

Oppgaver forkurs i informatikk - dag 2

Lage ny mappe for dag 2

Dersom du allerede har laget en forkurs-mappe i går, kan du navigere til denne ved å bruke `cd` kommandoen, altså `cd` og mappenavnet. Hvis mappen din heter *forkurs* skriver du `cd forkurs` i terminalen.

Er du usikker på om du har opprettet en mappe eller har glemt hva den heter kan du se hvilke mapper og filer som ligger i gjeldende mappe med `ls` kommandoen. Skriv `ls` i terminalen og se om mappen din er blant det som kommer opp.

Dersom du ikke har opprettet en forkurs-mappe kan du gjøre det nå. Du oppretter en mappe ved å bruke kommandoen `mkdir` og deretter navnet du ønsker at mappen skal ha. Dersom mappen skal hete *forkurs* skriver du `mkdir forkurs` i terminalen. For å se at mappen har blitt opprettet bruk `ls` kommandoen og se at mappenavnet kommer opp. Naviger til denne mappen med `cd` og mappenavnet, altså skriv `cd forkurs` i terminalen.

Lag nå en mappe for dag2 på tilsvarende måte som over, med `mkdir dag2`. Se at mappen er opprettet med `ls`, og naviger til den med `cd dag2`.

Sjekk at du er i mappen dag2 ved å bruke `pwd` kommandoen, altså skriv inn `pwd` i terminalen. Du burde få opp noe på formen `/uio/hume/student-u38/brukernavn/forkurs/dag2` hvor det i stedet for brukernavn står ditt brukernavn.

Åpne Atom

Du kan åpne Atom (teksteditoren vi skal bruke) ved å skrive `atom . &` i terminalen. Punktumet gjør at Atom åpnes i den mappen vi er, mens `&`-tegnet gjør at kommandoen kjøres i bakgrunnen, og sikrer at vi kan fortsette å bruke og skrive i terminalen mens Atom er åpen.

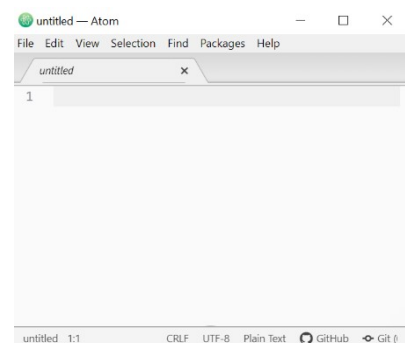
Det er også mulig å bare skrive `atom &`, eller `atom`. I noen tilfeller vil det ikke være mulig å skrive i terminalen dersom du har ikke har med `&`-tegnet, men da kan du åpne en ny terminal under *Applications*-menyen øverst i venstre hjørnet, under *Favorites* og *Terminal*.

Det er også mulig å åpne Atom fra *Application*-menyen øverst i venstre hjørnet under *Accessories* og *Atom*, eller *Programming* og *Atom*.

Lag en ny fil

Når du først åpner Atom vil du mest sannsynlig få opp flere faner med informasjon. Disse kan du lukke. Åpne en ny fil ved å gå under *File*-menyen i øverste venstre hjørnet og velg *New File*. Du burde da få opp en tom fane med navn *untitled* hvor det er mulig å skrive inn tekst. Skriv inn følgende tekst:

```
print("Dette er mitt første python program.")
print("Hurra!")
```



Lagre filen ved å velge *Save* under *File*-møyen og lagre filen som *utskrift.py* hvor *.py*-endelsen står for at dette er en fil med python kode. Pass på at du lagrer filen i mappen vi er i (altså i *dag2*-mappen under *forkurs*).

Kjøre python kode

Gå tilbake til terminalen for å kjøre programmet. For å kjøre programmet må vi være i samme mappe som vi har lagret *utskrift.py*. Du kan sjekke hvilken mappe du er i med *pwd* kommandoen. For å se at filen ligger i den gjeldende mappen kan du bruke *ls* kommandoen. Dersom du ikke er i riktig mappe kan du navigere deg frem til den med *cd* kommandoen.

For å kjøre programmet kan du nå skrive *python3* etterfulgt av filnavnet, altså: *python3*

utskrift.py

Hva skrives ut?

Endre koden

Gå tilbake i Atom og legg til din egen *print*-setning etter de to første. Du skriver da *print("din tekst her")* hvor du erstatter *din tekst her*, med det du ønsker at skal skrives ut i terminalen. Lagre filen ved å velge *Save* i *File*-menyen.

Gå tilbake til terminalen. For å rydde terminalen for tekst kan du skrive *clear*.

Kjør nå filen på nytt fra terminalen med *python3 utskrift.py*. Hva skrives ut nå?

Feilmeldinger

Når vi skriver kode er det viktig å skrive riktig, men vi skriver alle feil innimellom. Se hva som skjer dersom du i stedet for *print* et sted skriver *pint* (uten *r*). F.eks.

```
print("dette skrives ut først")
pint("så skrives dette ut")
print("og noe mer")
```

Lagre filen og kjør den på nytt. Hva skrives nå ut?

Programmet over ga følgende feilmelding

```
dette skrives ut først
Traceback (most recent call last):
  File "mittProgram.py", line 2, in <module>
    pint("så skrives dette ut")
NameError: name 'pint' is not defined
```

Denne feilmeldingen sier oss noe om hva som gikk galt, her at det ikke finnes noe som heter *pint*, og hvor det gikk galt, nemlig linje 2.

Rett opp programmet ditt (endre *pint* tilbake til *print*), lagre filen, og kjør det på nytt med *python3 utskrift.py*. Se at det nå gikk fint.

Variabler

Gå til *File*-menyen øverst i venstre hjørnet og velg *New File*. Du skal nå lage en variabel som holder på navnet ditt. Det gjør du ved å skrive variabelnavnet etterfulgt av = og navnet ditt inne i hermetegn. I eksempelet under er variabelnavnet *navn* og navnet er *Ditt navn*. Skriv så ut verdien som ligger i variabelen ved å skrive `print(variabelnavnet)`:

```
navn = "Ditt navn" print(navn)
```

Lagre filen ved å gå under *File*-menyen øverst i venstre hjørnet og velg *Save*. Lagre filen som `mittProgram.py`. Pass på at du lagrer det i den gjeldende mappen. For å sjekke om filen er der kan du skrive `ls` i terminalen.

Kjør så filen ved å skrive `python3 mittProgram.py`. Hva ble skrevet ut?

Kommentarer i koden

Det kan være veldig nyttig å ha kommentarer i egen kode, dvs tekst som kun er der for oss som leser programmet, og som ikke gjør noe når koden kjøres. Man kan lage kommentarer som går over en linje ved å starte med #, eller over flere linjer ved å starte og avslutte med `"""`. Under er et eksempel med kommentarer:

```
# Dette er en kommentar

"""
Denne kommentaren går over flere
linjer.
Det er vanlig å bruke kommentarer til
å beskrive hva koden gjør.
""" print("dette skrives ut først")
print("så skrives dette ut")
print("og noe mer")
```

Legg til kommentarer i ditt eget program og kjør det på nytt med `python3 mittProgram.py`. Se at kommentarene ikke endrer på hva som skrives ut til terminalen.

Prøv deg frem

Prøv deg gjerne frem ved å lage flere variabler eller legg inn flere `print` setninger i programmet ditt. Husk at dersom du har flere variabler må de ha forskjellige navn (se eksempel under)

```
navn = "Ditt
navn" tall = 4
tall2 = 7
print(navn)
```

Lagre filen mellom hver gang du har gjort endringer og kjør den med `python3 mittProgram.py` for å se hvordan endringene du har gjort endrer på hva som blir skrevet ut. Husk at du kan rydde terminalen for tekst ved å skrive `clear`.

Lever fil i Devilry

På IFI leveres obligatoriske innleveringer i Devilry. For å teste ut dette gå inn på

<https://devilry.ifi.uio.no/> og logg med UiO-brukernavn og passord. Lever `mittProgram.py`