

Kaares kokebok i programmering

for matematikk i
ungdomsskolen og VG1

2. utgave

Kaare E. Jørgensen

Stein A. Dahl

KAARE ERLEND JØRGENSEN
STEIN ALEXANDER DAHL

KAARES KOKEBOK I PROGRAMMERING

FOR MATEMATIKK I
UNGDOMSSKOLEN OG VG1

2. UTGAVE

Jørgensen Matematiske Ressurser

© 2. utgave Jørgensen Matematiske Ressurser 2021
1. opplag 2021

ISBN 978-82-692219-4-7

Materialet i denne publikasjonen er vernet etter åndsverkloven. Uten uttrykkelig samtykke fra rettighetshaverne er eksemplarframstilling, som utskrift og annen kopiering, bare tillatt når det er hjemlet i lov eller avtale med Kopinor (www.kopinor.no).

Utnyttelse i strid med lov eller avtale kan medføre erstatnings- og straffeansvar.

Boken har egen nettside: www.kaareskokebok.no

Henvendelser om denne utgivelsen kan rettes til: kaare@kaareskokebok.no

Omslag og omslagsillustrasjon: Kaare E. Jørgensen & Stein A. Dahl
Sats, figurer og formgivning: Kaare E. Jørgensen & Stein A. Dahl
Boken er satt med: Computer Modern 10/12, pdf_latex/TikZ/tcolorbox
Grafisk produksjon: Bokstav og Bilde AS

Takk

«Of all the characteristics needed for both a happy and morally decent life, none surpasses gratitude. Grateful people are happier, and grateful people are more morally decent.»

*Dennis Prager*¹

Disse kloke ordene fra Dennis Prager minner meg på at jeg har mye å være takknemlig for, og jeg har mange personer å takke. Jeg vil starte med å takke min kone Inna Jørgensen for at hun aldri mistet troen på bokprosjektet mitt. Hun var villig til å yte ekstra med den daglige driften med hus, hjem og barn. Min medforfatter Stein A. Dahl har vært uvurderlig med å løse tekniske og designmessige L^AT_EX-problemer. Han har forfattet flere viktige deler av boka, og gjennomgått det faglige med stor grad av nøyaktighet. Jeg vil takke min mor Turid Jørgensen for at hun har vært en meget god avlastning under mange travle dager og en god mor jeg har kunnet snakke med når ting har vært tøft. Jeg vil også takke min far Åge J. Marthinsen for at han har lært meg verdien av hardt arbeid, disiplin og høy arbeidsmoral.

Videre vil jeg takke Ellen Egeland Fløe for at hun anbefalte meg som kursholder til BRO AOF. Gjennom dem har jeg holdt flere kurs rundt omkring i det langstrakte Norge, fra Karasjok til Sarpsborg, som alle gav viktig kunnskap og erfaring om hvordan elever bør lære programmering. Jeg retter dessuten en takk til Nina Sundmark for nyttige tilbakemeldinger tidlig i prosjektet. En varm takk går også til Thomas F. Sturm², som har vært til stor hjelp med utformingen av fargede bokser, der blant annet koden er plassert. Gang på gang har jeg bladd i håndboken hans på 525 sider og oppdaget nye idéer til utformingen. En takk går for øvrig til Till Tantau³ for hans hjelp med grafer og figurer i Tikz. Håndboken hans på 1318 sider er litt av et kunstverk. Jeg takker videre Master Books forlag for at de har gitt meg tillatelse til å benytte eksempler og oppgaver fra den fantastiske geometriboka av Harold R. Jacobs⁴. Til slutt en takk til Inga Stener Olsen for konstruktive og verdifulle tilbakemeldinger til innledningen, til Audun Åby for forslaget om å lage stikkordsregister, og til Isabella Dahl for den siste runden med korrekturlesning.

Kaare Erlend Jørgensen

¹Prager, Happiness is a serious problem [6]

²Sturm, The tcolorbox package [12]

³Tantau, The Tikz and PGF manual [13]

⁴Jacobs, Geometry: Seeing, Doing, Understanding [4]

Innhold

Innledning	ix
------------	----

I	Det grunnleggende	1
----------	--------------------------	----------

1	De første instruksjonene	3
----------	---------------------------------	----------

1.1	Skapelsen	3
1.2	De første skrittene	5
1.3	Oppskrift på trekant	6
1.4	Tegne kvadrat og rektangel	7
1.5	Tegne med koordinater	8
1.6	Sirkler og farger	9
1.7	Oppsummering kapittel 1	10
1.8	Oppgaver	11

2	Gjentakelser med løkker	13
----------	--------------------------------	-----------

2.1	To gjentakelser	13
2.2	Tre eller flere gjentakelser	15
2.3	Vinkler og regulære mangekanter	16
2.4	Trapper opp og trapper ned	17
2.5	Utforskende kunst	19
2.6	Oppsummering kapittel 2	20
2.7	Oppgaver	21

3	Lagre kode i funksjoner	23
----------	--------------------------------	-----------

3.1	Mitt smil	23
3.2	Tannhjul og kvadrater	25
3.3	Ulike størrelser på kvadrater	27
3.4	Tunneffekt	28
3.5	Trekantmønster med midtpunkter	30
3.6	Oppsummering kapittel 3	31
3.7	Oppgaver	32

4	Flere muligheter	35
----------	-------------------------	-----------

4.1	Et hjul med trekanter	35
4.2	Tilfeldige vinkler	37
4.3	Den omskrevne sirkelen til en trekant	38
4.4	Blink	39
4.5	Avstandsalarm	41

4.6	Oppsummering kapittel 4	43
4.7	Oppgaver	44
5	Matematikk i Python	47
5.1	De fire regneartene	47
5.2	Potenser og parenteser	49
5.3	Kvadratrot	49
5.4	Bokstavregning	50
5.5	Variabler og avrunding	51
5.6	En høflig hilsen	52
5.7	Kalkulatorprogram	53
5.8	Bli en gangemester	54
5.9	Oppsummering kapittel 5	56
5.10	Oppgaver	57
II	Matematiske emner	61
6	Tallmønstre	63
6.1	Et kjent tallmønster	63
6.2	Apepopulasjon med hurtig vekst	64
6.3	Summer av oddetall	65
6.4	Sum av et geometrisk tallmønster	66
6.5	Tallmønster med sirkler	67
6.6	De 100 første naturlige tallene	69
6.7	Rektangelmønster	70
6.8	Oppsummering kapittel 6	72
6.9	Oppgaver	73
7	Potenser og kvadratrøtter	77
7.1	Drillprogram for toerpotenser	77
7.2	Utforske tierpotenser	79
7.3	Gjette på kvadratroten	81
7.4	Riskorn og sjakkbrett	82
7.5	Oppsummering kapittel 7	83
7.6	Oppgaver	84
8	Koordinatsystemet	85
8.1	Koordinatsystemet – et kjapt overblikk	85
8.2	Utforske x -koordinater med Python Turtle	86
8.3	Utforske y -koordinater med Python Turtle	87
8.4	Parallelle linjer	88
8.5	Oppsummering kapittel 8	89

8.6	Oppgaver	90
9	Funksjoner	93
9.1	Kaare på kaféen – uten funksjon	93
9.2	Kaare på kaféen – med funksjon	94
9.3	Pris per treningsøkt	95
9.4	Seter i en kinosal	96
9.5	Skoleveien	97
9.6	Bestemme bunnpunkt	99
9.7	Et maleri synker i verdi	100
9.8	Oppsummering kapittel 9	101
9.9	Oppgaver	102
10	Sannsynlighet	105
10.1	Simulere 100 fødsler	105
10.2	Telle opp antall jenter	106
10.3	To terninger	107
10.4	Mange terningkast	108
10.5	Kuler i eske – med tilbakelegging	109
10.6	Kuler i eske – uten tilbakelegging	110
10.7	To geiter og en bil	111
10.8	Skiskyting – visuell simulering	113
10.9	Oppsummering kapittel 10	114
10.10	Oppgaver	115
	Tillegg	117
A	Løsninger	117
A.1	Kapittel 1 – De første instruksjonene	117
A.2	Kapittel 2 – Gjentakelser med løkker	121
A.3	Kapittel 3 – Lagre kode i funksjoner	123
A.4	Kapittel 4 – Flere muligheter	128
A.5	Kapittel 5 – Matematikk i Python	132
A.6	Kapittel 6 – Tallmønstre	135
A.7	Kapittel 7 – Potenser og kvadratrøtter	139
A.8	Kapittel 8 – Koordinatsystemet	141
A.9	Kapittel 9 – Funksjoner	144
A.10	Kapittel 10 – Sannsynlighet	148
B	Tegne grafer i Python med Spyder	153
C	Læringsfilosofi	161
	Bildeliste	167

Figurer	168
Python-ord	171
Stikkordsregister	173
Bibliografi	182

Innledning

Hva handler denne boka om?

Boka gir opplæring i grunnleggende programmering i språket *Python*. I den første delen blir du introdusert for programmeringskonsepter på en visuell måte. Den andre delen integrerer programmering i sentrale matematiske emner. En rekke eksempler og oppgaver kan også knyttes til andre fag, som for eksempel kunst og håndverk og naturfag. I tråd med fagfornyelsen, demonstrerer boka hvordan programmering kan fungere som et nyttig verktøy i utforskning, eksperimentering og problemløsning.

Hvem er boka for?

Boka er i hovedsak for elever og lærere som har matematikk på ungdomsskolen, eller matematikkfagene 1P og 1T på videregående skole. Den er også for nybegynnere i programmering, herunder foreldre og foresatte. Faglærere kan bruke boka som ferdige undervisningsopplegg. Se gjerne tillegg C for de didaktiske prinsippene som ligger til grunn for boka.

Hvordan bør jeg lese denne boka?

Les de fem første kapitlene i den rekkefølgen de er skrevet. Ferdigheter du lærer i kapittel 1, vil du bruke videre i kapittel 2. I kapittel 4 forutsetter eksemplene og oppgavene at du har jobbet deg gjennom kapittel 1, 2 og 3. Fra og med kapittel 6 kan du hoppe til eksempler og oppgaver som er relevante.

Hvor skal jeg skrive koden?

Du trenger ikke å installere noe programvare for å komme i gang med programmering. Alle kodeeksemplene i boka er skrevet på nettstedet replit.com. Her bruker vi to versjoner av Python: *Python (with Turtle)* og *Python*⁵. Følgende to alternative nettsider er verdt å nevne:

- *Python Turtle*: trinket.io/features/pygame og pythonandturtle.com/turtle. Merk at på disse to nettsidene er det nødvendig å avslutte alle Turtle-programmer med kodelinja `done()`.
- *Python3*: trinket.io og pythonandturtle.com. På førstnevnte nettsted velger du New Trinket Python3.

⁵Når du velger *Python*, så velger du per dags dato *Python 3*. Unngå den eldre *Python 2*, da syntaksen til en del kommandoer i den versjonen er forskjellig.

Hva er nytt i 2. utgave?

Bokas 2. utgave inneholder en rekke strukturelle og designmessige forbedringer, herunder:

- Alle kapitlene avsluttes nå med både en **oppsummeringsside** og en **oppgavedel**.
- De fleste kapitlene er utvidet med mange **nye og spennende oppgaver**.
- Det er laget **komplette løsninger** til alle oppgaver. Løsningene finner du bak i boka.
- Det er utferdiget en **figurliste** over bokas mange illustrasjoner, en oversiktlig **indeks** over alle brukte Python-ord og en rik **stikkordsliste** over norske ord.

Hva betyr symbolene i margen?

I margen vil du av og til se et sidetall i en lilla boks. Dette forteller deg hvor du finner løsningen på oppgaven du holder på med. Lesere av digitalboka kan trykke på dette sidetallet for å navigere direkte til løsningen. Et trykk på overskriften til løsningen vil ta deg tilbake til oppgaven igjen.

117



Stjerne-symbolet betyr at denne oppgaven er utfordrende, og at du ikke finner løsningsforslag til den i boka.



Skilpadde-symbolet betyr at koden skal skrives med språket *Python Turtle*. Gå til replit.com, lag en ny fil og velg språket *Python (with Turtle)*. Hvis symbolet mangler, skal koden skrives på språket *Python*. Du kan også bruke lenkene replit.com/new/python_turtle og replit.com/new/python3.

Lykke til med opplæringen!

Kaare Erlend Jørgensen

Stein Alexander Dahl

DEL I

Det grunnleggende

KAPITTEL 1


De første instruksjonene

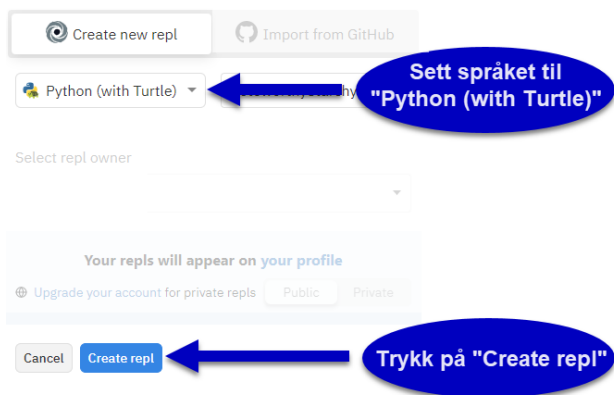
1.1 Skapelsen



Figur 1.1: En ekte skilpadde.

Vi skal nå fortelle datamaskinen at vi ønsker å skape en skilpadde. Vi kaller skilpadda Tom, og den skal gjøre nøyaktig det vi instruerer den. Tom forstår ikke norsk, bare språket Python.

- Åpne nettsiden replit.com. Lag en bruker, og logg inn. Anbefalte nettlesere er *Brave* og *Chrome*.
- Lag en ny fil ved å trykke på  øverst til høyre. Velg så språket «Python (with Turtle)», og trykk deretter på «Create repl». Se figur 1.2.



Figur 1.2: Lag en ny fil på replit.com.

- c) Skriv følgende kode for å skape skilpadda Tom.

Kode

```
1 from turtle import *
2 shape("turtle")
```



Figur 1.3: Resultatet av koden.

Når du skal kjøre programmet, trykker du på **Run ▶**, eller på **Ctrl** + **Enter** (Mac: **Cmd** + **Enter**) på tastaturet. Ser du skilpadda, slik figur 1.3 viser, eller fikk du en feilmelding?

- d) Hvis du ser skilpadda, går du til e) nå. Hvis du fikk en feilmelding (error), må du kontrollere om du kjenner igjen feilen nedenfor. Rett opp til den riktige koden.

Riktig	Feil
<pre>from turtle import *</pre>	<pre>from Turtle import * from turtle import* from turlte import * from import turtle *</pre>
Riktig	Feil
<pre>shape("turtle") shape('turtle')</pre>	<pre>shape 'turtle') shape(turtle') shape(turtle) shape('Turtle')</pre>

Merk at du kan skrive enten `shape("turtle")` eller `shape('turtle')`.

- e) Gratulerer! Nå har du skapt skilpadda og er klar for å gi den instruksjoner.

1.2 De første skrittene

Kode

```
1 from turtle import *
2 shape('turtle')
3 speed(1)
4 forward(50)
5 left(30)
```

117



Skilpadda Tom forstår instruksene `forward` og `left`. Han forstår ikke «forover» og «snu mot venstre».

- Les koden ovenfor. Hva tror du resultatet av den blir?
- Skriv av koden, og kjør programmet. Trykk `Run ▶` eller `Ctrl + Enter` (Mac: `Cmd + Enter`) for å kjøre programmet.
- La kodelinjene `forward(50)` og `left(30)` bytte plass. Gjøtt hva resultatet blir, og kjør deretter programmet.
- Legg til følgende instruksjer nederst i koden. Gjøtt hva resultatet blir, og kjør deretter programmet.

```
left(-30)
forward(50)
```

- La nå `left(30)` og `left(-30)` bytte plass. Gjøtt hva resultatet blir, og kjør deretter programmet.
- Skriv ferdig koden slik at resultatet blir figur 1.4.



Figur 1.4: Strektegning.

Bak i boka finner du løsninger til oppgavene. Løsningen til dette delkapitlet, delkapittel 1.2, finner du på side 117. Dette sidetallet er markert i margin som

117.

1.3 Oppskrift på trekant

117



Kode

```
1 from turtle import *
2 shape('turtle')
3 color('blue')
4 forward(75)
5 left(60)
6 forward(75)
```

Oppskrift

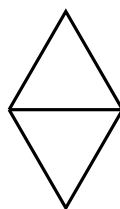
Skap skilpadda
Sett formen til en skilpadde
Sett fargen til blå
Gå 75 skritt forover
Snu 60 grader mot venstre
Gå 75 skritt forover

Å skrive kode kan sammenliknes med å skrive en oppskrift. Koden er på språket Python, mens oppskriften er på norsk.

- Les koden over. Gjett hva resultatet blir.
- Skriv av koden og trykk så på **Run ▶** eller **Ctrl + Enter** (Mac: **Cmd + Enter**) for å kjøre den. Hvordan stemte gjetningen din med resultatet?
- Fargelegg Tom med din favorittfarge¹ ved å endre `color('blue')` til en annen farge som for eksempel `color('brown')`.
- Endre `left(60)` til `left(120)`, og kjør programmet.
- Legg til kodelinjene `left(120)` og `forward(75)` i riktig rekkefølge slik at resultatet blir figur 1.5.



Figur 1.5: En likesidet trekant.



Figur 1.6: To likesidede trekanter.

- Legg til kodelinja `hideturtle()` nederst i programmet.
- Utvid programmet slik at resultatet blir figur 1.6.
- Sammenlign din løsning med den du finner bak i boka på side 117. Dette sidetallet er også markert i margen som **117**.

¹På nettsiden trinket.io/docs/colors kan du trykke på en farge og se hva den kalles i Python Turtle.

1.4 Tegne kvadrat og rektangel

Kode

```
1 from turtle import *
2 shape('turtle')
3 forward(200)
4 left(90)
5 # Og så videre
```



Figur 1.7: Kvadrat tegnet med Python.

På kodelinje 5 har vi laget en *kommentar* ved å skrive # før teksten. Kommentarer brukes blant annet til å forklare hva koden betyr.

- Skriv av koden, og kjør programmet.
- Fortsett videre på koden og skriv instruksene² `forward(200)` og `left(90)` tilstrekkelig mange ganger og i riktig rekkefølge, slik at kvadratet blir tegnet ferdig.
- Legg til kodelinja `speed(7)` rett under `shape('turtle')`.³ Hva skjer? De mulige tallene er 0 til 10.
- Endre to av tallene i koden slik at resultatet blir et rektangel med bredde 100 og lengde 200. Se figur 1.8.



Figur 1.8: Rektangel.

- Legg til kode for å tegne figur 1.9.



Figur 1.9: Rektangel delt i to.

Benytt deg av `color('red')` for å fargelegge det ene linjestykket rødt.

²Begrepene *kommando*, *kodelinje* og *instruks* betyr omtrent det samme.

³Nettsiden docs.python.org/3/library/turtle.html inneholder en oversikt over tilgjengelige kommandoer i Python Turtle.

1.5 Tegne med koordinater

118



Kode

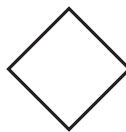
```
1 from turtle import *
2 shape('turtle')
3 penup()
4 goto(100, 0)
5 pendown()
6 goto(0, 100)
```

Tom forstår instruksene `goto` , `penup()` og `pendown()` .

- a) Les koden ovenfor, og gjett på hva resultatet blir.
- b) Skriv av koden, og kjør programmet.

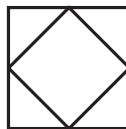
Når vi skal teste hva som skjer hvis vi fjerner en kodelinje, kan vi kommentere ut kodelinja med `#penup()` i stedet for å slette den.

- c) Undersøk hva som skjer hvis `penup()` fjernes fra programmet. Gjør dette ved å endre til `#penup()` .
- d) Endre `#penup()` tilbake til `penup()` . La nå `penup()` og `pendown()` bytte plass. Gjett alltid på resultatet før du kjører programmet på nytt.
- e) Legg til kodelinjen `goto(-100, 0)` nederst i programmet.
- f) Fullfør koden slik at du får figur 1.10.



Figur 1.10: En rombe.

- g) Legg til flere `goto` -kommandoer for å tegne figur 1.11.



Figur 1.11: En rombe i et kvadrat.