

Intro til jupyter

September 11, 2020

```
<div class="navbar-header">
  <a class="navbar-brand" href="_Oving1.ipynb">Øving 1</a>
</div>
<ul class="nav navbar-nav">
<li class="active"><a href="Intro%20til%20jupyter.ipynb">Intro til Jupyter</a></li>
<li><a href="Jeg%20elsker%20ITGK!.ipynb">Jeg elsker ITGK!</a></li>
<li><a href="Kalkulasjoner.ipynb">Kalkulasjoner</a></li>
<li><a href="Input%20og%20variable.ipynb">Input og variable</a></li>
<li><a href="Tallkonvertering.ipynb">Tallkonvertering</a></li>
<li><a href="Peppes%20Pizza.ipynb">Peppes Pizza</a></li>
<li><a href="Geometri.ipynb">Geometri</a></li>
<li><a href="Vitenskapelig%20notasjon.ipynb">Vitenskapelig notasjon</a></li>
<li><a href="Tetraeder.ipynb">Tetraeder</a></li>
<li><a href="Bakekurs.ipynb">Bakekurs</a></li>
<li><a href="James%20Bond%20and%20Operation%20round().ipynb">James Bond and Operation Round</a></li>
</ul>
```

1 Introduksjon til Jupyter

Hei, og velkommen til Jupyter, en annen måte å skrive kode på! Jupyter er et system som lar deg lage dokumenter som inneholder både tekst og kode på en gang. Det fine her er at du kan kjøre koden i dokumentet og se resultatet umiddelbart. Dette kan du prøve ut nå.

oppgave a) Klikk på kodeblokken under og trykk **ctrl + enter** på tastaturet for å kjøre koden. (Det er også mulig å klikke på kodeblokken for så å klikke **run** i menyen på toppen)

```
[ ]: print("Velkommen til jupyter")
```

Som du ser kommer resultatet av koden ut rett nedenfor kodeblokken. Dette er veldig praktisk og du kan kjøre koden så mange ganger du vil! Hvis du trykker på **ctrl + enter** i kodeblokken over en gang til vil du se at tallet til venstre for kodeblokken øker. Dette tallet brukes bare som referanse og er ikke noe du trenger å tenke på til vanlig.

Alle kodeblokker i et dokument kan endres på, og dette oppfordres på det sterkeste! Det er mye god læring i å endre kode, tenke seg til hva som skal skje og sjekke om dette faktisk skjer. Du kan for eksempel prøve å kjøre programmet under med **ctrl + enter**, gjøre et par endringer og sjekke om den nye versjonen din gjør det du hadde tenkt.

Oppgave b) Endre første linje i koden under til `print("Dette er mitt første Jupyter-program")`

```
[ ]: print("Dette er mitt første Jupyter-program") #endre denne linjen
      print("Nå skal programmet stille et spørsmål")
      navn = input("Hva heter du? ")
      print("Hei", navn)

      alder = int(input("Hvor gammel er du? ")) # Her må du kun skrive et tall
      print("Da er du", alder + 5, "år gammel om 5 år")
```

2 Variabler mellom kodeblokker

Noe som er verdt å merke seg er at data kan eksistere mellom kodeblokkene i en Jupyter Notebook. La oss se på et eksempel. Trykk **ctrl + enter** i kodeblokken nedenfor slik at den kjører.

```
[ ]: message = "Wow! Dette var kult!"
```

Kjør så kodeblokken nedenfor:

```
[ ]: print(message)
```

Som du ser får vi printet ut verdien av `message` selv om `message` ikke er definert i den nederste kodeblokken. Dette kan være veldig praktisk, men kan noen ganger være forvirrende. Prøv å endre på verdien til `message` ("Wow! Dette var kult!") i den første kodeblokken, for så å trykke **ctrl + enter** i den andre blokken.

Som du ser er ikke `message` blitt oppdatert. Dette er fordi **vi er nødt til å kjøre kodeblokken med `message =` for at `message` skal bli oppdatert.**

Prøv nå å kjøre kodeblokken med `message =` igjen for så å kjøre blokken med `print` på nytt. Da burde riktig melding printes.

Oppgave c) Endre `message` til "Wow, Jupyter er kult!", og print det ut i blokken under.

Dette gjelder ikke bare for *variabler*, men også for *funksjoner*, som dere skal lære å bruke etterhvert. Hvis du skriver en funksjon og ønsker å bruke den i en annen kodeblokk må du kjøre kodeblokken hver gang funksjonen endres akkurat som med variabler.

3 Restarting dersom problemer skulle oppstå

Hvis du ønsker å fjerne all output fra dokumentet og *starte på nytt* kan du trykke på **Kernel -> Restart and clear output** i menyen på toppen. Det vil komme opp en boks med en skummel rød knapp, men dette går helt fint. Å kunne restarte kan også være nyttig hvis dokumentet henger seg opp. Dette skal vi se et eksempel på nå.

Kjør kodeblokken under to ganger uten å taste inn noe i inputfeltet som dukker opp (du må trykke på blokken igjen for å kjøre den andre gang).

```
[ ]: navn = input("Hva heter du?")
      print("Hei,", navn)
```

Dette var veldig uheldig. Nå skjer det ingenting og vi kan heller ikke kjøre andre kodeblokker i dokumentet :(

Når en kodeblokk venter på input kan man ikke kjøre andre kodeblokker, så hvis man da prøver å kjøre andre kodeblokker vil disse bli satt på vent.

Nå ser vi at det står **In [*]** ved flere av blokkene våre, dette betyr at de venter på andre blokker før de selv kjører, i vårt tilfelle kjørte vi input-blokken på nytt, uten å gi inn noe til forrige kjøring av blokken. Programmet venter fortsatt på input til forrige kjøring av blokken, selv om feltet er borte, som ikke er helt optimalt! Om du ikke forstår helt hva som skjer her er ikke det noe farlig. For å komme oss ut av dette kan vi restarte med **Kernel -> Restart and clear output** i toppmenyen. **Merk: Dette endrer ikke på koden du selv har skrevet.**

Oppgave d) Restart notebooken med kommandoen beskrevet over.

4 Endring av tekst

Det er også mulig å endre på teksten i et dokument ved å dobbelklikke på en tekstboks, men dette er noe du sjeldent trenger å gjøre. For å gjøre teksten «vanlig» igjen etter at du har endret trykker du her også på **ctrl + enter**.

Jupyter bruker noe som heter markdown til formatering av tekst. Dette er ikke pensum, men hvis du ønsker å se litt på det finnes det en ganske fin oversikt (på engelsk) [her](#). Det er også mulig å legge til LaTeX (et tekst-format til å lage fine matteuttrykk) i jupyter-tekstbokser. Dette er heller ikke pensum.

Hvis vi skal be dere om et tekst-svar vil vi dere se noe sånt som under. Her kan dere selv fjerne det som står inne i krokodilletegnene.

Oppgave e) Endre tekstboksen under til ITGK er gøy. *Merk: I en tekstboks trenger man ikke skrive python-kode*

Ditt svar: <ITGK er gøy>

5 En advarsel

Jupyter er generelt ikke så glad i at du har samme dokument åpnet i flere faner. Hvis du har dette er det mulig at endringene du gjør i den ene fanen overskriver endringene du gjør i en annen fane, noe som kan være uheldig. Sørg derfor for at du aldri har mer enn en fane åpnet med det samme dokumentet.

6 Til slutt

Det er mange muligheter som ligger inne i jupyter, og vi skal ikke bruke alt i dette kurset. Det er ingenting som hindrer dere fra å finne andre jupyter-notatbøker på nettet selv hvis dere ønsker mer utfordring eller å utforske hva som er mulig.

Lykke til videre med jupyter!