



UPPSALA  
UNIVERSITET

# Välkomna till Programmeringsteknik I

Med Python som språk

Lärare: Torsten Andersson, Anna Eckerdal



# Kursmål

Efter godkänd kurs ska studenten kunna:

- förklara vad ett givet program i Python utför;
- använda befintliga moduler såsom Pythons standardbibliotek och numpy utifrån deras dokumentation och andra resurser;
- analysera och lösa problem med hjälp av programmeringskonstruktioner;
- använda en programutvecklingsmiljö;
- testa och felsöka program;
- redogöra för de grundläggande begreppen modul, funktion, klass, objekt och därtill hörande underbegrepp;



# Vad är programmering?

- ”Att få datorn att göra det vi vill” - för att människor ska slippa
- ”Att analysera och dela upp problem i väldefinierade, kända moment”
- Att skriva kod
- Att använda kod
- Att läsa kod
  - Kommunikation med människor, inte datorer
- Att använda alla tillgängliga resurser för att uppnå detta
  - Läroböcker, referensdokumentation, Google, Stackoverflow, ...
  - Men om något används är det viktigt att ändå förstår hur och varför
  - Lätt att hitta ”smarta tips” som är fel eller används fel



# Vad är Python?

- Ett generellt programmeringsspråk
  - Finns språk som bara är tänkta för ett användningsområde och främst finns där
    - MATLAB exempel på språk avsett för tekniska beräkningar
    - R för statistik
- Ett fritt programmeringsspråk: Mer än bara "gratis"
  - Python-specifikationen är öppet tillgänglig
  - Standard-implementationen av Python är öppen källkod
- Ett tolkat programmeringsspråk
  - Textkommandon kan köras direkt, påminner om MATLAB
  - Program måste inte byggas/kompileras innan de kan köras
  - Går (ofta) fort att skriva, långsammare att köra



# Vad är Python?

- Ett populärt programmeringsspråk både i universitetsutbildningar och i industrin inom olika tillämpningar:
  - Vetenskap
  - Artificiell intelligens (AI)
  - Skapande
  - Spel
  - Webbapplikationer



UPPSALA  
UNIVERSITET

# Introduktion till kursen

- Lärare:
  - Anna Eckerdal, [Anna.Eckerdal@it.uu.se](mailto:Anna.Eckerdal@it.uu.se)
  - Torsten Andersson, [Torsten.Andersson@it.uu.se](mailto:Torsten.Andersson@it.uu.se)
  - samt assistenter
- Studentportalen
- Kusens hemsida: <http://www.it.uu.se/edu/course/homepage/prog1/python/vt21>
- Se till att du har en fungerande mail och att du läser mailen emellanåt! Vi skickar mail till era [student.uu.se](mailto:student.uu.se)-email.
- Se till att du har zoom installerad. Du behöver kunna skapa egna zoom-rum. Se <https://uppsala.instructure.com/courses/16240/pages/aktivering-och-installation>
- Registrering / avregistrering



UPPSALA  
UNIVERSITET

# Python från början

- Jan Skansholm.
- Studentlitteratur
- **ISBN:** 9789144134932
- Svenska.
- Passar för  
Programmeringsteknik I (+?)





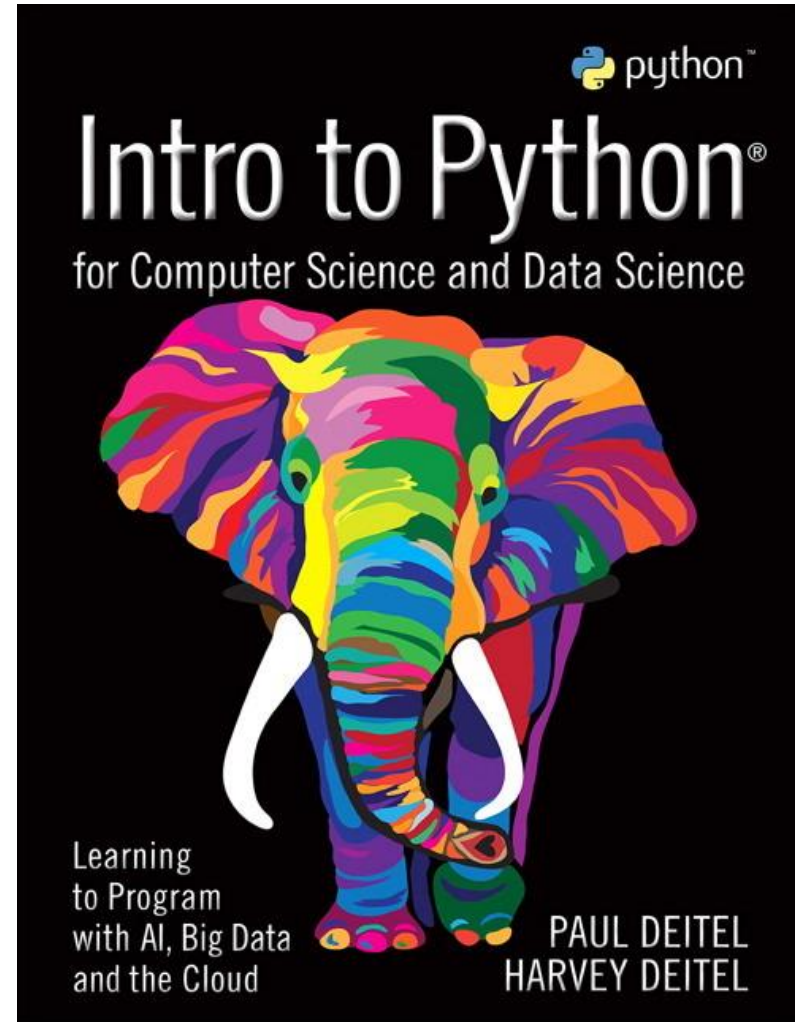
UPPSALA  
UNIVERSITET

# Intro to Python for Computer Science and Data Science

- Deitel & Deitel
- På engelska
- Passar både programmeringsteknik I och II
- Finns hos Akademibokhandeln LundeQ eller hos internet-bokhandeln, AdLibris eller Bokus.
- ISBN 9780135404676



Pearson







# Introduktion till kursen

- Undervisningsformer:
  - 8 föreläsningar
  - 26 laborationstillfällen (per grupp)
  - Alla föreläsningar kommer att ges via zoom
  - Alla laborationer kommer att ges via zoom, men labbsalar 2507, 2510, 2516, 1515 är öppna (inga labhandledare eller lärare där)
- Obligatoriska uppgifter: 3 hp
- Tentamen: 2 hp, 17 mars 2021
  - del A som är obligatorisk för alla och
  - del B för betyg 4 och 5.
- Kursmaterial
  - På nätet: Kursens material
  - Onlinetips: Se litteraturlänk på kurssidan



# Föreläsningar och laborationerna

- Ingen obligatorisk närvaro förutom vid redovisning (på zoom)
- Föreläsningarna ges via zoom. Zoom-länk finns på kurshemsidan.
- Under labbarna arbetar ni med 10 nätlektioner på kurshemsidan
  - Fem av dem är obligatoriska uppgifter (OU) som ska redovisas muntligen på zoom (TAR TID ATT GÖRA!)
  - Deadline för varje OU
    - Exakt datum beror på när din grupp är schemalagd
  - OK att gå på andra gruppers tider om det finns plats, men endast de som är schemalagda på tiden kan påräkna handledning och redovisningstid. (Tips: kl 8-10)



# Laborationerna

- Se till att arbeta själv med datorn
  - Egen dator hemma eller terminalen i datorsalen.
- Helt okej att ta hjälp och diskutera med varandra MEN skriv din egen kod!
- Kör du fast?
  - Skriv upp ditt namn på lista ”Hjälp och Redovisning på zoom” (Google-dokument) och vänta på att lärare/assistent ger tips om hur du kommer vidare
- Klar att redovisa?
  - Skriv upp ditt namn på lista ”Hjälp och Redovisning på zoom”
  - Skriv gärna upp namnet på den du redovisat för - det är lättare att reda ut om någon bokföring missas.



# Laborationerna

- Assistenternas uppgift är INTE att hitta felet i era program utan att tala om hur man ska bära sig åt för att hitta felet.
- Assistenterna uppgift är INTE att återberätta föregående föreläsningar eller lektioner.
- Om du har missat en föreläsning så måste du läsa på motsvarande material själv!
- Räkna med att du måste lägga ner en hel del tid utöver de schemalagda laborationerna! Kursen ska motsvara 133 timmar:
  - 8 föreläsningar: 16 h
  - 26 labbtillfällen: 52 h
  - Tentan: 5 h
  - Tid kvar att jobba hemma: 60 h



# Obligatoriska uppgifter, deadlines

- Se till att ha leg, eftersom detta är examination
  - Du förklara din kod individuellt, vad som skulle hända om man gjorde ändringar, svara på frågor från uppgiftstexten
  - Koden måste inte bara fungera, den måste vara tydlig också
  - Vid redovisningarna ges extra frågor och kodningsuppgifter.
  - Anteckna namnet på personen som du redovisat för!
- Redovisning måste ske senast vid deadline
- Absolut omöjligt?
  - Be oss om dispens. Assistenten kan inte ta emot redovisningar efter deadline utan dispens.



# Obligatoriska uppgifter, deadlines

- Om du inte blir godkänd på en redovisning så har du en vecka på dig att komplettera.
- Om du har flera uppgifter kvar får du göra uppgifterna nästa gång kursen ges (troligen period 1, ht 2021)
- Vi ordnar ibland några repetitionstillfällen inför omtentan i augusti. I samband med dessa kan det gå att redovisa någon enstaka uppgift.
- **Resultat förs in i Studentportalen: kolla "Framsteg"!**



# Hur ska man göra för att klara kursen?

- Pedagogisk forskning:
  - Det enskilt viktigaste momentet är eget arbete vid tangentbordet! (Men inte bara skriva av kod)
- Uppsalaundersökning:
  - Nybörjarprogrammerare som fick betyg 4 eller 5 på tentan har suttit dubbelt så lång tid vid tangentbordet som de som blev underkända. Inga andra faktorer hade signifikant påverkan.



# Laborationerna

- ***Installera Python på din egen dator, se kurshemsidan:***

<http://www.it.uu.se/edu/course/homepage/prog1/python/vt21>

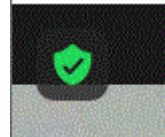
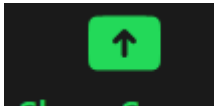
- Datorsystem i labsalarna
  - Linux med "tunna klienter" på institutionen.
  - Konto: UpUnet-S, lösenord A.
- Nätlektion 1 innehåller info om institutionens datorer och om hur du installerar programvara på din egen dator.
- Logga in med studentkontot på institutionens datorer.





UPPSALA  
UNIVERSITET

# Att få hjälp eller redovisa via zoom:

- *Laborationer: lab-salar på Polacksbacken finns för de som vill. Lärare och labassar kommer att vara tillgängliga via zoom (inte i labsal):*
  1. Så här skapar du ett nytt zoom-rum:  
<https://uppsala.instructure.com/courses/16240/pages/aktivering-och-installation>
  2. Hitta zoom-adressen: Text under "Participants" – "Invite" - "Copy Invite Link". Eller uppe till vänster under v:  

  3. Gå till Google-dokumentet "Redovisning och Hjälp" och skriv in ditt namn och adressen till zoom-rummet:  
<https://docs.google.com/spreadsheets/d/1vieIYS2iQpk72chYoXwDEQaO-foTpldvlpM1TeXdQPs/edit#gid=0>
  4. En lärare eller labass kommer till ditt zoomrum. Du kan dela din skärm.  




# Alla: Dags för lab!

- Målet i dag: gör klart nätlektion 1, börja på nätlektion 2:
  - Gå till kurshemsidan, välj Lektion 1 i vänstermenyn. Installera Thonny. Liten intro till Python.
  - Hjälp? Skriv upp namn och zoom-adress i Google-dokumentet "Redovisning och Hjälp"
- Lab idag: 9:15 – 12:00.
- Tips: kl 8:15 lätt att få hjälp.



UPPSALA  
UNIVERSITET

Programmeringsspråk kan se rätt olika ut.  
Här är ett exempel med språket APL. Raden  
skapar ett spel.

APL:

[https://en.wikipedia.org/wiki/Conway%27s\\_Game\\_of\\_Life](https://en.wikipedia.org/wiki/Conway%27s_Game_of_Life)

life←{↑1 ωV.∧3 4=+/, - 1 0 1°.Θ- 1 0 1°.Φ⊂ω}