

Introduktionsblad till Pythonkurs





*Välkomna till introduktionen till er första kurs i
programmering med Python!*

För att göra det så lätt som möjligt för er att komma igång med programmeringen i er första pythonkurs ber vi er att förbereda er genom att läsa igenom detta kompendium och gå igenom de uppgifter som specificeras här – dessa uppgifter innefattar både installationer av program samt lättare läsning och kommandon som ni bör känna till. För den som inte jobbat med python tidigare kan det vara en del att sätta sig in i och förstå och tanken är att ni när kursen börjar ska ha bekantat er lite med de verktyg ni kommer behöva i kursen.



1. Övergripande information	4
1.1 Kurslitteratur	4
1.2 Övriga resurser	4
2. Terminalen/kommandotolken	4
3. Installera Conda och python	5
3.1 Python	5
3.2 Anaconda (conda)	5
4. Git	6
4. Visual Studio Code	7
5. Jupyter notebooks	7



1. Övergripande information

Det du behöver göra och kunna till kursens start kan sammanfattas i en lista:

Vad behöver du göra innan kursens start?

1. Läs introduktionen till *A Whirlwind Tour of Python*.
2. Lära dig grundläggande kommandon i en terminal/kommandotolk
3. Installera conda och kunna starta en virtuell python-miljö.
4. Klona ner ett git repository från github.
5. Få en övergripande förståelse för git och anaconda
6. Installera Visual Studio Code.
7. Installera Jupyter Notebooks.

Punkterna i denna lista kommer att gås igenom i tur och ordning under följande avsnitt – och vi börjar med punkt 1.

1.1 Kurslitteratur

Under de första veckorna av kursen kommer vi att utgå ifrån boken *A Whirlwind Tour of Python* (WTP) av Jake VanderPlas. Boken är gratis att ladda ner som pdf [här](#), men kommer även att finnas i kursens git-repo. Till boken finns en del övningar (även de i ett git-repo) som kommer användas i kursen.

1. Inför kursen vill vi att du har läst igenom introduktionen till *A Whirlwind Tour of Python*.

Senare i kursen kommer vi även att använda oss av boken *Python Data Science Handbook*, också den skriven av Jake VanderPlas. Den boken finns gratis direkt på [denna hemsida](#), men om man gärna vill ha PDF eller fysisk bok finns den även att köpa. Även till denna bok finns det gratis övningar och notebooks.

1.2 Övriga resurser

Den intresserade kan även använda följande resurser för att sätta sig in i programmeringsspråket python:

- [Learn python the hard way](#)

2. Terminalen/kommandotolken

För att jobba med python (och diverse andra utvecklingsverktyg) så behöver du ibland använda någon sorts **terminal/kommandotolk**. Beroende på ditt operativsystem så kommer det att skilja sig något i vilka baskommandon som finns. Du kommer att använda terminalen i flera av de följande avsnitten och även kontinuerligt under kursens gång. Beroende på ditt OS rekommenderar vi följande:

- Windows – alternativen rangordnade:
 - Powershell – rekommenderad då den stödjer Unix-kommandon
 - Git bash – stödjer också Unix-kommandon
 - Cmd – stödjer ej Unix-kommandon
- Unix (Linux/Mac): Den inbyggda terminalen.



När du öppnar din terminal så kommer du stå i en viss mapp. När du utför kommandon så kommer de endast att ske i den mapp du befinner dig – kommandotolken ser bara filerna som finns i just den mapp där den står för tillfället. För att kunna röra dig runt mellan olika mappar och utföra olika uppgifter måste du därför kunna navigera i filsystemet via terminalen.

De operationer du behöver kunna är:

- Skapa ny mapp
 - o Windows/Mac/Linux: `mkdir`
 - o Exempel: `mkdir newfolder`
- Visa innehållet i den nuvarande mappen:
 - o Windows: `dir`
 - o Unix (Mac/Linux): `ls`
- Byta mapp
 - o Windows/Mac/Linux: `cd`
 - o Exempel: `cd newfolder`
 - o För att gå upp ett steg i filsystemet: `cd ..`
 - o För att gå upp 2 steg i filsystemet: `cd ../../`

Prova att öppna kommandotolken/terminalen på din dator och utför dessa exempelkommandon, för att se hur det fungerar.

Du kan läsa mer om kommandotolken och vilka unix-kommandon som finns och är bra att kunna på [denna](#) länk.

3. Installera Conda och python

3.1 Python

Python är ett programmeringsspråk som används mycket för dataanalys och maskininlärning (ML). Det går även att bygga backend till hemsidor med olika ramverk i Python, till exempel Flask och Django.

Om du har använt python tidigare har du det säkert installerat på din dator redan. Kanske har du även använt dig av virtuella miljöer också – **virtual environments**. I denna kurs kommer vi att använda oss mycket av virtual environments för att se till att alla paket vi behöver är installerade. Om du inte har python installerat på datorn så kommer det automatiskt när du installerar Anaconda.

När vi skriver vår kod i python kommer vi använda oss av textredigerare som **Visual Studio Code (VSC)** och även använda oss av program som kallas **Jupyter Notebooks**. Vi kommer att gå igenom konceptet Jupyter Notebooks i början av kursen, samt även se över textredigeraren VSC.

3.2 Anaconda (conda)

Med conda kan man skapa **virtual environments**, som är isolerade pythonmiljöer. Men varför vill man det egentligen? Jo, när man jobbar med python är det vanligt att man för ett givet projekt installerar olika paket och moduler (mer om det i kursen) och det är inte ovanligt att något paket krockar med ett annat. För att undvika onödiga krockar mellan olika projekt brukar man skapa en virtuell miljö för varje projekt, där man bara installerar de paket man ska använda för just det projektet. Detta kan man göra mycket enkelt med conda.

1. Installera conda via [deras hemsida](#)



2. Skapa en ny conda environment
 - a. Öppna din terminal
 - b. Ange kommandot: `conda create --name testenv python=3.8`
 - c. Ange kommandot: `conda activate testenv`
3. Du har nu skapat din första miljö och aktiverat den. Allt pythonrelaterat du gör i den terminalen du har öppen kommer att vara specifikt för miljön `testenv`.
4. För att se vilka paket du har installerat i din miljö kan du ange kommandot `conda list`.

Nästa gång du öppnar en terminal så kommer inte din miljö att vara aktiverad. Vill du använda den måste du återigen ange kommandot:

```
conda activate testenv
```

Mer information om conda environments kan du hitta [här](#).

4. Git

Git är ett verktyg som används för versionshantering och det används av nästan alla utvecklare, både i industrin, inom akademien och för privata projekt. Vi kommer inte att gå in på hur git fungerar i denna kurs, men vi kommer att distribuera kursmaterial via **GitHub** och ni kommer att lära er grunderna i git versioning. GitHub är en online plattform där man via git kan ladda upp sin kod och se andras kod.

1. Börja med att skapa ett konto på GitHub, om du inte redan har det.
2. Om du kör Linux/Mac OS har du antagligen redan git installerat. Om du kör Windows eller vill ha en nyare version av git kan du installera det genom att följa instruktionerna på följande länk:
 - [Installera git](#)
3. När du har skapat ditt konto och laddat ner git ska du kolla på denna video:
 - [Vad är git? En två minuters video](#)

Om du vill göra dina git-kommandon enklare i framtiden kan du även koppla på en SSH-authenticator till ditt GitHub-konto, genom att följa stegen på [denna](#) länk. Annars kan du använda dina inloggningsuppgifter när du utför dina git-kommandon i terminalen.

Vi har ett repository för kursen som du kan [hitta här](#). För att få ner det lokalt på din dator behöver du öppna en terminal och använda git-kommandon enligt följande instruktioner:

1. Öppna din terminal och navigera till en plats på din dator där du vill ha kursmaterialet
2. Skriv in följande kommando i terminalen:

```
git clone https://github.com/NordAxon/EC-python-course
```
3. Öppna mappen genom kommandot `cd EC-python-course`
5. Skapa nu en ny virtual environment med hjälp av conda och aktivera den. I mappen du står i finns det en fil som heter `requirements.txt`. Den filen innehåller en lista på pythonpaket som du ska installera i din conda environment. För att göra det så skriver du

```
pip install -r requirements.txt
```

`pip` är ett program som installeras automatiskt med python, som gör installationen av externa bibliotek till python mycket enkelt - `pip` är något du kommer använda mycket som pythonanvändare. Mer info om `pip` och hur det används kommer vi att gå igenom under kursen.



4. Visual Studio Code

Visual Studio Code är en gratis text-/kodredigerare från Microsoft som stödjer många olika språk. Under kursens gång kommer vi att utveckla kod framför allt i VSC, men om du föredrar PyCharm eller något annat program är du välkommen att använda det istället. Under föreläsningarna kommer vi att visa vår kod i VSC.

1. Installera VSC genom instruktionerna som finns [här](#).

5. Jupyter Notebooks

Jupyter Notebook är en webbapplikation som låter en skapa och dela dokument som innehåller live-kod, ekvationer, visualiseringar och berättande text. Användningsområden inkluderar data cleaning, dataanalys och transformation, numerisk simulering, statistisk modellering, datavisualisering, maskininlärning och mycket mer.

Ofta använder vi notebooks för att testa små kodsnuttar innan vi integrerar dem med våra hela skript. Ibland utvecklar vi även bara i notebooks när vi vill analysera och visualisera vårt data på olika sätt, då en notebook är mycket förlåtande då den sparar variabler mellan sina "celler".

För att kunna köra notebooks i din browser eller i VSC behöver du ha ett extra paket installerat i din conda environment.

1. Gör detta genom att köra följande kommando inuti din conda environment:
`conda install ipykernel`
2. Om du vill testa att öppna en jupyter notebook fil, kan du följa instruktionerna [här](#)