

Välkomna till Programmeringsteknik I

ht 2022, period 1 - 2

Lärare: Torsten Andersson
samt lärarassistenter

Kursen ges för åk 1, kandidatprogrammen i fysik och matematik samt gymnasielärarprogrammet.
Totalt cirka 140 st studenter

Kurshemsida

www.it.uu.se/edu/course/homepage/prog1/python/ht22_p12/

Tips: Gör ett bokmärke i din webbläsare för den sidan på din dator.

Kursen har ingen kursbok, utan kursmaterialet finns på kurshemsidan med länkar till annat material på nätet. Den som ändå önskar ha en bok:

Python från början

Jan Skansholm.

ISBN: 9789144134932

På Svenska

Passar för Programmeringsteknik I

Intro to Python for Computer Science and Data Science

Deitel & Deitel

ISBN: 9780135404676

På engelska

Passar både programmeringsteknik I och II

[Vi tar en titt på kurshemsidan...](#) (länk finns även i Studium)

Studium

- Studium: För meddelanden och bokföring av inlämningsuppgifter
studium.uu.se.
- Du har åtkomst till studium för kursen om du är registrerad på kursen.
- Kikar på hur det ser ut i kursens Studium...
- Registrering / avregistrering / omregistrering

Högskolepoäng och betyg

Kursen omfattar 5 hp (högskolepoäng)

- Obligatoriska uppgifter(OU): 3 hp
- Tentamen: 2 hp, 10 januari 2023, som planeras att vara en e-tentamen. Består av två delar:
 - del A som är obligatorisk för alla kan ge betyg 3 eller U
 - del B för betyg 4 och 5 givet att del A ger betyg 3.
- Kursbetyg är lika med tentamensbetyget, givet att alla OU är godkänd

Undervisningen

- 9 st föreläsningar, normalt 2x45 min
- 28 st schemalagda laborationstillfällen (per studentgrupp), normalt 2x45 min
- Vid laborationstillfällena är salar bokade (PC-salar eller vanliga lektionssalar).
- Rekommenderas dock att använda egen dator vid laborationerna.

Datorlaborationerna

- Datorlabbarna utgör merparten av undervisningen. Under dessa arbetar du med de 10 st **nätlektioner** som finns på kursens hemsida.
- De innehåller övningar och programmeringsuppgifter.
- Du får arbeta med dem när, var och hur du önskar, men gör nätlektionerna i den ordning de har.
- Under labbtillfällena finns lärare på plats i salen eller möjligtvis online (zoom) för att hjälpa till och svara på frågor.
- Tips: kl 8-10

5 st obligatoriska uppgifter (OU)

- Fem av nätlektionerna består av OU som ska redovisas muntligen vid labbtillfälle (i datasal/ev. online).
Notera: Dessa tar tid att göra.
- Varje OU skall redovisas senast vid ett givet labbtillfälle
Redovisa gärna vid ett tidigare labbtillfälle
 - Exakt datum beror på när din grupp är schemalagd

OU ...

Se till att ha legitimation, eftersom detta är examination

- Du förklarar 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 kan ges extra frågor och kodningsuppgifter.
- Anteckna namnet på personen som du redovisat för!
- Redovisning måste ske senast vid deadline (se kurshemsidan)
- Absolut omöjligt?
 - Be om dispens. Assistenten kan inte ta emot redovisningar efter deadline utan dispens.

OU ...

- Om du inte blir godkänd på en redovisning så har du en vecka på dig att komplettera.
- Resultat av redovisning (godkänd/rest) förs in i Studium.
- Efter ordinarie tentamen brukar vi ordna ett uppsamlingstillfälle där det kan gå att redovisa någon enstaka uppgift, (max två uppgifter)
- Om du har uppgifter kvar efter kursen får du göra uppgifterna nästa gång kursen ges (*). F.n. ges kursen fyra ggr per läsår.

(*) Om uppgifterna är ganska lika från kurs till kurs

Nätlektionerna

- Innebär eget arbete med programmeringsövningar, dvs man skriver program (pythonkod) och testkör dem.
- Övningarna kan utföras på egen tid och/eller vid schemalagda tider.
- Se till att arbeta själv med datorn
- Helt okej att ta hjälp och diskutera med varandra MEN skriv din egen kod!

Handledning

Handledning fås vid labtillfälle (i sal eller möjligtvis online). Redovisning av OU kan göras i datorsal eller möjligtvis online.

Online (möjligen)

- Skapa ditt eget zoom-rum (görs en första gång)
- Skriv upp ditt namn, adressen till ditt zoom-rum och ditt ärende på lista "Handledning och Redovisning på zoom", länk till listan finns i Studium.



UPPSALA
UNIVERSITET

Föreläsningarna

Föreläsningarna introducerar koncept och kompletterar nätlektionerna.

Programmering är ett hantverk

- Kan ej lära sig programmera genom att läsa en bok
- Måste öva, öva och åter öva.
- Skriva många program
- Bra att göra fel, ty man lär sig genom att göra fel
- Finns inget facit till en programmeringsuppgift, däremot lösningsförslag. Kan normalt lösa en uppgift på många olika sätt.
- Lär sig genom att se/förstå alternativa lösningar.

Kursen kräver eget arbete

- Räkna med att du måste lägga ner en hel del tid utöver de schemalagda labbtillfällena!
- Kursen är på 5 hp, 1.5 hp = en veckas heltidsstudier, dvs 5 hp motsvarar 133 timmar.

9 föreläsningar: 18 tim

28 labbtillfällen: 56 tim

Tentamen: 5 tim

Tid kvar att jobba "hemma": 54 tim

Lika
många
timmar



Lärarassistenterna

- Lärarassistenternas uppgift är INTE
 - att hitta felen i era program utan att tala om hur man ska bära sig åt för att hitta felen och ge ledtrådar på hur felen kan rättas till.
 - att återberätta föregående föreläsningar eller lektioner.
- Om du har missat en föreläsning så måste du läsa på motsvarande material själv.

Många tentamenstillfällen

Om du blir undergodkänd på tentamen (dvs ett U)

- Du får tentera nästa gång kursen ges.
F.n. ges kursen fyra ggr per läsår.
- Dessutom ges omtentor i juni och augusti.

Förkunskaper?

- Krävs inga som helst förkunskaper i programmering för kursen
- Kursen börjar med enklare övningar
- Sedan byggs det på med nya pusselbitar
- Måste vara med från början, annars blir det svårare...
- Mycket av övningarna kan göras på egen hand

Hur ska man göra för att klara kursen?

- Pedagogisk forskning:
 - Det enskilt viktigaste momentet är eget arbete vid tangentbordet! (Men inte bara skriva av annans 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.

Programmering

- Kul
- Intressant
- Kreativt
- Under kursen kommer ni få se många olika exempel på vad programmering kan användas till
- Stegvis ökar ni era färdigheter under kursen
- Kursen programmeringsteknik 2 är en försettningskurs.

Dags för nätlektion 1 idag!

Målet i dag:

- Göra klart nätlektion 1 – ***Grundläggande Python***
- Börja gärna med nätlektion 2.
- Ladda ned programmeringsmiljön Thonny/Python på egen dator (instruktioner finns i nätlektion 1)

Länk till kurshemsidan finns i Studium under "Välkommen till kursen"
Arbetet kan utföras på institutionens PC-datorer eller egen dator .

För att logga in på linux-dator: studentkontot, lösenord A.

- 9:15-12 i datorsal 10K1201 (KandMa, GyLär)
- 13:15-16 i datorsal 4103, 4104 (KandFy)



UPPSALA
UNIVERSITET

Vad är programmering?

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

Kursmål

Efter godkänd kurs ska studenten kunna:

- förklara vad ett givet program i Python utför;
- använda befintliga moduler såsom Pythons standardbibliotek;
- 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;



UPPSALA
UNIVERSITET

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