



# TDP003 Projekt: Egna datormiljön

## Egen utvecklingsmiljö

Kursmaterial till kursen TDP003



Höstterminen 2014  
Version 2.1

# Innehållsförteckning

1	Revisionshistorik.....	3
2	Introduktion.....	3
3	Installation av verktyg för portfolio-projekt.....	3
4	Flask.....	4
4.1	Installera Flask.....	4
4.2	Testa Flask.....	5
5	(X)HTML-mallar för Python.....	6
6	Teckenkodning.....	6
6.1	Tips om Unicode.....	6
7	Webbverktyg.....	6
7.1	Firebug.....	6
7.2	Selenium.....	7

## 1 Revisionshistorik

Ver.	Revisionsbeskrivning	Datum
1.1	Översyn av språkmissar. Borttagning av en inaktuell bild i kapitel 2. Borttagning av inaktuellt stycke på slutet om enhetstestning.	130826
1.2	Ersatt kapitel om Apache och mod_python med Flask.	130826
1.3	Uppdaterade instruktioner för Python3 samt installation av Flask. Skrev lite kort om Selenium.	140827
2.0	Uppdaterat layout. Infogat tabell under Revisionshistorik. Information uppdaterad till 2014.	140903
2.1	Datum och sidnummer i sidhuvud. Tagit bort referenser till Ubuntu. Refererar till HTML5 och CSS3. Rättstavning.	140908

## 2 Introduktion

Detta dokument innehåller information om de verktyg som du kommer använda för Portfolioprojektet. Dessa verktyg är utvalda, inte bara för detta projekt utan även för att det är viktiga verktyg som du bör lära dig och som du sedan ska kunna använda fortlöpande under utbildningen. Samtidigt undviker vi att föregå utbildningen och använda verktyg som förutsätter kunskaper som nya IP-studenter inte förutsätts att ha. Avancerade ramverk, så som Django, undviks av denna anledning.

Innan själva utvecklingsarbetet kan komma igång i ett projekt måste den egna datormiljön tas ett steg vidare till att bli en **utvecklingsmiljö**. Detta innebär att er datormiljö måste ställas i ordning för att stödja det tekniska arbetet. Främst gäller här att installera och lära sig de verktyg som ska användas. Det brukar även vara bra att börja lära sig mer om den teknik som ska användas i det färdiga systemet, i vårt fall portfoliosystemet.

I denna kurs stödjer vi sammanställandet av denna miljö i detalj, men i senare projektkurser förutsätts du normalt göra denna typ av förberedelser självständigt.

## 3 Installation av verktyg för portfolio-projekt

Din arbetsmiljö ska utgöras av följande:

- Linux-version
- Python3.3+ (finns ofta förinstallerat)
- Subversion
- Emacs med python-mode (och stöd för Git)

- Grafikverktyg t.ex. GIMP

Sök reda på de paket ni saknar och installera dem med lämpligt metod (t.ex. med apt-get).

I kursen använder vi **Git**. Det är baserat på öppen källkod, gratis och har ett brett stöd för olika operativsystem. Gällande Git behöver ni bara installera klientpaketet. Gitservern ni ska använda körs centralt på IDA. Läs mer om installation av Git under kursen *TDP001* (<https://www.ida.liu.se/~TDP001/>).

**Verifiera att installationen gått bra** genom att starta och bekanta er med de nya programmen.

**Vid projektstart** rekommenderas att ni installerar och bekantar er med följande tekniker. Mer detaljkunskap lär ni er när aktuellt moment nås.

Tekniker som kommer att användas:

- HTML5
- CSS3
- Flask webbramverk
- Jinja2 Mallspråk för webbsidor

## 4 Flask

I kursen kommer ni utveckla webb-delen till ert portfoliosystem med ett ramverk som heter Flask. Flask är ett så kallat mikro-ramverk för python, byggt för att underlätta utvecklingen av mindre webbplatser. I kursen kommer vi även använda oss av Flasks inbyggda webbserver för att underlätta vår utveckling av projektet. Dock går det utmärkt att köra ett system utvecklat med Flask på andra webbserverar, så som Apache.

### 4.1 Installera Flask

Det finns ett flertal olika sätt att installera Flask, det som presenteras är ett av dem som fungerar enkelt i Linux. I en terminal ge följande kommandon:

```
(>> sudo apt-get update) #alltid bra att göra
(>> sudo apt-get install python-setuptools) #kan vara förinstallerad
#hämta get-pip.py från
https://packaging.python.org/en/latest/tutorial.html#installing-the-tools
>> sudo python3 get-pip.py #Om ej installerad per default
>> sudo pip3.4 install flask #jinja2 installeras också
```

## 4.2 Testa Flask

Skapa en fil som heter `test_flask.py` och öppna den i emacs. Skriv/kopiera in följande pythonkod:

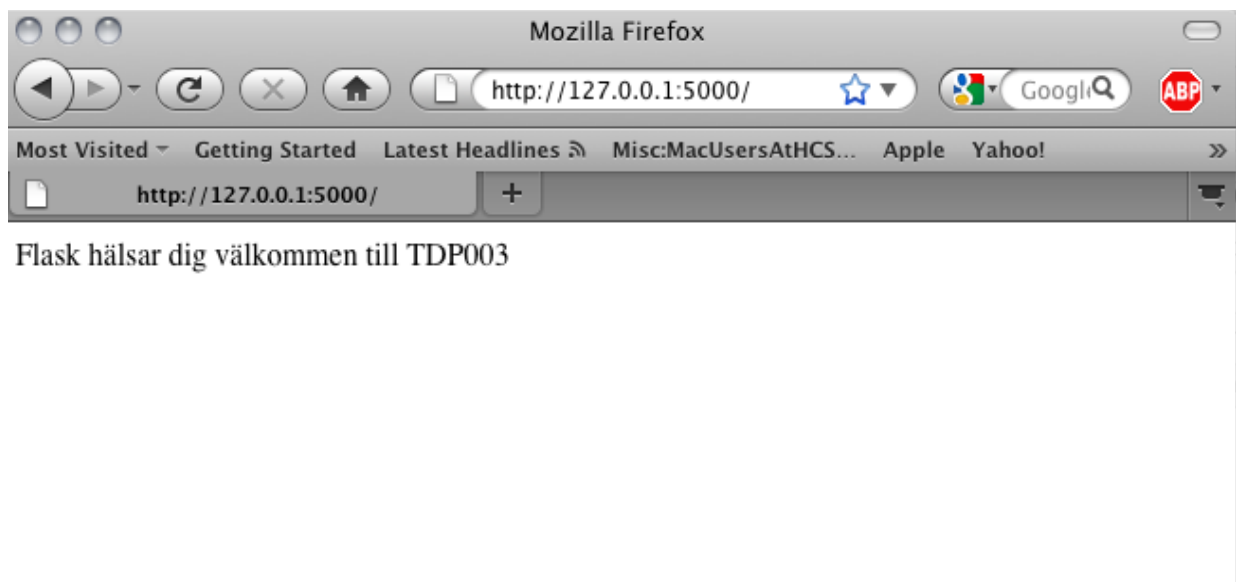
```
# -*- coding: utf-8 -*-
from flask import Flask
app = Flask(__name__)
@app.route("/")
def hello():
    return "Flask hälsar dig välkommen till TDP003"

if __name__ == "__main__":
    app.run()
```

Kör sedan (i samma katalog som filen):

```
python3 test_flask.py
```

Öppna url:en <http://127.0.0.1:5000> i valfri webbläsare. Du bör då se något i stil med :



Ni har nu startat Flask's inbyggda webbserver och skrivit er första minimala webbapplikation. Ni behöver, just nu, inte oroa er för att ni inte förstår kod-snutten eller vad som händer i bakgrunden.

## 5 (X)HTML-mallar för Python

Det som återstår är ett hjälpbibliotek för Python som förenklar skapandet av webbsidor med dynamiskt innehåll. Det finns ett flertal motorer för mallspråk till python, vi rekommenderar att ni använder Jinja2, vilket även installeras tillsammans med Flask.

## 6 Teckenkodning

### OBS viktigt!

När man lagrar text i filer (eller databaser) eller överför den i form av en webbsida måste man hålla reda på teckenkodningen. Ursprungligen använde man bara engelska bokstäver och en del andra tecken (ASCII) men det duger ju inte idag! För att kunna använda alla olika tecken som finns kan de flesta programmeringsspråk idag hantera Unicode. Man brukar dock inte använda Unicode för överföring eller lagring av text då det skulle kräva onödigt mycket bandbredd respektive utrymme.

I portfolioprojektet kommer vi att använda teckenkodningen UTF-8 för såväl datalagringen som utdata som skickas till webbläsaren. Det finns många saker som talar för detta val:

- utrymmessnålt
- stöds av Python
- standard för HTML5-dokument
- standard för utdata i mallar

### 6.1 Tips om Unicode

I den bästa av världar fungerar det här idag av sig självt, men det är ändå viktigt att se till att hela kedjan från indata till utdata sker i samma teckenkodning. **Se till att ha till exempel svenska tecken i testdata så att eventuella problem kan upptäckas!**

I Python3+ används Unicode som standard. Men Unicode är inget teckenspråk i sig, utan är mer av en container. Unicode kan sedan implementeras med olika typer av teckenkodningar, där ni ska se till att den utgörs av UTF-8. Ni kan läsa mer om Unicode på bland annat <http://www.unicode.org>.

Det finns fortfarande moduler i Python som inte hanterar Unicode av sig själva. Ni får ta reda på det här och hitta vägar runt eventuella problem.

### 7 Webbverktyg

#### 7.1 Firebug

Firebug är ett tillägg till Firefox som gör att man kan undersöka en webbsida på alla möjliga sätt. Det ger mycket god hjälp att komma till rätta med problem i JavaScript-kod, CSS-layouer och mycket mer. Det finns även numer bra debuggingmöjligheter inbyggda i Firefox, dessa kan man komma åt genom att trycka Ctrl-Shift-i.

- <http://getfirebug.com>

#### 7.2 Selenium

Selenium är ett program för att skapa automatiserade manuella tester. Ni kommer att använda

Selenium senare i andra kurser, men har redan nu möjlighet att bekanta er med verktyget.

- <http://www.seleniumhq.org/>