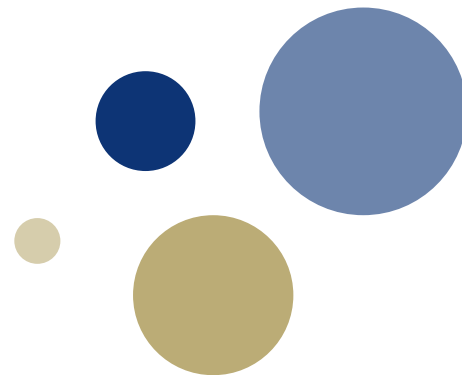




NTNU – Trondheim
Norwegian University of
Science and Technology



«Prosjekt Python»

Pilot 2019/2020

Egil Krystad
Universitetslektor, matematikk
Forkurs NTNU Trondheim

Høringsutkast 2019 (Frist 18/6)

- Læreplan i matematikk R2



Noen sentrale kompetansemål er fjernet:

- Differensiallikninger
- Sannsynlighetsregning

Et par nye kompetansemål, relevante for programmering:

- bruke programmering til å utforske rekursive sammenhenger og presentere egne framgangsmåter
- formulere algoritmer som beregner bestemte integraler numerisk og bruke digitale verktøy til å utføre algoritmene

(Fysikk ikke på høring ennå)

Hva?

- **Grunnleggende programmering i Python** for forkursstudenter
- Foreleses ikke; frivillig pilotprosjekt med innleveringer
- Basert på matematikkpensumet

Python (programming language)

From Wikipedia, the free encyclopedia

For other uses, see [Python](#).

Python is an [interpreted](#), [high-level](#), [general-purpose programming language](#). Created by [Guido van Rossum](#) and first released in 1991, Python's design philosophy emphasizes [code readability](#) with its notable use of [significant whitespace](#). Its language constructs and [object-oriented](#) approach aim to help programmers write clear, logical code for small and large-scale projects.^[27]

Hvorfor?

- **Beregningsbaserte digitale hjelpemidler** på frammarsj i VGS
- Python brukes i økende grad for hvert år på...
 - ingeniørstudier ved NTNU
 - flere andre institusjoner
 - arbeidsplasser
- Python er gratis
- Beregnings-/ objektorientert programmering kan gi et annet og dypere syn på realfagene



Hvordan?

- Frivillig prosjekt med innleveringer
- Tar hovedsakelig utgangspunkt i pensum i matematikk (fysikk også mulig)
- «**Learning by doing**» - Studentene tester ut kode først; leser forklaring og besvarer spørsmål etterpå.

Oppgave 1

PP1 Kom i gang: «Hei verden»

1. Gå til www.anaconda.com. Last ned og installer [Anaconda](#).
2. Etter installasjon skal programmet [Spyder](#) ligge i en mappe som heter [Anaconda](#) (på start-meny e.l.). [Spyder](#) er IDE-en (Integrated Development Environment) som vi skal bruke til å skrive Python-kode og kjøre den. [Spyder](#) er blant de mest populære IDE-ene for Python og er veldig fin til både små og store Python-programmer. Det er selvsagt helt i orden å bruke andre IDE-er om du vil, men dette kurset kommer til å basere seg på [Spyder](#).
3. Lag en tom mappe på datamaskinen hvor du vil oppbevare Python-programmene dine.
4. Åpne [Spyder](#), og skriv ditt første Python-program:

- Lag en ny fil.
- Lagre fila med filnavn **heiVerden.py** i ønsket mappe.
- Alt som står i fila fra før, kan du bare la stå. Det er ikke en del av programmet. Helt øverst i fila vår skal det stå følgende (for å tillate bokstavene æ, ø og å): `#!/usr/bin/env python3`
utf-8 -*- coding: utf-8 -*-
Skriv inn koden under. Rød tekst skal byttes ut med dine egne data.

```
navn = "Fornavnet ditt"
fodselsar=Fødselsåret ditt
print("Hei verden! Her er et program skrevet av "+navn+".")
# Her kommer en if/else-setning!
if len(navn)>8:
    print("Jeg har et langt navn.")
else:
    print("Jeg har et kort navn.")
alder=2019-fodselsar
print("Jeg fyller "+str(alder)+" år i år.")
```

5. Trykk F5 (eller grønn play-knapp øverst i [Spyder](#)) for å kjøre programmet. Resultatet skal dukke opp i vinduet nederst til høyre (Console).
6. Gratulerer! Du har skrevet og kjørt et Python-program. Prøv å leke deg litt med koden ved å endre på ting.
7. Du kan også skrive direkte i Console for å teste små kodelinjer. Prøv f.eks. å skrive `a=1910`
`print("NTH ble opprettet i "+str(a))`

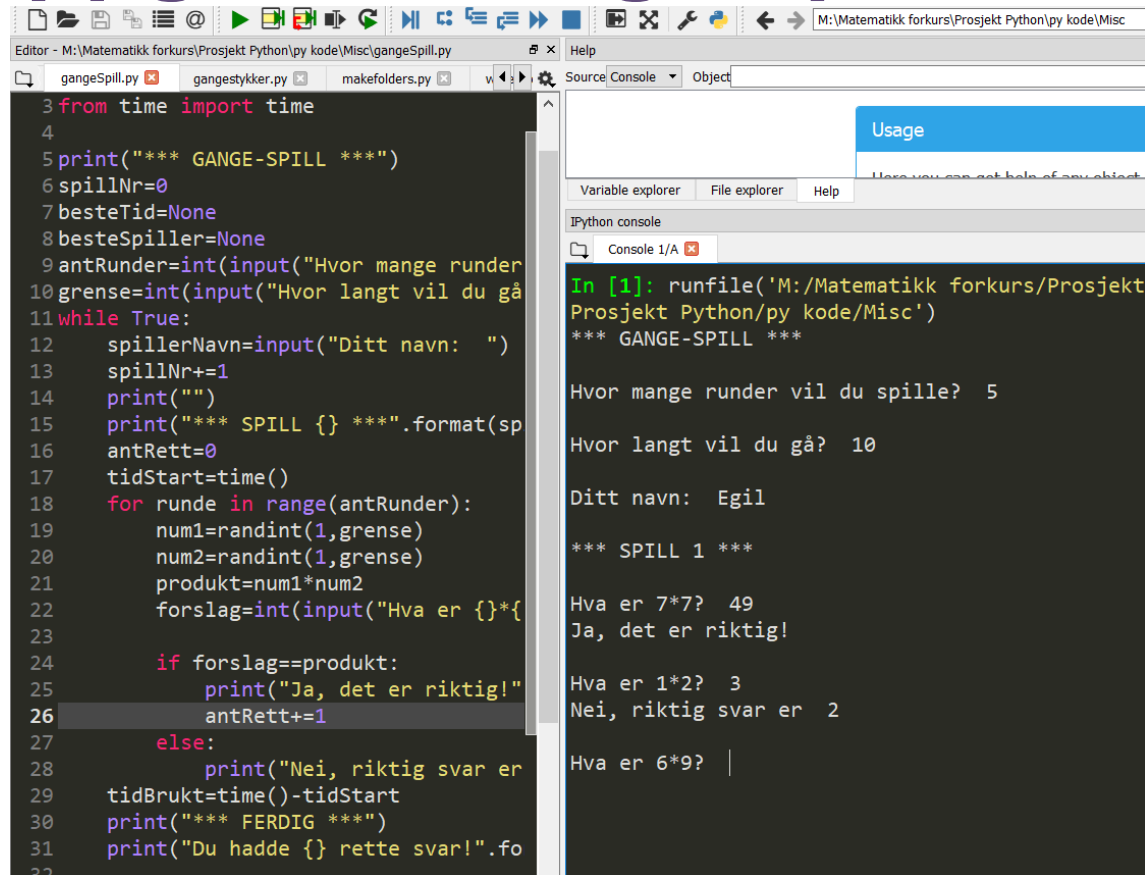
Forklaring:

- I koden vår opprettet vi en variabel som het navn. Variabler kan vi kalle hva som helst. En variabel oppretter vi for senere å bruke den. Dette er særlig nyttig hvis en verdi skal brukes flere ganger. Det går også an å endre en variabel etter at den er opprettet.
- Variabler som vi vil at Python skal oppfatte som tekst, må vi bruke hermetegn (") rundt. Da har vi skrevet en *tekststreng* (*string*). I stedet for hermetegn kan vi også bruke *apostrof* (').
- Tekst (*str*) og heltall (*int*) er to *datatyper*. Skriver vi et heltall, oppfattes det automatisk som datatypen *int*.
- I Python (og de fleste andre programmeringsspråk) bruker vi veldig ofte ulike *funksjoner*. Da skriver vi:
funksjonsnavn(argument1, argument2, ...). De tre funksjonene vi har brukt i *heiVerden.py* er blant de aller vanligste. Det er veldig vanlig at funksjoner *returner* noe. Funksjonen *str(...)* returnerer alltid en tekststreng. Funksjonen *len(...)* returnerer alltid et heltall.
- Linjene 5-8 i koden vår kalles en *if/else-setning*. Du la kanskje merke til at et innrykk ble lagt til automatisk når du trykket *enter* etter den første linje av *if/else*-setningen. Disse innrykkene kan være på kun ett mellomrom hvis vi vil, men det er lurt å holde seg til samme avstand hver gang. Vi kan trykke TAB-tasten for å få et passelig innrykk. Innrykk bruker vi til å lage en *blokk*. En blokk kjøres hvis den betingelsen som ble satt på linja over blokka er oppfylt. Innrykk er en veldig praktisk måte å gruppere på, og er noe av det som har vært med på å gjøre Python så populært.

Oppgaver:

1. Når spiller mellomrom en rolle i koden (utenfor tekststrenger)? Når spiller de ingen rolle?
2. Hva skjer når vi skriver et pluss etter en tekststreng, etterfulgt av en annen tekststreng?
3. Hvilken funksjon skriver vi i *.py*-fila for å få tekst eller tall til å skrives til Console?
4. Hva heter de tre funksjonene vi har brukt i *HeiVerden*?
5. Hva returnerer funksjonen *len(...)* når den tar inn en tekststreng?
6. Hvilket krav stiller vi i *if/else*-setningen for at første blokk skal kjøres?

Oppgave: Gange-spill



The image shows a Python IDE window with a code editor on the left and an IPython console on the right. The code in the editor is a game program that asks for the number of rounds and the length of the game, then plays a multiplication game for each round. The console shows the program's execution, including user input and output.

```
3 from time import time
4
5 print("*** GANGE-SPILL ***")
6 spillNr=0
7 besteTid=None
8 besteSpiller=None
9 antRunder=int(input("Hvor mange runder
10 grense=int(input("Hvor langt vil du gå
11 while True:
12     spillerNavn=input("Ditt navn: ")
13     spillNr+=1
14     print("")
15     print("*** SPILL {} ***".format(sp
16     antRett=0
17     tidStart=time()
18     for runde in range(antRunder):
19         num1=randint(1,grense)
20         num2=randint(1,grense)
21         produkt=num1*num2
22         forslag=int(input("Hva er {}*{
23
24         if forslag==produkt:
25             print("Ja, det er riktig!"
26             antRett+=1
27         else:
28             print("Nei, riktig svar er
29             tidBrukt=time()-tidStart
30             print("*** FERDIG ***")
31             print("Du hadde {} rette svar!".fo
32
```

IPython console output:

```
In [1]: runfile('M:/Matematikk forkurs/Prosjekt Python/py kode/Misc')
*** GANGE-SPILL ***
Hvor mange runder vil du spille? 5
Hvor langt vil du gå? 10
Ditt navn: Egil
*** SPILL 1 ***
Hva er 7*7? 49
Ja, det er riktig!
Hva er 1*2? 3
Nei, riktig svar er 2
Hva er 6*9? |
```