


Programmering i Python

27 oktober	Grunder i Python
10 november	Slingor, moduler
17 november	Funktioner, läsa data
1 december	Läsa data forts., dict + objekt + klasser
 8 december	Repetition + frågor + uppgifter

Idag

- Repetition
- Genomgång uppgift från förra gången
- Klasser i olika filer
- Uppgifter eller eget “projekt”

Dictionaries repetition

Dictionaries är en datastruktur i python där man lagrar “nycklar” och värden. Skillnaden mot en lista är att nycklarna inte behöver vara heltal, utan kan också vara strängar.

Dictionaries skrivs i curly brackets { } och kan se ut så här:

```
land = {  
    "namn": "Sverige",  
    "befolkning": 1000000,  
    "språk": "svenska"  
}  
print(land)
```

Dictionaries - hämta eller ändra värde i en dictionary

Inne i dictionary:n så har vi flera par av nycklar och värden (ex "namn": "Sverige"). För att nå värdena så använder vi oss av nycklarna. Ex:

```
print(land["befolkning"])
```

OBS! Om vi försöker hämta ett värde med en nyckel som inte finns, så blir det problem. Man kan kontrollera att en nyckel existerar med:

```
if "befolkning" in land:  
    # Gör det du vill
```

I listor ändrar vi värdet på en viss index genom att skriva lista[index] = värde

I dictionaries är det likadant, men med nyckeln istället för ett index.

```
land["namn"] = "Norge"  
land["befolkning"] = 2000000  
print(land)
```

Dictionaries - Lägga till/radera en nyckel och värde

Lägga till

```
land["latitud"] = 0.2832938
```

```
land["longitud"] = 0.38403
```

Ta bort

```
del dictionary[key]
```

Ex.

```
del land["longitud"]
```

```
del land["latitud"]
```

Hämta data från internet (om ni vill)

Först måste vi installera python-paketet requests:

1. Öppna CMD

2. `python -m pip install --user requests` (MacOS: `python3 -m pip install requests`)

Python-scriptet finns i förra veckans mapp

Klasser repetition

```
class Land:  
    name = "Sverige"  
    befolkning=10000000
```

```
sverige = Land() # Vi skapar ett objekt av klassen Land  
print(sverige.befolkning)
```

Vad är klasser?

Alla objekt tillhör minst en klass. Kan tänka som i verkliga världen:

Min bil tillhör klassen Bil, jag tillhör klassen Människa

En klass innehåller vissa parametrar/egenskaper och metoder.

T.ex. En bil har en egenskap som är registreringsnummer, ålder, etc.

En människa har egenskaper som namn, ålder, längd etc.

Man kan också se klassen som en ritning för objekt, medan objektet är implementationen (det konkreta)

Klasser - repetition

```
class Land:
    def __init__(self, namn, befolkning):
        self.namn = namn
        self.befolkning = befolkning

    def printBefolkning(self):
        print(self.befolkning)
    def printNamn(self):
        print(self.namn)
```

```
sverige = Land("Sverige", 10000000)
```

```
norge = Land("Norge", 3000000)
```

```
sverige.printNamn()
```

```
norge.printBefolkning()
```

`__init__` är en method som
auomatiskt kallas på när vi skapar
ett nytt objekt av den klassen.

←

←

Genomgång Hämta väderdata

1. Ändra programmet där vi fick väderinformation så att användaren får välja vilken stad den får väderinformation om

Tips: trafiklab.se

Innehåller massa API:er för trafik i Sverige och inkluderar SL reseplanerare-API

Steg att gå vidare

- **TKInter** - gör grafiska interfaces istället för bara textbaserade program
 - `import tkinter as tk`
- **Pygame**
 - Gör 2d-Spel
- Skaffa en mer kraftfull IDE bättre än **IDLE**
 - Ex. **Pycharm**
- Skriv dina Python-program i en textredigerare och kör programmen i **Terminalen** (**CMD** eller **PowerShell** på Windows)
- Kolla upp roliga API:er som man kan ha kul med, eller göra praktiska saker med
- Kolla upp praktiska paket, tex för att skapa csv-filer eller liknande

Uppgifter

1. Gör en funktion som tar in ett namn (sträng) och returnerar strängen i rövarspråket (hej => hohejoj, erik => erorikok) och testa den med några strängar
2. Skapa en dictionary som innehåller några engelska ord som nyckel och deras svenska översättningar som värden, skapa sedan en funktion som tar in ett engelskt ord och returnerar det svenska ordet. Om översättningen inte finns så returnera *"Sorry, not found"*
3. Utveckla programmet som sade vädret i en specifik stad till att skriva ut mer än bara en beskrivning av vädret (t.ex. temperatur, solens upp- och nedgång)
4. Utveckla Car-classen, så att den har en egenskap "speed" som kan ökas med metoden "increase_speed" och minskas med metoden "decrease_speed". Ökas = adderas
5. Skapa ett program som ber användaren om en hållplats inom SL och printar ut information om alla nästkommande avgångar (buss/tunnelbana/pendel). Använd trafiklab.ses API:er [SL Platsuppslag](#) och [SL Realtidsinformation 4](#). Fråga mig om inloggning eller skapa en själv.
6. Kom på ett eget projekt eller fråga mig! Exempel: Twitterbot, tre i rad spel.