader
from .forms import NimiForm

#erilaisia tervehdys kokeiluja

def index(request):
    template = loader.get_template('index.html') #juuressa
    context = {
        "henkilo": "tuntematon",
        "tyhja": True
    }
    return HttpResponse(template.render(context, request))

def nimella(request, nimi):
    template = loader.get_template('index.html')
    context = {
        "henkilo": nimi,
        "tyhja": False
    }
    return HttpResponse(template.render(context, request))

def lomake(request):
    # jos POST pyyntö
    if request.method == 'POST':
        nimi = request.POST['nimi']
        return HttpResponseRedirect('/') + nimi
    # jos GET pyyntö
    else:
        form = NimiForm()
        return render(request, 'lomake.html', {'form': form})

def terve(request):
    return HttpResponse("<h1> TERVE </h1>")

def numero(request, id):
    return HttpResponse("Terve taas, onnennumerosi on: %s" % id)
```

urls.py

```

from django.urls import path

from . import views

urlpatterns = [
    path('', views.index, name='index'),
    path('terve', views.terve, name='terve'),
    path('lomake', views.lomake, name='lomake'),
    path('<int:id>/', views.numero, name='numero'),
    path('<str:nimi>/', views.nimella, name='nimella'),
]

```

forms.py

```

from django import forms

class NimiForm(forms.Form):
    nimi = forms.CharField(label='Nimi', max_length=100)

```

lomake.html

```

<!DOCTYPE html>
<html lang="fi">
  <head>
    <meta charset="utf-8" />
    <link rel="stylesheet" href="https://stack-
path.bootstrapcdn.com/bootstrap/4.4.1/css/bootstrap.min.css" in-
tegrity="sha384-Vkoo8x4CGs03+HhXv8T/Q5PaXtkKtu6ug5TOeNV6gBiFeW-
PGFN9MuhOf23Q9Ifjh" crossorigin="anonymous">
    {% load static %}
    <link rel="stylesheet" type="text/css" href="{% sta-
tic 'style.css' %}">
    <title>Hello -lomake!!</title>
  </head>
  <body>
    <div class="container">

      <form class="mt-4" action="{% url 'lomake' %}" method="POST">
        {% csrf_token %}
        <div class="form-group">
          {{ form }}
          <button type="submit" class="btn btn-pri-
mary">Lähetä</button>
        </div>
      </form>

    </div>
  </body>
</html>

```

index.html

```

<!DOCTYPE html>
<html lang="fi">
  <head>
    <meta charset="utf-8" />
    <link rel="stylesheet" href="https://stack-
path.bootstrapcdn.com/bootstrap/4.4.1/css/bootstrap.min.css" in-
tegrity="sha384-Vkoo8x4CGs03+HhXv8T/Q5PaXtkKtu6ug5TOeNV6gBiFeW-
PGFN9MuhOf23Q9Ifjh" crossorigin="anonymous">
    {% load static %}
    <link rel="stylesheet" type="text/css" href="{% sta-
tic 'style.css' %}">
    <title>HelloWorld!!</title>
  </head>
  <body>
    <div class="container">
      <h1>Terve {{henkilo}} !!</h1>
      {% if tyhja %}<a class="btn btn-primary" href="lomake" role="but-
ton">Kerro nimesi</a>{% endif %}
    </div>
  </body>
</html>

```

testausta/urls.py

```

from django.contrib import admin
from django.urls import include, path

urlpatterns = [
    path('', include('helloWorld.urls')),
    path('admin/', admin.site.urls),
]

```