

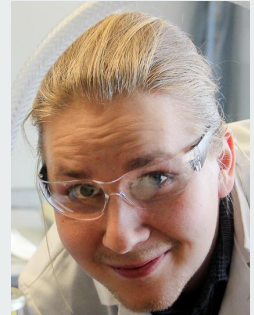


Databearbetning

Steget innan datavetenskap

Lektion 2 - Python övningar, OOP, Bibliotek

Dennis Biström
bistromd@arcada.fi



Kursinnehåll



Kompetensmål

Målsättningen med kursen är att bekanta er med centrala begrepp inom datavetenskap och lära sig behandla olika typer av data på praktisk nivå.

Kurslitteratur

Största delen av materialet är i elektronisk form
Ni klarar er långt på pandas och pythons dokumentation, förutsatt att ni *orkar läsa...*

<http://pandas.pydata.org/>

<https://docs.python.org/3/reference/index.html>

“Some of us just like to read” - Lady Gaga
Python for Data Analysis, Wes McKinney (O'Reilly Media)

Jag ska givetvis försöka ge er effektivare och mer intressanta resurser t.ex i videotutorialformat.

Kolla gärna **systerkursen** vid Harvard: [CS109 Data Science](#)

Föreläsningssinnehåll

Datahantering med Python och Pandas
Web scraping och data APIs
Visualisering, Tidsserier och signalbehandling
Bilder som data

Läranderesultat

Efter avklarad kurs förväntas den studerande vara förmögen att

- Hämta, formatera, och visualisera data med hjälp av Python, Pandas och matplotlib
- Behandla olika sorters data (ljud, bild, csv, json...)
- Göra web scraping och använda data APIs (GET & Parsing)
- Känna till signalbehandling (t.ex. ljudfil med excel, audacity)
- Analysera bilddata och grafer (info i alpha, visualisering)

Praktiskt om kursen - Kursplan och förväntningar



Elevens ansvar

Kursen består av 7 lektioner (8). Jag använder inte lektionerna bara på att stå och föreläsa, utan vi använder en stor del av lektionstiden på uppgifter.

Föreläsarna på Lynda är världsbäst på vad de gör, och varje kurs har ett planerings & analysteam bakom sig, något jag inte har. Använd alltså Lynda!

Utöver lynda så ska kursen träna er i informationssökning, ert viktigaste verktyg som programmerare.

Jag menar inte att ni går kursen på egen hand, jag ger resurser enligt förmåga när jag märker att ni "kör fast".

Kursplan

Ja presenterar en deluppgift (grovt baserad på teorin)

Vi gör en del av uppgiften i klassen

Jag hjälper er komma vidare med uppgiften

Resten av uppgiften slutför ni till nästa gång

Tidsplanering

Från eleven väntas en insats á 133h. De här timmarna är fördelade på följande sätt:

Föreläsningar	6/20 h
Praktiska övningar	2/13 h
Projekt- och produktionsarbete	0/60 h
Självstudier	0/40 h

Praktiskt om kursen - Bedömning och närvaro



Bedömning

Vitsordet bestäms på basis av era lösningar på kursuppgifterna.

Maxpoäng 10 per uppg

Uppgifterna ska lämnas in inom utsatt tid för att man skall kunna få fulla poäng för dem. 25% avdras för sen inlämning.

Varje uppg bygger på kunskap från den föregående, så försök att inte falla av kälken.

Närvaro

Jag använder mig av en närvarolista.

De som inte har deltagit på nån av de två första föreläsningarna blir borttagna från ASTA

För mig är det viktigt att ni deltar på lektionerna så ni inte faller efter med uppgifterna.

Ifall ni närvarar på mindre än 50 % av kursen förlorar ni mycket av er "förhandlings- och klagomålsrätt" då det kommer till er kursprestation och ert vitsord.

Intensiv 4 veckors kurs! - 8 lektioner, 4 uppg



Upplägg

Föreläsningar med exempel - Var på plats, följ med!

Videoföreläsningar (60–120 min) att se på hemma

Veckouppgifter med deadline **varje** vecka.

Inget kodtilfälle! Använd F369 och fråga kaveri?

Kursverktyg

Python

Pandas (Python Data Analysis Library)

Jupyter Notebook

Installering: Anaconda (Linux / Mac / Windows)

<https://www.continuum.io/anaconda-overview>

Bedömning

Vitsordet bestäms på basis av era lösningar på kursuppgifterna. Maxpoäng 110p

Varje uppg är värd 20p. 3 förhör 10p, 3 läxor 10p

Bonus upp till 10p för smarta lösningar elr tilläggsfunktioner

5p avdrag per förseningsvecka

Närvaro

Jag använder mig av en närvarolista.

De som inte har deltagit på nån av de två första föreläsningarna blir borttagna från ASTA

<70% närvaro => begränsad klagomålsrätt

Upplägg - 8 lektioner, 4 inlämningsuppg



Lektion 1 - Kursinfo, verktyg & resurser, Intro till Databearbetning. My first python app	Läxa 1 ut
Lektion 2 - Python teori & quirks, My second and third app. Läxa 1 hjälp?	Förhör 1 ut, Läxa 2 ut
Lektion 3 - Listor, Numpy & Pandas basics, Uppg 1 start	Förhör 1 in, Uppg 1 ut
Lektion 4 - Datasets och webscraping, BeautifulSoup, Pandas på DataCamp	Läxa 3 ut, Förhör 2 ut
Lektion 5 - Visualisering, Numpy och Matplotlib Övning	Uppg 1 in, Uppg 2 ut
Lektion 6 - Förhör 2 (Numpy, Matplotlib & Pandas), Ljud och Bilder som data	Förhör 2 in, Uppg 2 in, Uppg 3 ut
Lektion 7 - Inlämningsuppg 3 fortsättning, Övning med Bilder och Signaler	Uppg 3 in, Uppg 4 ut
Lektion 8 - Inlämningsuppg 4. Kodande & Feedback, Julglögg?	Uppg 4 in

Har ni gjort läxan? - Det blir förhör!



Kolla igenom resurserna på itslearning & slides

Visst har ni fortsatt på learnpython.org och gått genom basics fram till functions?

Lär er om datatypes i python med sokratica

Kolla på båda lynda tutorials upplägg.

Hur kändes svårighetsgraden? Kommentarer?

Idag - Bibliotek, Upppg 1 och Läxa 1 forts.



Python recap från senast och fortsättning med encapsulation moduler och klasser

Slutför Läxa 1

- Sten Sax Påse
- Mind Reader
- Optimus Prime

Bibliotek - Vilka använder vi, varför och vilka är bra att känna till

Vad är datavetenskap? - Senast

Datavetenskap handlar om att hitta ny kunskap från data via beräkningsmodeller, statistik och visualisering

Varför datavetenskap? [Slides](#)

[Nate Silver](#)

Det fanns tidigare en sida som hette isnatesilverawitch.com

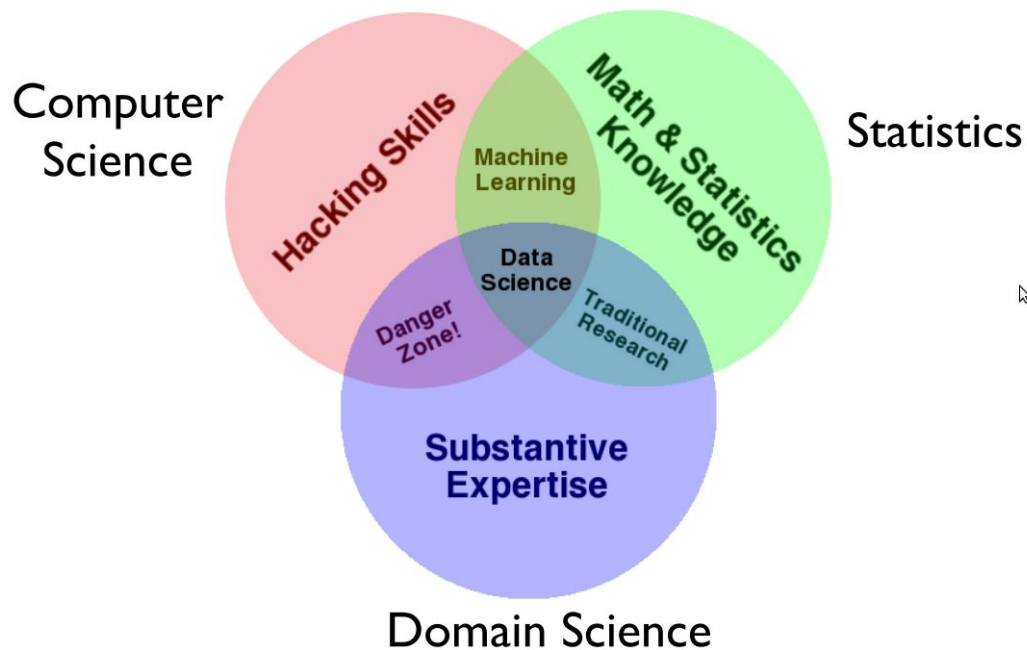
2008 lyckades han förutspå röstresultatet i 49 av 50 stater. På sajten stod det "no" på med förklaringar över hur Nate hade kommit till sin prediction mm.

2012 lyckades han förutspå röstresultatet i alla 50 stater. Efter det stod det "While we cannot say yes or no with any certainty, Nate Silver might, in fact, be a witch."



Vad är datavetenskap? - Senast

A data scientist is someone who knows more statistics than a computer scientist and more computer science than a statistician



OOP - Encapsulation refresher

En **klass** innehåller instruktioner över hur man skapar ett **objekt**

```
class Car:
    ''' Instruktioner hur man använder metoden '''
    def __init__(self, color, prod_year):
        self.year = prod_year
        self.color = color

    def age(self):
        ''' Räkna ut åldern på bilen '''
        age = (2018-self.year)
        return(age)
```

```
gamla_bettan = Car("red", "1956")
print( gamla_bettan.age() )
```

Docstring!

init metod aka konstruktör

Vi sparar värden i objektet

Märk input values vs field name

Vi lägger till funktionalitet

Skapa instans av bil

Använd klassens funktion = **METOD**

Python in one slide? - Senast

Python quirks - Indentation styr koden, försiktigt med mellanslag! Kolontecken efter if och else, *and or not*

Python - GPP, bygga webbsidor, analysera data, koda verktyg # Kommentar, även `""" Multiline comment """`

Variabler - behöver endast ett namn, tolken känner igen typen `"Sträng" + str(int) + "."`

Strings - "Text" eller 'strängar' **Escape chars** med \ för att skriva t.ex citattecken bland strängar.

Numror - Decimaltecken . | j för komplexa tal | int -> float -> complex. Tolk konv till bredare innan aritmetik.

Aritmetik - % modulo returnerar resten, // returnerar kvoten, ** fungerar som exponent. *Se upp!* $2^{1/2} = ?$

Booleans - = för tilldelning (assignment), == för utvärdering. Efter `str(True)` går variabeln inte att använda i logik!

If elif else - `raw_input("Mata in en sträng")`. (Error handling) med `try: except Error: + if else` för input validation

Interaktiv hjälp:

dir() - Se vilka moduler, objekt, klasser och metoder ni har.

help(someObject.someMethod()) - Få tilläggsinformation om objekt, metod eller funktion

someObject.someMethod? - Visa docstring

jupyter-notebook - tryck tab 1-4 gånger för att utöka information om det ni håller på att skriva just nu

Python forts. - Moduler och Klasser

Moduler - En modul innehåller python **Objekt**. Exempel på en moduler `__builtins__` eller `math`

Objekt i moduler kan innehålla **Klasser**. Många fungerar även som funktioner ex: `datetime.time(6,30)`

Objekt kan innehålla funktioner, som ofta tar emot **parametrar** ex: `math.cos(90)`

En metod är en funktion, men en funktion behöver inte vara en metod.

Metod ~ funktion, men vi kallar funktioner av en instans för en metod ex: `myTimeVariable.isoformat()`

Klasser innehåller instruktioner om hur man skapar objekt (med funktioner och data)

Data inuti objekt sparas i **fält**

Exempel:

`gamla_bettan` är en instans av klassen `bil`

`gamla_bettan.color`

`gamla_bettan.accel(10)`

`~/bistromd | $` 82 km/h

`#klassen bil` innehåller instruktioner över hur man skapar bilar

`#Klassen bil` innehåller även data som färg och märke

`#Klassen bil` innehåller även metoder, som tar emot parametrar

`#Returvärde för metoden accel()` kunde vara hastigheten

Läxa 1 (två av tre) - Vilken siffra jag tänker på?



Gissa numret! Det här är en övning i programlogik & while loopen

1. Generera en siffra mellan 0 och 10
2. Be spelaren gissa vilket nummer du tänker på
3. Bygg logik som returnerar "För högt" eller "För lågt"
4. När spelaren gissar rätt siffra, berätta för den hur många gissningar det tog
5. Inkludera en undantagsregel som "stänger" spelet
6. Bonus: Fråga ifall de vill spela igen?

Utmaningar:

1. Kortaste möjliga koden
2. State machine?
3. Undantaget
4. Bonus: Hur implementerar ni spela igen?

OOP - Message passing refresher

Objekt har ofta metoder()

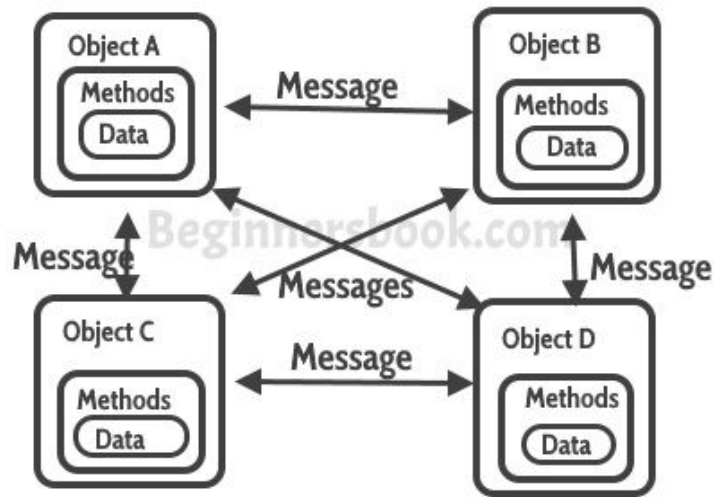
Istället för att skriva all kod efter varann som i en uppsats, så lär vi oss att dela upp koden i olika metoder/funktioner.

Metoder används för att kapsla in beteende. När den är inkapslad kan vi enkelt återanvända samma beteende.

Att dela upp funktionalitet i metoder. *Encapsulation*

Metoder kommunicerar med varandra genom parametrar och returvärden.

```
def multiply(a,b):  
    return (a*b)  
  
}
```



Läxa 1 (tre av tre) - Encapsulation med primittal



Skapa ett program som kollar ifall det inmatade numret är ett primittal

1. Fråga efter en siffra
2. Kolla om siffran är ett primittal
3. Svara med ett snyggt svar "Siffran X är/är inte ett primittal"

Utmaningar:

1. Logiken
2. Utskrift
3. Specialfall

Idag - Bibliotek, Upppg 1 och Läxa 1 forts.



Python recap från senast och fortsättning med encapsulation moduler och klasser

Slutför Läxa 1

- Sten Sax Påse
- Mind Reader
- Optimus Prime

Bibliotek - Vilka använder vi, varför och vilka är bra att känna till

Bibliotek - [Intro to data science Numpy, MatPlot & Panda](#)



1. **NumPy** stands for Numerical Python. The most powerful feature of NumPy is n-dimensional array. This library also contains basic linear algebra functions, Fourier transforms, advanced random number capabilities and tools for integration with other low level languages like Fortran, C and C++
2. **Matplotlib** for plotting vast variety of graphs, starting from histograms to line plots to heat plots.. You can use Pylab feature in ipython notebook (ipython notebook –pylab = inline) to use these plotting features inline. If you ignore the inline option, then pylab converts ipython environment to an environment, very similar to Matlab. You can also use Latex commands to add math to your plot.
3. **Pandas** for structured data operations and manipulations. It is extensively used for data munging and preparation. Pandas were added relatively recently to Python and have been instrumental in boosting Python's usage in data scientist community.
4. **Scrapy** for web crawling. It is a very useful framework for getting specific patterns of data. It has the capability to start at a website home url and then dig through web-pages within the website to gather information.
5. **Requests** for accessing the web. It works similar to the the standard python library urllib2 but is much easier to code. You will find subtle differences with urllib2 but for beginners, Requests might be more convenient.
6. **BeautifulSoup** is a Python library for pulling data out of HTML and XML files. It works with your favorite parser to provide idiomatic ways of navigating, searching, and modifying the parse tree. It commonly saves programmers hours or days of work.

Bibliotek - [Intro to data science Numpy, MatPlot & Panda](#)



6. **SciPy** stands for Scientific Python. SciPy is built on NumPy. It is one of the most useful library for variety of high level science and engineering modules like discrete Fourier transform, Linear Algebra, Optimization and Sparse matrices.
7. **Scikit Learn** for machine learning. Built on NumPy, SciPy and matplotlib, this library contains a lot of efficient tools for machine learning and statistical modeling including classification, regression, clustering and dimensionality reduction.
8. **Statsmodels** for statistical modeling. Statsmodels is a Python module that allows users to explore data, estimate statistical models, and perform statistical tests. An extensive list of descriptive statistics, statistical tests, plotting functions, and result statistics are available for different types of data and each estimator.
9. **Seaborn** for statistical data visualization. Seaborn is a library for making attractive and informative statistical graphics in Python. It is based on matplotlib. Seaborn aims to make visualization a central part of exploring and understanding data.
10. **Bokeh** for creating interactive plots, dashboards and data applications on modern web-browsers. It empowers the user to generate elegant and concise graphics in the style of D3.js. Moreover, it has the capability of high-performance interactivity over very large or streaming datasets.
11. **Blaze** for extending the capability of Numpy and Pandas to distributed and streaming datasets. It can be used to access data from a multitude of sources including Bcolz, MongoDB, SQLAlchemy, Apache Spark, PyTables, etc. Together with Bokeh, Blaze can act as a very powerful tool for creating effective visualizations and dashboards on huge chunks of data.

Bibliotek - [Intro to data science Numpy, MatPlot & Panda](#)




12. **SymPy** for symbolic computation. It has wide-ranging capabilities from basic symbolic arithmetic to calculus, algebra, discrete mathematics and quantum physics. Another useful feature is the capability of formatting the result of the computations as LaTeX code.

Additional libraries, you might need:

- **os** and **sys** for Operating system and file operations
- **networkx** and **igraph** for graph based data manipulations
- **regular expressions** for finding patterns in text data
- **BeautifulSoup** for scraping web. It is inferior to Scrapy as it will extract information from just a single webpage in a run.

Brush up ur skills - Med några bra videoresurser



1. Kolla [Socratica tutorialen](#) videorna 1-17
2. [Derek Banas "Learn Python in one video"](#)
3. [Intro to python for data science - DataCamp](#)
4. [CH1: Writing your own functions - DataCamp](#)
5. Gör sedan excersizes.ipynb på IL
6. Gör förhöret

Sök hjälp bland resurserna om du kör fast.

Slutför Läxa 1 - Lämna sen in på itslearning



Vi började med ett Sten Sax Påse spel på Lektion 1

- Spelet var en övning med if-else, input och lite standardsyntax.

Vi fortsatt på Lektion 2 med Mind reader spelet

- Spelet skulle introducera lite conditionals, en programloop och moduler.

På Lektion 2 skapade vi också appen Optimus Prime

- En övning i encapsulation, lite aritmetik och konkatenering.

Fulla poäng:

Fick du programloopen implementerad så spelet frågar ifall man vill spela igen? (while)

Visst använder du inkapsling med minst två funktioner, en för logiken och en för inmatningen?

Bra, då tar du säkert även emot parametrar så andra i princip kunde använda sig av funktionen?

Vilken typ är parametrarna, och hur ska andra devs veta det? Visst har du en [docstring](#)?

Visst använder du sys biblioteket för att avsluta programmet och du läst dig in på "how to exit python script cleanly"? ;)

Lämna in spelet som en .ipynb notebook på itslearning för upp till 10p!

Förhöret är öppet - Gör det innan måndag!




Förhöret

1. Problem med itslearning? Rättningslogik? "Fråga av fel typ?"
2. Hur kändes nivån? Nån fråga mycket svårare än andra?
3. Hur kändes innehållet? Var det relevanta frågor om Python?

Inlämningsuppg 1 - 100k filmratings och ratearnas demografi

1. Läs in användardata, ratingdata och filmdata
2. Filterövning: Visa endast användare som är bibliotekarier och män
3. Kombinationsövning: Medelrating per film, vilka filmer har högst rating?
4. Kombinationsövning2: Vilka filmer har högst rating enligt kön/arbete?

Lynda och resurser - Kolla även itslearning!

- 
- Cheat sheet: [Anaconda Cheat Sheet - Getting Started - PDF](#)
[Pandas Cheat Sheet - PDF](#)
- Manual/Docs [Conda package manger - Docs](#)
[Pandas - QuickStart & Cookbook](#)
- Tutorials (text) [Anaconda Getting Started - User Guide](#)
[Python - Intro till avancerat - Övningar och förklaringar](#)
[Pandas tutorial - PythonSpot](#)
[Intro to data science Numpy, MatPlot & Panda](#)
[\(Pandas - How do pivot tables work - ExcelCampus\)](#)
- Tutorials (video) [Socratica python tutorial - Youtube](#)
[Derek Banas - "Learn Python in one video"](#)

Lynda och resurser2 - Kolla även itslearning!



Interaktiva:

[Intro to **python** for data science - Gratiskurs - DataCamp](#)

[Intro to python for data science - Ch4 - **Numpy** \(DataCamp\)](#)

[Intermediate python for data science - Ch1 - **MatPlotLib** \(DataCamp\)](#)

Lynda:

[6h nybörjarkurs **Python** för datavetenskap med Lillian Pierson - Lynda](#)

[2h intermediate - **Numpy** Data Science Essentials - Charles Kelly](#)

[Intermediate - Ch3: **Numpy**, Ch4: **Pandas**, Ch 9: **matplotlib** - Miki Tebaka](#)

[2h intermediate kurs i **Pandas** med Jonathan Fernandes - Lynda](#)

[2h intermediate **Pandas** för Datavetenskap med Charles Kelly - Lynda](#)

[Big Data Analysis in python using **Numpy** and **Pandas** - Michele Vallisneri](#)

Lynda och resurser3 - Kolla även itslearning!



Interaktiva roligheter

[The Python Challenge](#)

[Roliga övningar i logisk ordning - Practice Python](#)

[How to think like a Computer Scientist](#)

[CodeSignal - Interaktiva utmaningar, badges, points etc.](#)

[Reddit daily programmer challenges](#)

Vill du vinna 1 miljon \$

[7h gratis tutorial på Kaggle - Känner ni till kaggle?](#)

Python 2 vs Python 3 trubbel, kolla [här](#)

Om Verktygen



Anaconda - En python platform med verktyg för datavetenskap och maskininlärning

[Getting started](#)

Anaconda har 6milj användare och är i sig självt skriven i python ;)

Conda - OS, Xplatform paket OCH omgivningshanterare. Går alltså att köra app1 i ena env och app2 i andra env
`conda install NumPy, pandas, matplotlib, Seaborn, Jupyter`

Jupyter - Spinoff till [Fernando Pérez](#) tidigare IPython (-2014)

Koda, kör, visa output och kommentera i ett och samma dokument.

.ipynb filen är egentligen data i JSON format, men jupyter filer går att ladda ner som bl.a. HTML, LaTeX, PDF, Markdown och såklart Python filer. (Genom att använda [nbconvert](#))

Man kan såklart utveckla python med en IDE som t.ex JetBrains [Pycharm](#), men själv tycker jag bra om .ipynb

[Skriv kod, skicka till kompis](#) kör med shift+enter, inkludera kommentarer med [LATEX](#) syntax.

[DOCS](#)