

Jensen, Schmeding

Jupyter notatbok i programmering

Etter at LK20 ble innført, er programmering blitt en del av det ordinære utdanningsløpet i flere fag. Matematikkfaget er utpekt som hovedansvarlig for opplæring i programmering. Et slikt ansvar betyr at elevene i matematikkfaget skal tilegne seg kunnskap i programmering ved å utforske matematiske kontekster, samtidig som programmeringen de lærer, skal gagne samfunnet (Vidje, 2019).

En utfordring for matematikklærere er å balansere opplæring i matematikk og opplæring i programmering. Hva elevene skal lære om programmering på hvert trinn, er riktignok definert gjennom egne kompetansemål, men det kan være vanskelig å tolke hva disse kompetansemålene innebærer (Mæland & Myklebust, 2022). Spesielt fordi målene ikke alltid er knyttet til spesifikke matematiske tema. Det kan derfor være

grunn til å tro at læreboka vil bli brukt som rettesnor for hvordan og når lærere skal inkludere programmering i undervisningen. Den fungerer for mange lærere som et bindeledd mellom kompetansemålene i læreplanen og den pedagogiske praksisen (Pepin et al., 2013).

En progresjon for opplæring i programmering som gjenspeiler seg i nye lærebøker og tilhørende digitale læremidler i matematikk, er at elevene begynner med aktiviteter knyttet til analog programmering og de grunnleggende læringsmålene knyttet til algoritmisk tenkning allerede på småtrinnet. Deretter fortsetter de med blokkprogrammering i Scratch eller tilsvarende, før de begynner med tekstbasert programmering i løpet av ungdomstrinnet. For eksempel innfører både Maximum fra Gyldendal og Matemagisk fra Aschehoug Python fra 8. trinn. I bøker for videregående opplæring, som Aschehougs Matematikk IT, møter elevene også Python.

Matematikklærere må med andre ord gi elevene grunnleggende opplæring i ulike programmeringsspråk, sørge for at undervisningen er i tråd med kompetansemålene, og finne gode innfallsvinkler til arbeid med programmering, samtidig som mange selv må lære seg å programmere. Dolonen et al. (2019) konkluderte med at få lærere hadde kompetanse til å undervise i programmering, og understreket behovet for å finne frem til gode læringsressurser og undervisningsopplegg. Selv om det er grunn til å tro

Mari Solberg Jensen

Nord universitet
mari.s.jensen@nord.no

Alexander Schmeding

NTNU
alexander.schmeding@ntnu.no

På Tangentens nettsted (<https://www.tangenten.no/jupyter>) ligger det en utvidet versjon av denne artikkelen med programkoder og lenker.

at mange lærere har økt sin kompetanse siden 2019, vil behovet for gode ressurser alltid være til stede. Hensikten med denne artikkelen er derfor å presentere en innfallsvinkel til arbeid med programmering for lærere på ungdomstrinnet som har opparbeidet seg kompetanse i Python.

For selv om elever har tilgang på læreboka og eventuelle digitale ressurser, er det ikke uvanlig at lærere bruker andre opplæringsressurser i undervisningen (Lepik et al., 2015). McEwan og Dubé (2015) påpeker i en undersøkelse om bruk av nettbrett i undervisning at selv om det finnes mange gode oppgaver, nettsider og læringsvideoer, så kan det være problematisk for elever å bruke forskjellige ressurser samtidig. Spesielt utfordrende er det å sjonglere mellom kilder som formidler ulike typer fagstoff om matematisk innhold, grunnleggende opplæring i programmering, programkoder og tilhørende forklaringer. Hvordan lykkes da?

For det første bør det være sammenheng mellom innhold i undervisning, oppgaver og vurderingssituasjoner, slik at elevene ikke selv må se hvordan ulike kilder relaterer seg til hverandre (Viberg et al., 2020, s. 10). For det andre er det viktig at det er sammenheng mellom design av aktiviteter, de digitale læremidlene og undervisningskonteksten for at integrering av digital

teknologi skal være vellykket (Drijvers, 2015). Hvilke læremidler som brukes, spiller derfor en avgjørende rolle, og det er viktig at de valgte læremidlene inkluderer både programmering og matematiske ideer samtidig. Dessverre viser aktuell forskning at dette ikke alltid er tilfellet (Viberg et al., 2020, s. 9). Et verktøy som derimot gjør det mulig å kombinere matematiske ideer og programmering, er Jupyter Notebook (heretter kalt Jupyter notatbok), et filformat som blant annet inkluderer Python-programkoder.

Artikkelen vil videre vise hvordan en Jupyter notatbok ser ut, og belyse noen muligheter filformatet gir for opplæring i programmering. I tillegg gir teksten eksempler på tilgjengelige programvarer, nettsider lærere kan bruke som opplæring, og materiell som kan brukes direkte i undervisning.

Jupyter notatbok

Fordelen med en Jupyter notatbok er at dette er en filtype som kan inkludere programkoder, tekst og illustrasjoner. Denne egenskapen gir elevene en ressurs hvor de kan samle oppgaver, forklaringer, nødvendig tilleggsinformasjon og kode på ett sted. For at dette skal være mulig, har en Jupyter notatbok to typer celler, som vist i Figur 1.



Figur 1: En kodecelle, etterfulgt av en tekstcelle med illustrasjon og forklaring av innholdet.

Tekstceller

I tekstcellene kodes det ikke. Her skrives tekst, slik det gjøres i en vanlig teksteditor som Microsoft Word. Læreren kan legge inn forklaringer og instruksjoner til elevene, og inkludere lenker og bilder, noe som ikke er mulig i en standard kodeeditor. I første omgang kan tekstceller brukes på samme måte som i Word, for å legge inn tekst med enkelt formatering. Er det ønskelig med flere og mer avanserte muligheter, kan mark-up-språket «Markdown» brukes i tekstcellene.

Kodeceller

Kodeceller kjører programkodene, i dette tilfellet i Python. Figur 1 viser en kodecelle, men det er mulig å inkludere flere kodeceller i samme Jupyter notatbok, og det kan kjøres flere kodeceller uavhengig av det som skjer i de andre cellene. Forklaringer og illustrasjoner kan plasseres mellom kodeceller ved at en legger til tekstceller.

Den største fordelen ved Jupyter notatbøker er nettopp at det er mulig å bruke flere kodeceller. Selv om hver celle inneholder en egen programkode, så kan flere kodeceller kjøres etter hverandre, en etter en. Det gjør det mulig å utføre lengre programmer eller algoritmer trinn for trinn. Python husker hvilke celler som allerede har blitt utført, og bruker tidligere beregninger og variabler. Hvis vi for eksempel setter en variabel a med verdi 1 i den første cellen, lagres denne verdien og er tilgjengelig i senere celler. Beregninger kan derfor benytte variabler som allerede er definerte.

Jupyter-notater lagres og vises ved bruk av en editor for Jupyter-notater eller online ved at en for eksempel bruker Google Colab, som er en gratis skytjeneste fra Google. Her har elevene mulighet til å laste inn Jupyter notatbøker og laste opp notater som læreren sender. Et annet alternativ er å dele notatene online, ved å laste notater direkte inn i Colab fra Github. Github er en kostnadsfri plattform på nett, hvor lærere kan laste opp og lagre filer, for eksempel i planleggingsfasen før timen. Elevene taster

inn navnet på lærerens GitHub-konto og kan begynne å arbeide med notatene. Notater, med endringer eller svar, lagres på Google Drive. Det er viktig å poengtere at det er enkelt å bruke GitHub for å dele notater, men at det ikke er nødvendig å bruke.

Bruk i klasserommet

En Jupyter notatbok kan i prinsippet inneholde alt fagstoff og alle oppgaver som trengs i ei undervisningsøkt. Lærere kan lage sine egne notater og introdusere en oppgave, en påstand eller lignende, og bruke de ressursene de ønsker. Et annet alternativ kan være å ta utgangspunkt i oppgaver som finnes i læreboka elevene bruker, og legge til nødvendige forklaringer, eksempler og kodeceller, slik at alt er samlet på ett sted. Det er også mulig å legge inn lenker til nettsteder. Da kan elevene fordype seg i relevant fagstoff, for eksempel i tverrfaglige opplegg, eller tekstcellene kan brukes for å hjelpe elever som trenger tilpasninger. Læreren kan også legge inn ferdige koder hvor elevene skal finne feil, eller som elevene skal forbedre eller endre. I lignende oppgaver i lærebøker må elevene ofte plote disse kodene først. Når koden allerede er lagt inn i en kodecelle, kan elevene bruke tiden sin på det oppgaven spør etter.

Når læreren har lagt en Jupyter notatbok på forhånd, kan den brukes både til gjennomgang av matematikk og programmering og for å aktivisere elevene. I timen vises Jupyter-notatboken på Smartboard eller lignende, og materialet kan brukes til gjennomgang sammen med elevene. Under gjennomgangen kan elevene ha en kopi av notatet på sin PC, og legge inn egne notater. I tillegg kan læreren enkelt bytte undervisningsform fra gjennomgang av fagstoff til individuelt arbeid, noe som også gjør Jupyter notatbøker egnet for bruk i digital undervisning.

Selvstendig arbeid

Når elever skal lære å bruke et nytt digitalt verktøy, som Python, må elevene både lære seg hvordan de bruker verktøyet rent teknisk, og hvor-

dan det kan utnyttes med tanke på det faglige innholdet (Viberg, 2020). Bruk av Jupyter notatbøker gjør det enkelt å ivareta begge deler, siden det er mulig å kombinere koding og forklaringer av både matematikk og programmering. Elevene har oppgaver i Jupyter-notatboken, og løser disse ved å kode i kodeceller i det samme dokumentet eller ved å legge inn svar og forklaringer ved hjelp av tekstceller. Jupyter notatbøker kan dermed utformes slik at de er selvforklarende og egnet for bruk når elevene skal arbeide selvstendig. Dette kan være enten i undervisningsøkter, som hjemmearbeid, eller for å undervise elever som ikke er fysisk til stede. I tekstceller er det også mulig å legge inn hint eller løsninger på oppgaver som ikke umiddelbart er synlig for elevene, som vist i Figur 2.

Elevene ser ikke hintet eller løsningen før de trykker på en knapp i teksten i Jupyter notatboken.

Vurdering

Jupyter notatbøker egner seg også for å lage oppgaver som læreren ønsker å vurdere. I Google Colab finnes det en funksjon som evaluerer hele Jupyter-notatboken ved å tilbakestille alt som er gjort tidligere, og kjøre alle kodecellene elevene har programmert i etter hverandre. Colab begynner med den øverste cellen i notatet og jobber seg nedover til alle celler er evaluert, eller til første feil i koden blir oppdaget. Å evaluere et notat er ikke nyttig bare når elevene skal bygge opp store program. Siden automatisk evaluering av et notat avsluttes når Python oppdager første feil i en kodecelle, avdekkes det også om elevene har klart å lage koden uten programmeringsfeil. Oppgavearket retter seg med andre ord selv, noe som sparer lærere for mye arbeid. Riktignok

avdekkes det ikke om koden er riktig svar på oppgaven som ble gitt. Viberg (2020) poengterer at det er viktig å overvåke elevenes bruk av teknologi og sette dem på rett spor om det er behov for det. Funksjonen vi nå har omtalt, kan gi lærerne en slik mulighet, samtidig som elevene kan bruke den selv for å se om det de har gjort, gir forventet resultat.

Figur 3 og 4 viser et eksempel på hvordan et slikt oppgaveark kan se ut. Elevene skal i denne oppgaven lære å lage sine egne funksjoner i Python, for eksempel gjennom Collatz-sekvensen. Sekvensen fungerer slik at det for et gitt heltall n lages nye tall, enten ved å halvere n hvis n er partall, eller ved å multiplisere med 3 og legge til 1 hvis n er oddetall. Figur 3 viser en oppgave som ber elevene om å lage en funksjon vi kaller «minCollatz». Denne funksjonen skal skrive ut Collatz-sekvensen for et tall brukeren oppgir.

Introduksjon til funksjonen er allerede gitt i Jupyter-notatboken. For å teste funksjonen legger læreren til en ny celle, som vist i figur 4.

Det forhåndsdefinerte navnet på funksjonen brukes for å teste for ulike eksempelverdier. For disse verdiene kan resultatet beregnes, eller be elevene om å beregne dem selv, uten bruk av datamaskin. Evalueres hele notatet, vil funksjonen elevene har lagd, automatisk bli brukt.

Hvis man vil bruke Jupiter notatbok, finnes det ferdige notater på nettet, som er lagd for bruk i ungdomsskolen og videregående skole. Dermed kan lærere ta Jupyter-notater i bruk, utforske og bli kjent med mulighetene de gir. Det er ikke et mål at all undervisning i Python skal bli gitt ved bruk av Jupyter-notater, men formatet gir altså noen fordeler som gjør opplæring i programmering enklere, ryddigere og mer effektivt for både lærere og elever.

```
[ ] # Skriv kode ditt under og trykk på Shift+Enter for å kjøre
```

Dobbelklikk her for løsningsforslag.

Figur 2: Løsningen er bare synlig etter et dobbelklikk på linjen som inneholder «her».

```
[ ] def minCollatz (x):  
    #Skriv en forklaring her, som forklarer hva funksjonen minCollatz gjør  
  
    #Skriv koden din her
```

Figur 3: Eksempeloppgave Collatz-sekvensen.

```
[ ] #Ikke endre noe i cellen, dette er bare for å teste  
    #funksjonen du har laget  
  
minCollatz(5) #5, 16, 8, 4, 2, 1 (sjekk første del av oppgaven)  
  
minCollatz(7) #Forventede resultat:  
              #7,22,11,34,17,52,26,13,40,20,10,5,16,8,4,2,1  
minCollatz(13) #Forventede resultat: 13,40,20,10,5,16,8,4,2,1
```

Figur 4: Kodecelle som tester funksjonen elevene skulle lage.

Referanser

- Dolonen, J. A., Kluge, A., Litherland, K. & Mørch, A. I. (2019). *Litteraturgjennomgang av programmering i skolen*. <http://urn.nb.no/URN:NBN:no-79405>
- Drijvers, P. (2015). Digital Technology in Mathematics Education: Why It Works (Or Doesn't). I S. J. Cho (Red.), *Selected Regular Lectures from the 12th International Congress on Mathematical Education*, DOI 10.1007/978-3-319-17187-6_8
- Lepik, M., Grevholm, B. & Viholainen, A. (2015). Using textbooks in the mathematics classroom – the teachers' view. *Nordic Studies in Mathematics Education*, 20(3–4), 129–156.
- McEwen, R. N. & Dubé, A. K. (2015). Engaging or Distracting: Children's Tablet Computer Use in Education. *Educational Technology & Society*, 18(4), 9–23.
- Mæland, M. D. & Myklebust, M. (2022). Forventning, forvirring og forundring. *Tangenten – tidsskrift for matematikkundervisning*, 33(1), 20–26.
- Pepin, B., Gueudet, G. & Trouche, L. (2013). Investigating textbooks as crucial interfaces between culture,

- policy and teacher curricular practice: two contrasted case studies in France and Norway. *ZDM – The international Journal on Mathematics Education*, 46, 685–698.
- Viberg, O., Grönlund, Å. & Andersson, A. (2020). *Integrating digital technology in mathematics education: a Swedish case study*. Interactive Learning Environments.
- Vidje, G. (2019). Nye læreplan skal ivareta alle elever. *Statped Magasinet 2*, 54–55.

Nyttige lenker

- Google Colab: <https://colab.research.google.com/>
- Introduksjon til Python i Jupyter: <https://github.com/ASchmeding/Intro-2-Python-norsk>
- Jupyter notatbøker for ungdomsskolen: <https://github.com/ASchmeding/JuPyter-for-ungdomsskole>
- Edudata (Jupyter notater for undervisning i norske skoler): <https://edudata.no>