

# DD1310 Programmering för OPEN1

## LAB1: Laborationer i Python läsåret 2012/2013

Räkna med att varje laboration kan ta upp till 10 timmar att göra. Du behöver alltså tillbringa en hel del tid i datorsalarna. Om du behöver hjälp så finns allmänhandledningen på plats varje vardag kl 11-13 och 17-20, se <http://www.nada.kth.se/handledning/handledare4/general.php>

Fyll i ditt namn och personnummer nedan med bläck eller motsvarande. Denna kvittenssida ska du spara som kvitto, för att visa upp om du senare upptäcker att dina labbar inte har blivit inrapporterade.

Kursledare är Sten Andersson, [stene@csc.kth.se](mailto:stene@csc.kth.se).

Namn ..... Personnr .....

### Laborationer i Python

Laboration 1	Godkänt den ..... (bonusdatum 2012-08-30)	Kvitteras .....	Timmar .....
Laboration 2	Godkänt den ..... (bonusdatum 2012-09-06)	Kvitteras .....	Timmar .....
Laboration 3	Godkänt den ..... (bonusdatum 2012-09-13)	Kvitteras .....	Timmar .....
Laboration 4	Godkänt den ..... (bonusdatum 2012-09-20)	Kvitteras .....	Timmar .....
Laboration 5	Godkänt den ..... (bonusdatum 2012-09-26)	Kvitteras .....	Timmar .....

### P-del

Spec	Godkänt den ..... (senast 2012-10-29)	Kvitteras .....	
Redovisning	Godkänt den ..... (senast 2012-12-XX)	Kvitteras .....	Timmar .....
Granskning	Godkänt den ..... (senast 2012-12-XX)	Kvitteras .....	

# Laboration 1: Introduktion till datormiljön och Python

**Nyckelord:** *katalog, fil, program, variabel, tilldelning, inläsning, utskrift, funktion, anrop.*

**Mål:** *Att du efter laborationen ska kunna använda Linux, och Emacs samt skriva och köra enkla Python-program.*

**Föreberedelse:** *Deltagit i föreläsningar och övningar.*

## Sammanfattning av labben

I den här labben ska du lära dig om operativsystemet **Linux** och redigeringsprogrammet **Emacs**. Du ska också registrera dig på kursen, och sist men inte minst skriva ett enkelt Pythonprogram.

## Inloggning

Sätt dig vid en dator tillsammans med din labbkompis och logga sedan in. Längst upp till vänster finns en "Ubuntu" ikon som ungefär motsvarar startknappen i Windows. Via menyn som denna knapp aktiverar startar man många program. Klicka på den och välj "Terminal" för att öppna ett terminalfönster. Terminalfönstret motsvarar "Command Prompt" i Windows. När du skriver i det fönstret ger du kommandon till kommandotolken. Kom ihåg att kommandotolken skiljer mellan stora och små bokstäver.

## Filer och kataloger

Med en filhanterare kan du utforska datorns och nätverkets *kataloger*. En katalogs innehåll kan bestå av andra kataloger och *filer*. Kataloger som ligger i en annan katalog kallas ibland *underkataloger*.

Kataloger med underkataloger bildar en grenliknande struktur där varje gren är en katalog. Denna grenstruktur kallar man ofta *katalogträd*. Till skillnad från verkliga träd brukar man dock låta katalogträd breda ut sig nedåt och till höger istället för rakt upp. En bieffekt av detta är att katalogträdet *rot* hamnar längst upp till vänster samt att man pratar om att "gå ned" i en katalog (eller underkatalog).

En fullständig beskrivning av den plats en fil ligger på brukar kallas *sökväg* och kan till exempel se ut på följande sätt:

```
/info/prgo12/studiehjalp.py
```

```
/home/a/labbkompisens_kontonamn/studiehjalp.py
```

Notera att katalognamn och filnamn separeras av tecknet '/' och inte av '\' som i Windows. Ett annat speciellt tecken är '~' (uttalas "tilde") som är en förkortning av den inloggades hemkatalog vilken i verkligheten ligger någonstans under katalogen /home i katalogträdet (/home/...). Hemkatalogen är den katalog du hamnar i när du loggar in och i vilken du har rätt att spara filer och kataloger.

Starta filhanteraren genom att i menyn "Places" uppe till vänster välja "Home Folder"

## Tillkalla handledare med Sima-manager

Om du vill fråga något eller redovisa använder du *sima manager* som är ett program du kör på din dator. Skriv följande i terminalfönstret:

```
module add sima
```

För att starta programmet ska du sedan skriva `sm` och vänta en stund. För att tillkalla handledare leta fram kursen `prgo` från listan: tryck på knappen "login" och slutligen tryck på knappen "Queue" för att få hjälp. Vänta några minuter så kommer snart en handledare till dig.

Om du vill slippa att skriva "module add sima" i framtiden kan du lägga till "module add sima" i filen `.modules` i din hemkatalog.

Prova och ställ dig i kön för att få hjälp med detta, alltså om du behöver hjälp. Välj `prgo12` när du har schemalagd labb och `allmanhand` övriga tider så får du hjälp av rätt handledare.

## Användbara kommandon

Terminalfönstret har en mängd kommandon och nedan finns exempel på de mest användbara för denna kurs.

Funktion	Kommandoexempel
Lista innehåll i aktuell katalog	<code>ls</code>
Lista innehåll i aktuell katalog, visa även gömda filer	<code>ls -a</code>
Byt aktuell katalog till hemkatalogen <code>~</code>	<code>cd</code>
Byt aktuell katalog till underkatalogen <code>lab1</code>	<code>cd lab1</code>
Byt aktuell katalog till "katalogen ovanför"	<code>cd ..</code>
Skapa en katalog som heter <code>lab1</code>	<code>mkdir lab1</code>
Kopiera filen <code>exempel</code> till din <code>lab1</code> -katalog	<code>cp /info/prgo12/exempel ~/prgo12/lab1/</code>
Kopiera filen <code>.emacs</code> till hemkatalogen	<code>cp /info/prgo12/.emacs ~/</code>
Ta bort filen <code>studiehjalp.py</code> <b>Obs!</b> Filen försvinner för alltid	<code>rm studiehjalp.py</code>
Ta bort katalogen <code>lab2</code> och alla underkataloger <b>Obs!</b> Innehållet försvinner för alltid	<code>rm -r lab2</code>
Exekvera Pythonprogram	<code>python3 studiehjalp.py</code>
Skriv ut filen <code>studiehjalp.py</code> till skrivaren	<code>print studiehjalp.py</code>

## Terminalfönstret

Vissa kommandon skriver man i terminalfönstret (allt ovanstående går också att göra med skrivna kommandon). Till exempel kan du lista alla filer i en katalog med kommandot `ls` (list). Vill du lista alla underkataloger och filer i kurskatalogen skriver du `ls /info/prgo12`.

Det du framför allt behöver veta är hur du flyttar dig mellan olika kataloger i terminalfönstret. För att gå ner i katalogen `prgo12` skriver du `cd prgo12`, vilket utläses *change directory to prgo12*. För att gå upp en nivå skriver du `cd ..` (glöm inte mellanslaget före punkterna).

Det finns genvägar för att slippa skriva så mycket i terminalfönstret; till exempel behöver du sällan skriva ut hela filnamnet. Tryck på TAB eller (längst till vänster) så fylls kommandon, filnamn och katalognamn i om de är entydiga (detta kallas "TAB completion" på engelska). Vill du få upp föregående kommando räcker det att trycka på uppåt-piltangenten som sitter nere till höger på tangentbordet. Trycker du flera gånger bläddrar du successivt tillbaka bland gamla kommandon. Vill du veta mer om kommandot `kommando` kan du ta fram en manual genom att skriva `man kommando`; till exempel kan du skriva `man more` för att få reda på mer om kommandot `more`. Tryck ner tangenten `q` på tangentbordet om du vill avsluta `man`.

## Ni som arbetar i grupp

Ni som arbetar i grupp behöver skapa en katalog som båda kommer åt och kan använda och även registrera den av er som inte är inloggad. För att åstadkomma detta använder ni följande kommandon, där `labbkompis1` är användarnamnet som först loggade in och `labbkompis2` är användarnamnet på den andra personen i gruppen.

```
mkdir prgo12
course include prgo12 labbkompis2
```

Det första kommandot skapar katalogen `prgo12` hos `labbkompis1` och det andra kommandot ändrar rättigheter på katalogen så att även `labbkompis2` kan arbeta i katalogen. Testa nu med kommandot `ls` eller titta i filhanteraren för att säkerställa att katalogen är skapad. Alla kataloger som någon av er skapar under `prgo12` är gemensamma för labbgruppen.

Logga sedan ut `labbkompis1` genom att trycka på "Kugghjulet" längst upp till höger. Logga sedan in `labbkompis2` och skriv sedan

```
course labdir prgo12 labbkompis1
```

Testa med `ls` eller filhanteraren för att säkerställa dig om att katalogen `prgo12` är åtkomlig. Kommandot ovan skapar inte en ny katalog utan en så kallad länk till katalogen `prgo12` hos `labbkompis1`, så att ni i praktiken får en gemensam katalog.

## Webbläsare och kursinformation

Programmet *Firefox* är en webbläsare du startar genom att i "Ubuntu"-menyn välja "Firefox Web Browser". Du kan även starta det genom att skriva `firefox&` i ett terminalfönster. Klicka en gång i det fönster du nu får upp och skriv in följande URL-adress i adressfältet:

```
http://www.csc.kth.se/DD1310/
```

Klicka på länken "prgo12". Du ska nu få upp en sida med kursinformation för programmeringsteknikkursen för OPEN1. Denna sida är viktig! Här kommer vi att lägga ut information under kursens gång, så vänj dig att alltid titta på sidan när du loggar in.

För att slippa klicka dig fram till sidan varje gång ska du spara adressen till sidan med ett bokmärke. Du gör det genom att välja menykommandot `Bookmarks` → `Bookmark This Page`.

## Redigeringsprogrammet Emacs

För att skapa och ändra filer används ett redigeringsprogram. Emacs är ett kraftfullt redigeringsprogram som har flertalet finesser om man t.ex. skall skriva *Pythonkällkod*. En av de viktigaste är *indentering*. Med indentering menas att texten i Pythonkällkoden skjuts in en bit beroende på vilken del av programmet som texten utgör.

Innan du fortsätter med Emacs ska du skapa en ny underkatalog `lab1` i katalogen `prgo12`. Kontrollera att katalogen `lab1` finns. Skapa sedan ytterligare en katalog i `prgo12` för varje laboration (`lab2`, ..., `lab5` och `puppgift`) så blir det enklare för dig att hålla reda på filerna under kursens gång. Kopiera från hemsidan filen `studiehjalp.py` till din katalog `lab1`.

## Filer och buffertar

Två viktiga begrepp i Emacs är *fil* och *buffert* ("file", "buffer"). En fil är något som finns sparad på en hårddisk. Om datorn slås av kommer en fil finnas kvar och kan utnyttjas när datorn startas igen. En buffert är något som används tillfälligt under tiden du skriver. Ändringarna i en buffert finns inte automatiskt kvar om datorn slås av. **Det är alltså viktigt** att spara ändringar i en buffert till en fil lite då och då. Detta gäller speciellt om filen ska utnyttjas till något annat (t.ex. exekvering).

## Öppna en existerande eller en ny fil

Starta Emacs genom att skriva `emacs&` i ett terminalfönster.

Öppna filen `studiehjalp.py` som du kopierat tidigare. Man kan öppna en fil via menyn (`Files` → `Open File`) eller genom tangentkombinationer (`C-x C-f`, `dvs` håll ned kontrolltangente, tryck på `x`, håll ned kontrolltangente, tryck på `f`). Längst ned i Emacs dyker då en rad med den aktuella sökvägen upp. Ersätt den aktuella sökvägen med `~/prgo12/lab1/studiehjalp.py`. Notera hur Emacs använder tecknet `/` för att separera katalognamn och filnamn, precis som i terminalfönstret. **Tips!** "TAB completion" kan användas.

När hela sökvägen är inskriven trycker du på returtangente för att bekräfta valet. Innehållet i filen `studiehjalp.py` ska nu dyka upp i en av Emacs buffertar och börja enligt:

```
# -*- coding: Latin-1 -*-
# Program för studiehjäl
...
```

Observera att om den valda filen inte existerar, så kommer Emacs öppna en ny fil med det valda namnet. Med andra ord används med fördel `Files` → `Open File` eller `C-x C-f` både för att öppna en befintlig fil och för att skapa en ny!

## Redigering och användbara kommandon

Så fort du redigerar lite i en buffert i Emacs så kommer buffertens innehåll inte stämma överens med den sparade filens innehåll. Detta visas i Emacs genom att markeringen `**` visas till vänster om filnamnet längst ned. Sparas bufferten i en fil så försvinner markeringen.

Börja med att spara den aktuella bufferten i en fil med namnet `boksidor.py`. Att spara till en fil med annat namn görs med `Files` → `Save Buffer As` eller `C-x C-w`. **Var noggrann** med `STORA` och små bokstäver!

Ändra någonting i bufferten. Notera hur markeringen `**` dyker upp. Spara bufferten (`Files` → `Save Buffer` eller `C-x C-s`) och notera hur markeringen `**` försvinner.

Du har nu använt några av de vanligaste kommandona i Emacs. Det finns otroligt många fler, men för denna kurs kan nedanstående kommandon vara bra att komma ihåg. Med `C-` avses kontrolltangente nedtryckt, med `M-` avses "meta"-tangente nedtryckt. På en Linux-dator fungerar både "diamant"-tangente (höger om Kontrolltangente) och `Esc`-tangente (uppe till vänster) som metatangent. Trycker man `M-x` kan man skriva in namnet på ett emacs-kommando och köra det. Den som vill lära sig mer om emacs kan läsa Emacs tutorial som nås genom att trycka `F1 F1 t`.

Funktion	Menyval	Kommando
Avbryt		C-g
Öppna befintlig fil/ skapa ny fil	Files → Open File...	C-x C-f
Spara buffert i fil	Files → Save Buffer	C-x C-s
Spara i fil under annat namn	Files → Save Buffer As...	C-x C-w
Stäng buffert	Files → Kill Current Buffer	C-x k
Ångra	Edit → Undo	C-_
Klipp ut från markören till slutet av raden (kan upprepas)		C-k
Start av markering		C-mellanslag
Slut av markering (kopiera)	Edit → Copy	M-w
Slut av markering (klipp ut)	Edit → Cut	C-w
Klistra in	Edit → Paste	C-y
Skriv ut aktuell buffert	Tools → Print → Print Buffer	
Sök	Search → Search...	C-s
Sök och byt	Search → Query Replace...	M-%

## Skrivaren

Skrivarna på CSC heter oftast samma som salen de är placerade i. Kommandot som skriver ut är `print`. Om du vill skriva ut filen `studiehjalp.py` som finns under `~/prgo12/lab1/` skriver du `print ~/prgo12/lab1/studiehjalp.py` i terminalfönstret *Spara papper! Skriv inte ut i onödan!*

## Byt lösenord!

Det lösenord du fick med kontot är säkert svårt att komma ihåg, dessutom kan någon ha sett det. Du ska därför byta lösenord. Byte av lösenord gör du i terminalfönstret med kommandot `passwd`. Skriv `passwd` och tryck returtangenten. Det kommer upp instruktioner på skärmen som berättar vad du ska göra.

Användardatabaserna på CSC och KTH centralt är tyvärr inte synkroniserade, mer än att man initialt har samma lösenord i de båda databaserna. Ett lösenordsbyte på ett av dina konton kommer inte att påverka lösenordet på ditt andra konto.

## Programmeringsövningar

**Uppgift:** Besök kurshemsidan under länkar, övriga filer och ladda ner det programskal som finns där. Komplettera detta program så att det fungerar.

Om du vill kunna använda å, ä och ö i kommentarer och print-satser skriver du raden

```
# -*- coding: Latin-1 -*-
```

allra överst i programmet (så vet Python att du använder svenska bokstäver).

Krav för programmet i labb 1:

Ditt program ska visa att du kan hantera variabler, tilldelning, inläsning, utskrift, funktionsdefinitioner och funktionsanrop.

Glöm inte att skriva en kommentar med ditt och labbkompisens namn längst upp i programmet!



## Kontroll inför redovisning

- Jag vet hur jag hittar information på kurshemsidan.
- Jag vet hur man skapar, öppnar och sparar filer.
- Jag kan använda Emacs.
- Jag kan skriva program i Python som skriver ut text.
- Jag kan vet hur ett tillåtet variabelnamn i Python ska se ut.
- Jag kan skriva program som läser in data från tangenbordet.
- Jag kan definiera och anropa funktioner.
- Jag vet vad som händer när ett program som körs kommer till ett funktionsanrop.
- Jag vet vad som händer när ett program som körs kommer till slutet av en funktion.

## Redovisning

Använd `sima` för kursen `prgo12` att redovisa din laboration. Du kan kontrollera att dina resultat har blivit rapporterade på "<https://rapp.csc.kth.se/rapp/>"  
Tänk på att laborationerna tar betydligt längre tid än den schemalagda datorsalstiden, så förbered dig och påbörja laborationen långt innan det schemalagda laborationstillfället! Du har tillgång till CSCs terminaler dygnet runt, men salarna kan vara bokade vissa tider. Om du behöver hjälp, se länken "Hjälp" på kurshemsidan.