

Lärarhandledning

1. Eleverna i undervisningsgruppen går årskurs 1 på Naturvetarprogrammet på gymnasiet. Till en början kan vi inte förvänta oss att de har med sig varken programmeringskunskaper eller specifikt hur man skriver text i programmeringsspråket Python.

Först så går vi igenom begreppen från det centrala innehållet, algebraiskt under en lektion där de även löser uppgifterna som finns i boken, så att de bekantar sig med matematiken. Påföljande lektion är tänkt som programmeringslektion där de får undersöka begreppen genom programmering, baserat på det centrala innehållet i kursen (se nedan).

Under kursens gång kommer lektioner med programmering en gång per kapitel (sex gånger).

Programmeringslektionens upplägg kommer att vara att eleverna, två och två, får kod som de skriver in och kör (övning 1-7), för att se vad som händer. Eleverna får sedan utgå ifrån ursprungsprogrammet för att göra ändringar enligt uppgifterna som följer i häftet. Viktigt då är att de även testar själva vad som händer när de ändrar i programmet så att de inte bara följer en mall utan är kreativa och undersökande.

När de har gått igenom övningarna så finns uppgifter på två betygsnivåer (E och C), som de skriver program till och kör.

Tiden som detta ska ta är en 1,5 h lektion. Läraren(jag) har inte någon genomgång utan eleverna ska kunna få häftet och direkt köra igång och börja programmera, där läraren är behjälplig i klassrummet.

Målet är att de ska få en start till hur program skrivs och körs i programmeringsspråket Python, samt att de ska kunna tillverka program som löser matematiska uppgifter som bygger på primtal och delbarhetsbegreppen. Samt allt eftersom i kursen fler och fler matematiska begrepp. Progressionen under kursen är att till en början introduceras ett programmeringstekniskt begrepp, så att eleverna bara testat det till att programmet blir mer avancerat.

2. Riktlinjerna från Skolverket (2018) för kursen Matematik 1c, som används i Lektion 1 finns i det centrala innehållet:

- **Taluppfattning, aritmetik och algebra.** Egenskaper hos mängden av heltal, olika talbaser samt begreppen primtal och delbarhet.
- **Problemlösning.** Strategier för matematisk problemlösning inklusive modellering av olika situationer, såväl med som utan digitala verktyg och programmering.

Referens:

Skolverket (2018). Läroplan, examensmål och gymnasiegemensamma ämnen för gymnasieskola 2018. Stockholm: Skolverket publikationer.

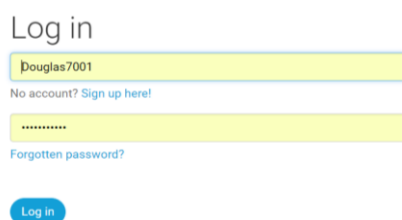
Lektion i matematik 1c i programmering (Python).

Vi ska i kursen Matematik 1c göra en lektion per kapitel. Viktigt att ni två och två arbetar ihop och diskuterar varför det inmatade blir som det blir när ni kör programmet (RUN). Våga även ändra i programmet så att ni förstår vad varje del gör för resultatet.

Innan ni kan programmera måste ni:

Skriva in www.pythonanywhere.com i webbläsaren. Ni når direkt inloggningsfönstret där ni väljer **sign up here** (se figur 1 nedan), eftersom in inte har något konto ännu.

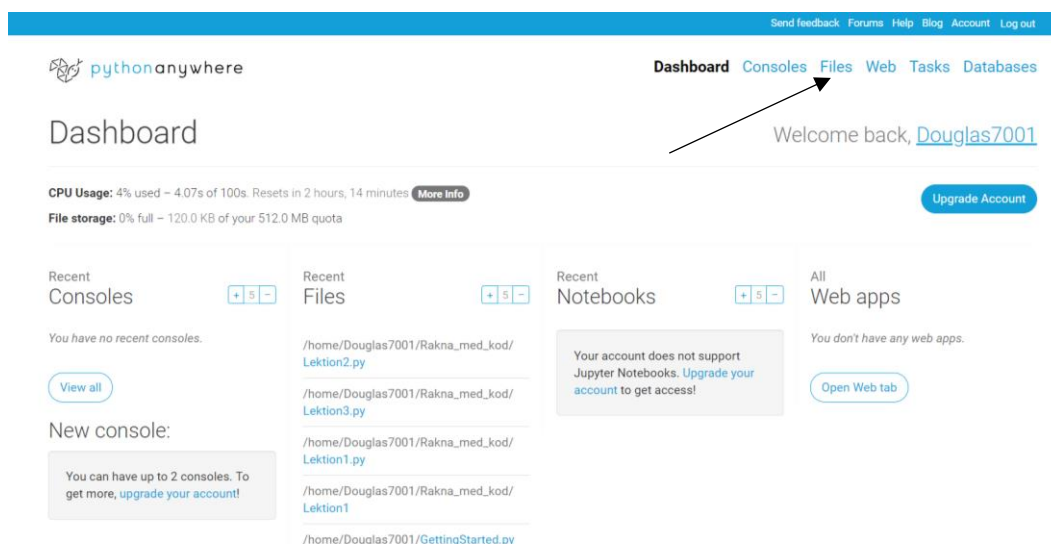
Figur 1



Ni matar in informationen som ny användare. Tänk på att välja ett användarnamn samt lösenord, som ni även skriver ner någonstans.

Starta en fil, genom att gå in på **Files** (Se figur 2 nedan) och **New file** (Se figur 3 nästa sida). Filen döper ni till: lektion1.py (Viktigt att ni väljer .py så att programmet vet att det är i programmeringsspråket PYTHON ni programmerar).

Figur 2



Figur 3



Lektion 1 Primitiva och delbarhet

Övning 1:

I utmatningsfältet (det understa fältet) skriver ni följt av Enter.

#Övning 1

$2+3$

$5-1$

$4*3$

$4**3$

Skriv i inmatningsfältet (det övre).

$a = 4$

$b = 3$

Skriv sedan $a+b$ i utmatningsfältet.

Vad innebär de olika inmatningarna?

Övning 2

2A.

Skriv in i inmatningsfältet följt av Enter

#Övning 2

$5/2$

$5//2$

$5\%2$

Diskutera vad de tre olika inmatningarna gav för resultat. Vad innebär det att skriva ett /, två // samt % mellan två tal? Testa det ni kommit fram till genom att skriva egna beräkningar och text.

Fundera på hur övning 1 och 2 går att använda för att få fram om ett tal är primtal eller sammansatt tal. Är talen 31 och 32 primtal eller sammansatt tal? Vilka beräkningar behöver göras?

Övning 3:

Skriv i utmatningsfältet

```
#Övning 4
```

```
a==3 samt a==4
```

Vad blir resultatet nu?

Vad är skillnaden mellan att skriva = och ==?

Övning 4

Skriv i inmatningsfältet

```
#Övning 5
```

```
def parity(n):
```

```
    if n % 2 == 0:
```

```
        return str(n)+" is even"
```

```
    else:
```

```
        return str(n)+" is odd"
```

Kör programmet (RUN) och skriv i utmatningsfältet parity(5) och Enter.

Diskutera vad programmet gör.

Rad 2: Vad innebär: `def parity(n)`?

Rad 3: Vad gör `if n%2 == 0` ?

Rad 4: Vad gör `return str(n)+" is even"`

Rad 5: Varför `else`?

Testa några olika tal n.

Övning 5:

Skriv i inmatningsfältet

```
#Övning 5
def delbarMed(n,p):
    if n%p==0:
        return str(n)+ " är delbar med "+ str(p)
    else:
        return str(n)+ " är ej delbar med "+str(p)
```

Vad gör programmet. Vad är indatan till funktionen, vilken beräkning görs samt vad blir utdatan?

Övning 6

Skriv i inmatningsfältet

```
#Övning 6
def delbar(m):
    for n in range (1,m):
        if n % 2 == 0 and n % 5 == 0:
            return str(n)
```

Rad 2: Vad innebär *def delbar(m)*:

Rad 3 Vad innebär det att skriva *for n in range (1,m)*: Vad gör programmet? Hur hänger m:et i *delbar(m)* med m:et i *range(1,m)* ihop?

Tips: *range(a,b,c)*. *a* är startvärdet, *b* är stoppvärdet (minus ett) och *c* är steglängd.

Rad 4: Vad innebär *n % 2 == 0 and n % 5 == 0*. Vad görs för beräkning för varje n? Byt ut 2 och 5 mot egna tal och se vad som händer när ni kör programmet.

Uppgift: Vad händer om ni byter ut *and* mot *or*? Vad blir skillnaden?

Uppgifter som följer på övning 1-7, där alla delar blandas ihop:

E- nivå

1. Skriv ett program som beräknar:
 - a) 13 delat med 3
 - b) Heltalsdivisionen av 13 och 3.
2. Skriv ett program som beräknar resten när man
 - a) Dividerar 13 med 2
 - b) Dividerar 19 med 5
 - c) Dividerar 181 med 30
3. Kör programmet:

```
for n in range(1,181):  
    if n%3 == 0 and n % 5 ==0:  
        return str(n)
```

 - a) Vad gör programmet?
 - b) Ändra programmet så att det i stället undersöker tal som är delbara med 2 eller 5.
4. Skriv om programmet i uppgiften här ovanför så att det skriver ut alla tal mellan 1 och 101 som är delbara med
 - a) 2 och 5
 - b) 2 eller 5 eller både 2 och 5
5. Para ihop varje program med rätt resultat (A och B) med (1 och 2).
 - A. for x in range (1,16):

```
    if p % 4 == 0 and p % 6 == 0:  
        return str(p)
```
 - B. for q in range (1,16):

```
    if q % 4 == 0 or q % 6 == 0:  
        return str(q)
```
 1. 4
 2. 12
 - 6
 - 8
 - 12

6. Skriv ett program som kontrollerar om ett inmatat heltal från tangentbordet (input) är:
 - a) Jämnt
 - b) Udda
 - c) Delbart med 5

C-nivå

7. Skriv ett program som tar reda på om ett inmatat heltal är:
 - a) Jämnt och större än 10
 - b) Udda och mindre än 100
 - c) Negativt och delbart med 3
 - d) Negativt och delbart med 2 och 3
 - e) Varken delbart med 2 eller med 3

(Tips: ej lika med skrivs != i Python)
8. Skriv ett program som anger alla tal mellan 50 och 150 som är delbara med 5 eller 6, men inte med både 5 och 6.
9. Skriv ett program som letar upp nästa primtal efter 100.
10. Skriv ett program som frågar efter två tal, skriver ut vilket som är störst samt skriver ut medelvärdet.
11. Skriv ett program som räknar ut hur många:
 - a) Minuter och sekunder som tex 100 sekunder är.
 - b) Timmar, minuter och sekunder som tex 12345 sekunder är.
12. Skriv ett program som skriver ut alla primtal mellan 1 och 100.
13. Skriv ett program som skriver ut de första 25 talen som är delbara med 7.
14. Skriv ett program som beräknar vilka äkta delare ett tal består av tex 20 har 2,4,5 och 10.

15. Skriv ett program som listar primtalen mellan 1000 och 1100.

