



## Introduksjon

Python har en flott egenskap i det at den er plattformuavhengig. Det betyr at om du skriver et Python-program på Windows kan andre som bruker for eksempel Mac eller Linux kjøre det samme programmet på sin maskin. Det som kan være et problem er at om du vil dele et spill du har laget med noen venner, så må de også installere Python for å spille det. I dag skal vi se på hvordan vi kan pakke sammen et Python-program slik at det kan brukes også om noen ikke har Python installert selv.

## Steg 1: Installasjon av cx\_freeze

For å gjøre om et enkelt Python-program til noe som kan installeres på en annen maskin skal vi bruke en pakke som heter `cx_freeze` (tenk at vi fryser Python-programmet inn i en installasjonspakke). Denne pakken er *ikke* en del av standard Python, slik at vi må installere den på egen hånd.

### ✓ Sjekkliste

- ☐ På **Windows**: Åpne et kommandovindu ved å skrive `cmd` enten i det lille søkefeltet som dukker opp når du klikker start-knappen, eller ved å bare å begynne å skrive på startsiden.
- ☐ På **Mac** eller **Linux**: Åpne en terminal. På Mac gjør du dette ved å trykke command-knappen samtidig som mellomromtasten, og deretter skrive `terminal`. På Linux trykker du super-knappen (som regel windows-knappen), og deretter skriver du `terminal`.
- ☐ Skriv den følgende kommandoen for å laste ned og installere siste versjon av `cx_freeze`:

```
pip install cx_Freeze
```

## Steg 2: Lag en beskrivelse av programmet ditt

For at `cx_freeze` skal vite hvilke program og filer vi vil pakke sammen må vi lage en beskrivelsesfil.

Vi antar nå at du har et laget et Python-program som heter `hello.py`. Hvis programmet ditt heter noe annet så bytt ut referansene til `hello.py` med navnet på ditt program nedenfor.

### ✓ Sjekkliste

- ☐ I den samme katalogen som `hello.py` (eller programmet ditt) ligger, lag en fil som heter `setup.py` og ser ut som følger (husk å endre navnet på programmet om nødvendig, og oppdater også `description` og `author` slik at de blir riktige):

```
from cx_Freeze import setup, Executable

executables = [
    Executable('hello.py'),
]

setup(name='hello',
      version='0.1',
      description='Mitt lille Python-program',
      author='Geir Arne Hjelle',
      executables=executables,
      )
```

## Steg 3: Pakk sammen programmet ditt

Når vi har beskrevet programmet er vi klare til å pakke det sammen.

### ✓ Sjekkliste

- ☐ Gå tilbake til kommandovinduet eller terminalen du brukte tidligere. Nå skal vi bruke dette til å pakke sammen programmet.
- ☐ Først må vi finne katalogen hvor programmet vårt ligger. Det enkleste vil være å finne programmet i en filutforsker (for eksempel Windows Utforsker eller Finder på Mac). Derfra kan du kopiere *katalogstien*. Skriv den inn i kommandovinduet eller terminalen med kommandoen `cd` (*change directory*) foran. For eksempel

```
cd C:\Hjem\Spill\
```

- ☐ Nå kan vi skrive kommandoen som pakker ting sammen for oss. Skriv følgende i kommandovinduet eller terminalen:

```
python setup.py build
```

Dersom dette virker dukker det opp en ny katalog som heter `build`. Denne katalogen inneholder alt som er nødvendig for å kjøre programmet ditt uten å installere Python. Inne i denne katalogen vil du finne en programfil som du kan kjøre, for eksempel ved å dobbelklikke på den. I eksempelet over vil denne filen hete `hello`.

## Steg 4: Lag et enkelt installasjonsskript

På Windows og Mac er det også mulig å lage et installasjonsskript ved hjelp av `cx_freeze` slik at programmet ditt kan installeres på akkurat samme måte som vanlige programmer.

### ✓ Sjekkliste

- ☐ På **Windows**: Bruk følgende kommando i stedet når du pakker koden din:

```
python setup.py bdist_nsi
```

Dette lager en vanlig Windows-installasjonsfil i katalogen `dist` som du kan dele med andre.

- ☐ På **Mac**: Det samme kan du gjøre på en Mac med

```
python setup.py bdist_dmg
```

**Lisens:** CC BY-SA 4.0 **Forfatter:** Geir Arne Hjelle