

# Kunskapskontroll 2 – Fördjupad Pythonprogrammering

## Bakgrund

Många företag och organisationer har automatiserade processer.

Det kan exempelvis vara en *pipeline* med flera steg, som ett Pythonprogram som hämtar data från ett API, där datan sedan går igenom ett eller flera steg där den bearbetas för att möta krav utifrån exempelvis kvalitet och hur känslig datan är. Slutligen uppdateras en SQL-tabell med den nya datan. Även detta sker automatiserat utan mänsklig övervakning.

Men hur görs detta? Genom att “schedulera” (schemalägga) program!

I den här kunskapskontrollen kommer du använda Schemaläggaren i Windows för att schedulera ett Pythonprogram att exekvera en vald tid. I denna video kan du se ett exempel på hur Schemaläggaren används: <https://www.youtube.com/watch?v=lzy8KNnqV0I>

## Instruktioner

De som satsar på betyget *Godkänd* ska göra ett flöde där ett Pythonprogram exekverar automatiskt genom att scheduleras. Programmet ska läsa in data och uppdatera en SQL-tabell. Innehåll och data väljer ni själva. Datan kan komma från en fil på din dator (till exempel i csv-format), eller hämtas från ett API eller från en SQL-databas. Det primära är att flödet fungerar för att bli godkänd.

Koden ska vara väldokumenterad och följa konventioner för kodstandarder.

De som satsar på *Väl godkänd* ska göra ett flöde där ett Pythonprogram exekverar automatiskt, bearbetar datan (exempelvis genom att ändra datatyp eller att formatera värden så att de följer ett visst format), och uppdaterar en SQL-tabell. Datan kan komma från en fil på din dator (till exempel i csv-format), eller hämtas från ett API eller från en SQL-databas.

Skriptet måste hantera eventuella *exceptions* och logga dem till en loggfil.

Det ska även finnas ett separat skript med automatiska test.

Koden ska vara väldokumenterad och följa konventioner för kodstandarder.

## Vad ska lämnas in i Omniway?

I Omniway ska du lämna in en GitHub länk som innehåller filerna för din inlämning.

Gör gärna en tydlig GitHub som bl.a. innehåller en README-fil, se t.ex. följande repository för hur en GitHub-struktur kan se ut: [LucaCappelletti94/ugly\\_csv\\_generator](https://github.com/LucaCappelletti94/ugly_csv_generator).

Notera, du anpassar GitHubens utseende utifrån dina behov.

Det skall även finnas ett separat dokument som heter självutvärdering där du besvarar följande frågor:

- Utmaningar du haft under arbetet samt hur du hanterat dem.
- Vilket betyg du anser att du skall ha och varför.
- Något du vill lyfta fram till Linus och Antonio?

### **Principer för betygssättning**

#### **För betyget Godkänd ska den studerande**

- Ha nått samtliga läranderesultat för kursen **För betyget Väl Godkänd ska den studerande**
- Uppnått kraven för betyget Godkänd • Den studerande demonstrerar en detaljerad förståelse inom valt fördjupningsområde och presenterar detta på ett systematiskt sätt med välskriven kod och pedagogiska exempel

Frågor? Prata med Linus på lektionen.

Lycka till!

Linus & Antonio