PROGRAMMERING I MATEMATIK MED PYTHON

ÅRSKURS 7-9 OCH GYMNASIET

Ulrihca Malmberg

Dagens session

- Programmeringsmiljöer – installerade och online
- Matematikuppgifter med fokus på lärande i matematik och programmering (enklare syntax)

- Lite mer text i PPT som publiceras, bl.a.
  - Installation och användning programmeringsmiljöer, enskilt och för klass
  - Programmeringsprocessen
  - En del kommandon
- All kod tillhandahålls i komprimerad fil

Centralt innehåll åk 7-9 vs. Gymnasiet

Åk 7-9

Sannolikhet och statistik

- Bedömningar av risker och ch utifrån datorsimuleringar och st material

Algebra

- Hur algoritmer kapas och användas vid progra g. Programmering i olika program gsmiljöer.

Prob sning

- Hur algoritmer kan skapas, testas och förbättras vid programmering för matematisk problemlösning

Fokus utformande av algoritmer

Gymnasiet

Problemlösning

- Strategier för matem problemlösning in modellering av olika situati väl med som utan digitala verk n programmering.”

Fokus strategier för problemlösning

Centralt innehåll Teknik vs. Matematik åk 7-9

MATEMATIK

Bedömningar av risker och chanser utifrån datorsimuleringar och statistiskt material

Hur algoritmer kan skapas och användas vid programmering. Programmering i olika programmeringsmiljöer.

Hur algoritmer kan skapas, testas och förbättras vid programmering för matematisk problemlösning.

TEKNIK

Tekniska lösningar som utnyttjar elektronik och hur de kan programmeras.

Teknikutvecklingsarbetets olika faser: identifiering av behov, undersökning, förslag till lösningar, konstruktion och utprövning. Hur faserna i arbetsprocessen samverkar.

Egna konstruktioner där man tillämpar styrning och reglering, bland annat med hjälp av programmering.

Textbaserade programmeringsspråk

Språk	Kommentar
MATLAB	Kraftfullt för matematiska beräkningar
Octave	Liknar MATLAB
Python	www.python.org. ”Ren” kod. kraftfullt. Relativt låg tröskel. *Gmail, Google maps, Youtube, NASA*
Scala	Ofta kompakt kod.
JavaScript och PHP	Vanligt inom webb-utveckling.
Java, C++, C#	Java och C++ - mycket vanliga. C# - liknar Java. Utvecklat av Microsoft.

Programmeringsmiljöer Python

Program installerade på egna datorn

- **Pycharm** (https://www.jetbrains.com/pycharm-edu/)
- **IDLE** (https://www.python.org/downloads/)

Webb-applikation

- **Repl.it** (https://repl.it/)
  Stöder flera språk. Har kurser och guider.
- **Codingground** (http://www.tutorialspoint.com/codingground.htm)
  Stöder flera språk.
- **Google colaboratory** (https://research.google.com/colaboratory/)

Installera på datorn
Python 3.6.2

Python 3.6.2

Ladda ner installationsfilen från https://www.python.org/downloads/.

Öppna filen och följ instruktionerna.

PyCharm Edu

Ladda ner installationsfilen från https://www.jetbrains.com/pycharm-edu/

Skriva program i IDLE

1. Se till att IDLE är igång
2. Välj File / New File i menyn
3. Skriv programmet nedan.

```
# DittNamn.py
namn = input("Vad heter du?\n")
print("Hej ", namn)
```

Vi sätter # framför kommentarer som inte är kommandon som datorn ska köra. Kommentarer är för att underlätta för den som ska förstå vad programmet gör.

Input: Programmet skriver ut texten på skärmen och lagrar informationen i variabeln "namn".

Print: Programmet skriver ut på skärmen

4. Spara filen med File / Save och ge det filnamnet DittNamn.py
5. Kör programmet med Run / Run Module. Programmet körs i skalfönstret.

Repl.it

- Inlogg: t.ex. Google-konto
- Välj Python 3 som språk
  (Vill man göra grafik väljer man språk: *Python (with turtle)* på startsidan).
- Create classroom – för att skapa och
  tillhandahålla material
- Skapa *Assignment*. Här kan du lägga in
  kod och skriva instruktioner.
  Välj *Next – Manual – Publish now (eller senare)*
- När man har skapat en assignment (välj ”manual” för koll av elevresultat”) –
  Viewable in community
- **Dela med eleverna:**
  Teacher uppe till höger. Klicka på aktuellt *classroom*. Scrolla till *Student overview* och välj *Invite more*. Kopiera länk och dela med eleverna.

Att lösa ett problem med programmering

1. Identifiera problemet som ska lösas
2. Ta fram en grundidé till lösning
3. Indata och utdata
4. Dela upp problemet i mindre delproblem
5. Bestäm ordningen på delproblemen
6. Sätt ord på varje steg i lösningen - **pseudokod**

Ofta på papper

Pseudokod

Skriv ut programmets uppgift
Input: startkapitalet (heltal)
Input: räntesats (flyttal, procent)
Input: Antal år (heltal)

Beräkna förändringsfaktorn

Använd exponentialfunktion för beräkning av slutkapital
Skriv ut slutsaldot

```
print("Värdet på ditt bankkonto efter givet antal år.")

start = int(input("Ange ditt startkapital i kronor: "))
räntesats = float(input("Ange bankens årsränta: "))
år = int(input("Ange antal år du sparar: "))

FF = 1 + räntesats/100

slut = round(start * FF**år,2)
print("Beloppet på ditt konto är ", slut, "kr")
```

Arbetsformer i klassrummet

Tinkering *(göra små förändringar för att förbättra eller rätta något)*

- Tolka/förstå färdig kod
- Modifiera färdig kod

Vidareutveckla färdig kod

Designa en lösning i pseudokod

Skriva egen kod utifrån design

Fullständig programmering

EPA

Glöm inte matematiken!

Reflektioner om gymnasiet

Kodskrivning i t.ex. Python med smidigare system som Geogebra och Excel

Styrkan med kodskrivning inte uppenbar

Hantera och bearbeta stora datamängder, t.ex. från filer, databaser, nätet…

Ordlista.py (ordlista.txt)

NU KÖR VI!

DAGENS KODER

https://repl.it/classroom/invite/VhtSOQg

Obs! Kräver att du skaffar inlogg (gratis). Välj ”teacher”.

Instruktioner hur du som lärare skapar ett klassrum, lägger in uppgifter och kod samt delar med elever finns i PPT som lägg ut.

Variabler och skriva ut

Koden

```
print("Nu kör vi")
```

Ger utskriften

```
Nu kör vi
```

Koden

```
namn = input("Skriv ditt namn: ")
print("Ditt namn är", namn)
```

Ger utskriften

```
Skriv ditt namn: Ulrihca
Ditt namn är Ulrihca
```

Koden

```
ålder = 51
print("Din ålder är", ålder)
```

Ger utskriften

```
Din ålder är 51
```

Matematiska operatorer

Inspirerad av Malmö stads programmeringskurs för lärare:
https://sites.google.com/skola.malmo.se/programmeringsfortbildning/

```
a = 8
b = 3
c = a + b                   # summa

print(c)
print(a - b)                # differens
print(a * b)                # produkt
print(a / b)                # kvot
print(a % b)                # resten vid division
print(a // b)               # heltalsdelen vid division
print(a ** b)               # värdet av a^b
print(round(a / b, 2))      # Avrundar divisionen till 2 decimaler
```

Ger utskrifterna:

11, 5, 24, 2.6666666666666665, 2, 2, 512, 2.67

Tid – enhetsomvandling

Fil: *tidsomvandling.py* (fördjupningsuppgift)

Programmering: Matematiska operatorer

Matematik: Enhetsomvandling för tid

Uppgift:

1. Hur går man från tid i timmar, minuter och sekunder till enbart sekunder? Sätt upp en generell regel.
2. Hur kan ett program se ut? Låt användaren ange antal timmar, minuter och sekunder som separata variabler.

Fördjupning:

1. Hur går man från tid i sekunder till timmar, minuter och sekunder? Sätt upp en generell regel.
2. Hur kan ett program se ut?

Operatorer för jämförelse

`a == b`	lika med
`a > b`	a större än b
`a < b`	a mindre än b
`a >= b`	a större än eller lika med b
`a <= b`	a mindre än eller lika med b
`a != b`	a skiljt från b

När man gör en jämförelse blir resultatet antingen **True** eller **False** (booleska operatorer).

Matematiska funktioner

Det finns färdiga matematiska funktioner som kan hämtas från ett Python-bibliotek och sedan användas.

Alla matematikfunktionerna importeras genom att skriva

```
from math import *
```

eller en specifik funktion:

```
from math import sqrt
```

```
print(sqrt(25))
print(pow(4,2))
```

Ger utskriften:

```
5.0
16.0
```
(Samma som 4**2)

Datatyper

I Python sätts en variabel automatiskt till en datatyp beroende på vad man skriver in.

Detta kan man dock vilja styra ibland och vid behov ändra.

Vanliga datatyper är:

int	integer - Heltal
float	float - Decimaltal
bool	boolean - kan anta värdena False / True
str	sträng - Text

Pythagoras sats

Fil: *pythagoras_1.py* (uppgift) och *pythagoras_2.py* (fördjupning)

Programmering: Matematiska operatorer, matematikbiblioteket

Matematik: Pythagoras sats, kvadratrötter, formelskrivning, koordinatsystem, punkter

Uppgift: Skriv kod som frågar efter kateternas längd och beräknar hypotenusan.

Fördjupning:

- Ange hypotenusan och en katet och beräkna andra kateten
- Ange koordinater för två punkter och beräkna avståndet mellan dessa (*pythagoras_2.py*)
- Kunna välja om man ska ange kateter eller hypotenusan som input
- Ge två svar – ett om bägge indata förutsätts vara kateter, ett om den ena är hypotenusan

Förändringsfaktor

Fil: *förändringsfaktor_1.py* (uppgift) och *förändringsfaktor_2.py* (fördjupning)

Programmering: beräkning av variabler, matematiska operatorer, villkor och loop (fördjupning)

Matematik: procent, förändringsfaktor, exponentiell förändring, formelskrivning, omvandling procent – decimaltal, ränteberäkning

Uppgift: Du sätter in 5000 kr på banken. Årsräntan är 2 %. Hur mycket pengar har du efter 3 år?
(1) Lös med valfri metod; (2) Upprepa med andra värden; (3) Se mönster; (4) Skriv kod där användaren anger relevanta värden.

Fördjupning: Utveckla koden, t.ex. avrundning, kontots värde årsvis, uttag/insättning under perioden, beräkning av tid (logaritmer)

Villkor: **if – elif - else**

```
a = 0
if a < 5:
    print("a är mindre än 5")
elif a == 5:
    print("a är 5")
else:
    print("a är större än 5")
```

Ger utskriften:

```
a är mindre än 5
```

Linjära funktioner – att hyra bil

Fil: *hyrbil.py* (fördjupning)

Programmering: Villkor, matematiska operatorer

Matematik: Linjära funktioner, formelskrivning, definitionsmängd, värdemängd

Uppgift:

(1) Hitta på en prismodell för biluthyrning som har en given startkostnad och ett km-pris; (2) Räkna några exempel; (3) Finn en generell regel som visar sambandet mellan åkta km och pris; (4) Skriv ett program till biluthyraren som snabbt räknar ut kostnaden beroende på hur långt man åker. Använd er egen funktion.

Fördjupning:

(1) Lägg in definitionsmängd och/eller värdemängd (*hyrbil.py*); (2) Olika hyrkategorier; (3) Jämförpriser mellan olika alternativ; (4) Olika prismodeller beroende på körsträcka

Formler - volymberäkning

Fil: *formel_1.py* (uppgift), *formel_2.py* (fördjupning)

Programmering: Matematiska operatorer, matematiskt bibliotek, villkor

Matematik: Formler, lösa ut variabler

Uppgift: Skriv kod som frågar efter radie och höjd och beräknar volym för en kon.

Fördjupning:

1. Användaren styr variabel som söks (formel_2.py)
2. Effektivisera formel_2.py
3. Komplettera med andra geometriska objekt i samma program.

Villkor och loopar: **while**

Instruktionerna utförs så länge villkoret är sant.

```
a = 0
while a < 5:
    print(a)
    a = a + 1
```

Ger utskriften:

```
0
1
2
3
4
```

Villkor och loopar: **for**

Variabeln *number* antar alla värden från 1 upp till 10 med 2 stegs intervall.

```
for number in range(1,10,2):
    print(number)
```

Ger utskriften

1
3
5
7
9

Att slumpa

Biblioteket *random*

```
from random import *
```

Exempel

randint(2, 5)	*slumpar heltal mellan 2 och 5*
randrange(0, 101, 2)	*slumpar jämna heltal mellan 0 och 101*
random	*slumpar decimaltal mellan 0.0 och 1.0*
sample([100, 33, 59, 71, 112, 130], 4)	*slumpar fyra av talen i listan*
choice(ruter, spader, hjärter, klöver)	*slumpar någon av färgerna*

Tärningskast – lika antal prickar

Fil: *tärning.py*

Programmering: slumpbiblioteket, villkor/loop

Matematik: experimentell sannolikhet, utfallsdiagram.

Uppgift: (1) Kasta två tärningar och registrera antal kast och där de visar lika. Lägg samman klassens resultat och beräkna *P(lika);* (2) Vad gäller generellt? Brister i metoden?; (3) Skriv kod som slår tärningar åt oss; (4) Beräkna teoretisk sannolikhet (utfallsdiagram). Jämför alla resultat

Fördjupning: Kod för liknande uppgifter, t.ex. given summa. Bygga ut så programmet kan göra många olika typer av tester.

Lagra data i en lista

Kod	Beskrivning
lista = [Kalle, Anna, Oliver, Rutger]	*Kalle, Anna, Oliver* och *Rutger* är lagrade i listan
print(lista[0])	Index är 0 och det motsvarar första elementet. Skriver ut *Kalle*.
print(lista)	skriver ut hela listan
listnamn.append(Kalle)	lägger till *Kalle* sist i listan
listnamn.count(Kalle)	räknar hur många Kalle det finns l listan
len(listnamn)	Anger antalet element i listan
listnamn.sort()	Sorterar listan i bokstavs- eller storleksordning
sum(listnamn)	Summerar elementen i listan (om de är tal)

Statistik

Fil: *statistik.py*

Programmering: lista, villkor/loop, matematiska operatorer, statistikbiblioteket

Matematik: statistik, medelvärde, median

Uppgift:

(1) Samla in minst 30 datavärden (tal), beräkna medelvärde och median; (2) Skriv kod som gör beräkningarna åt dig, lagra datavärdena i en lista. Beräkna medelvärde ”manuellt” och använd bibliotekets funktion för median.

Fördjupning:

(1) Skriv pseudokod för att beräkna medianen genom att arbeta med listan direkt (ej bibliotekets median-kommando); (2) Skriv kod för annan statistik, t.ex. typvärde, standardavvikelse, kvartiler…

Aritmetisk talföljd

Fil: *Aritmetisk_talföljd.py*

Programmering: Matematiska operatorer, villkor/loop

Matematik: aritmetisk talföljd, summa av talföljd, formelskrivning

Uppgift:
Skapa en talföljd utifrån givet starttal och differens samt antal tal. Skriv ut talföljden och summera talen. Utforska med papper, hitta ett mönster och skriv kod.

Fördjupning:

- Motsvarande för geometrisk talföljd
- Identifiera ett tal för givet index i talföljden
- Talföljd baserad på annan regel

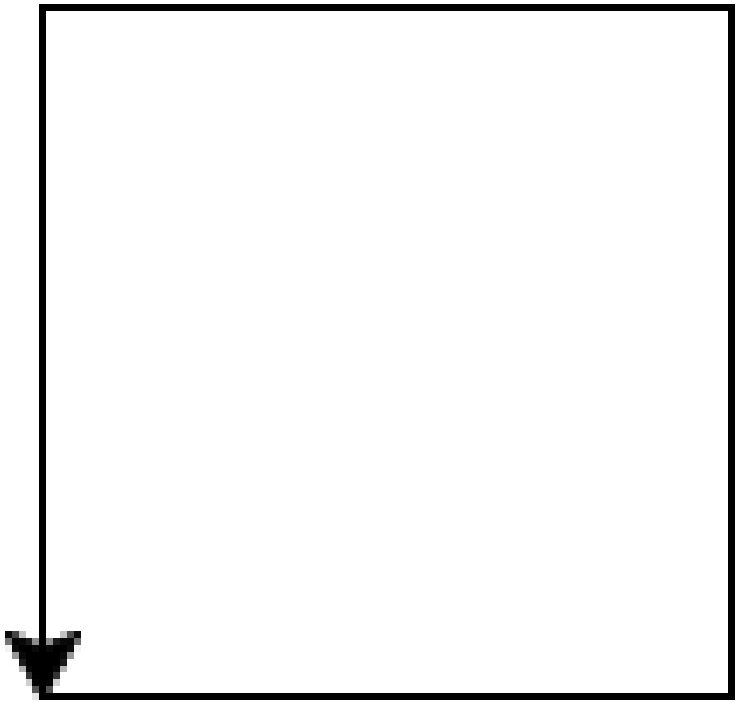
Grafik

Rita mönster, diagram, funktioner…

Ibland omständligt i standardmodul, kan behövas tilläggsprogram

VAD RITAR VI NU?

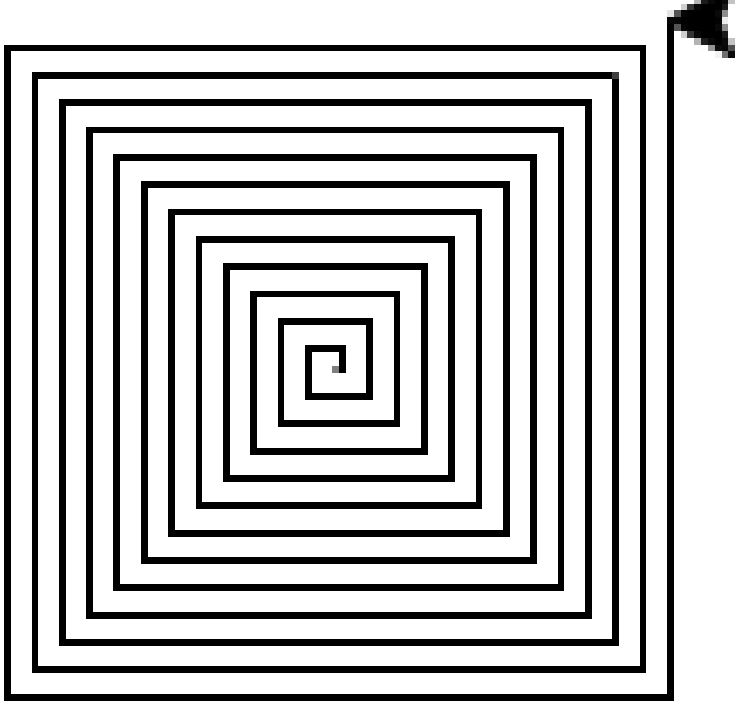
```
import turtle
t = turtle.Turtle()
t.forward(100)
t.left(90)
t.forward(100)
t.left(90)
t.forward(100)
t.left(90)
t.forward(100)
```



import	hämtar tillgång til färdig kod i ritbiblioteket turtle
t = turtle.Turtle()	Vi använder bokstaven t i stället för ritkommandot turtle.Turtle()

VAD RITAR VI NU?

```
import turtle
t = turtle.Turtle()
for x in range(1,100,2):
    t.forward(x)
    t.left(90)
```

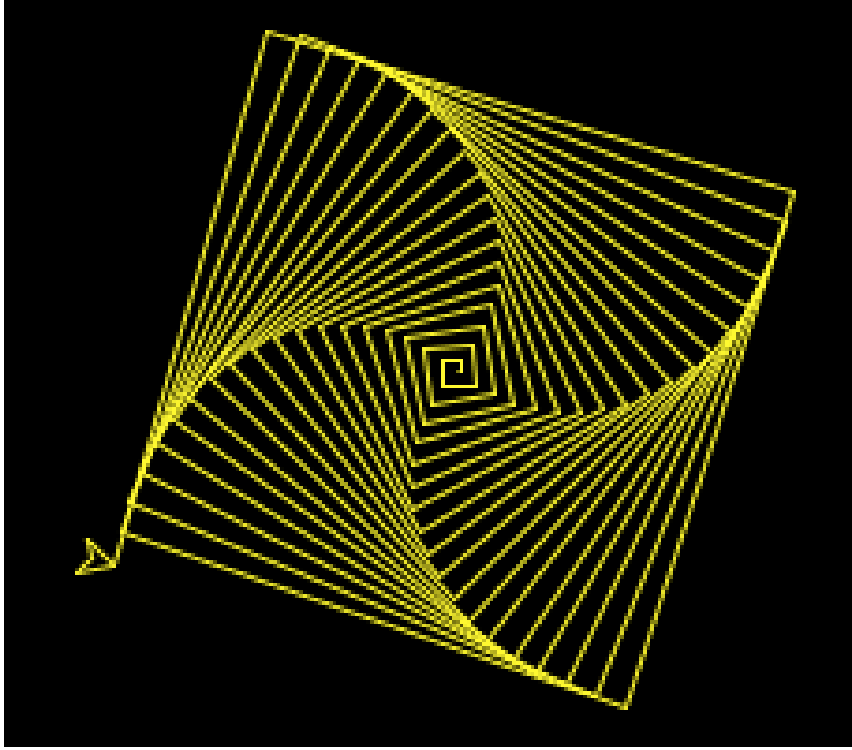


https://repl.it/@FrokenUlle/grafik1

For: Start på en loop. Alla indragna rader (tab) tillhör "for"-loopen.
Det indragna upprepas 100 gånger.
x varierar mellan 1-100 och ökar med 2 varje gång.

VAD RITAR VI NU?

```
import turtle
t = turtle.Turtle()
turtle.bgcolor('black')
t.pencolor('yellow')
for x in range(1,150,2):
    t.forward(x)
    t.left(91)
```



https://repl.it/@FrokenUlle/grafik2

VAD RITAR VI NU?

```
import turtle

colors = ['red','purple','blue','green','yellow','orange']
t = turtle.Turtle()
rita = turtle.Turtle()
rita.speed("fastest")
turtle.bgcolor('black')
for x in range(360):
    t.pencolor(colors[x % 6])
    t.width(x/100+1)
    t.forward(x)
    t.left(59)
```