

# Gymnasiearbete på Vetenskapens Hus

Träff 3: Introduktion till programmering

- Första halvan:
  - En (snabb) introduktion till python/programmering
  - Hur en löser SIR-modellen
  - Visualisering med riktig data
- Andra halvan:
  - Eget arbete med exempelvis
    - Grundläggande programmering
    - SIR-modellen (testa olika parametrar)
    - Visualisering av olika datan
  - Tid för frågor!!

# Introduktion till programmering

- Skriva och köra vårt första program
- Variabler och tilldelning
- Typer
- Funktioner
- Att använda bibliotek
- Differentialekvationer
- Visualisering med riktig data

Allt material + kodexempel för alla olika koncept finns här:

<https://github.com/TypAnna/VHGY20>

# Vårt första program

- HelloWorld.py - ett program som skriver ut "Hello World!"

```
print("Hello World!")
```

# Tilldelning

Vad händer egentligen när vi skriver `x = 10`?

1. Ett objekt med typen `int` skapas
2. Detta objekt sätts till värdet `10`
3. Variabeln `x` binds till detta objekt

Kan tyckas självklart men det är viktigt att kunna vad som händer i bakgrunden.

# Typer

- Heltal (integers)
- Flytpunkter (floating points): decimaltal
  - 0.1
  - 7.000000238471821
- Strängar: representerar text, en följd av karaktärer och markeras med citattecken
  - "KatjaKaj"
- Listor: är en lista med objekt. Markeras med hak-klamrar
  - ["katt", "tiger", "lo"]
- Sanningsvärden (booleans): är antingen true eller false. Finns enbart två sanningsvärden

# Typer

Finns mängder med typer som alla har olika egenskaper och det finns en mängd funktioner som kan appliceras på dessa!

- Heltal och flytpunkter kan exempelvis summeras - precis som i vanlig matte
  - $1 + 0.3 = 1.3$
- Strängar kan också läggas ihop, men då sätts strängarna ihop
  - `"1" + "0.3" = "10.3"`
  - Vi kan också ta reda på längden av en sträng genom den inbyggda funktionen `len()`.
- Vi kan hämta element från listor genom
  - `ettElement = enLista[i]`, där `i` indikerar vilket element i listan vi vill hämta. Om `i=0` hämtar vi det första elementet.

# Funktioner

Funktioner är *extremt* viktiga då det är dom som står för allt grovjobb i ett program.

Funktioner tillåter oss att återanvända kod och göra koden mer lättförståelig - vilket leder till bättre kod med mindre fel!

Det finns *massor* med inbyggda funktioner - men vi kan också definiera egna funktioner, vilket är superduperbra!!



# Funktioner

Vi definierar en funktion med nyckelordet (detta är ord som har specifik betydelse) 'def' följt av namnet av funktionen, tillsammans med eventuella parametrar/argument.

```
def helloYou(namn):  
    print("Hello", namn)
```

För att anropa funktionen skriver vi: `helloYou("Kaj")`

# Funktioner

Uppgift: Skriv en funktion som beräknar hur gammal en person är utifrån vilket år hen är född och skriv ut detta.

# Bibliotek

Det finns massor av *bibliotek* som innehåller funktioner som andra har skrivit som vi kan använda oss av! Vi kan använda dessa för att

- Rita kurvor/grafer
- Träna och testa modeller (maskininlärning/AI)
- Hantera stora datamängder
- Lösa differentialekvationer
- Hantera datum och tid
- Och massa, massa, massa, massa, massa, massa mer!!!!!!

För att ladda in ett bibliotek skriver vi `import`  
`namnPåBibliotek.`

# Differentialekvationer

- Låt oss använda oss utav allt vi gått igenom för att lösa differentialekvationer!
- Vi behöver:
  - [Matplotlib](#): för att rita grafer
  - [Numpy](#): för att hantera stora listor (något förenklat)
  - [Scipy](#): för att lösa differentialekvationer

# Visualisering med riktig data

- [Här](#) finns massor med riktig data.
- Kom ihåg att kolla så att den datan du är intresserad av finns!
  - Exempelvis så innehåller inte datasetet hur många som tillfrisknat varje dag, och olika länder rapporterar olika data.

# Nästa steg

- Eget arbete med exempelvis
  - Frågeställningen
  - Grundläggande programmering
  - SIR-modellen (testa olika parametrar)
  - Visualisering av olika datan
- Tid för frågor!!