

Plan

1. Python installering og setup

Først installerer vi Python på maskinen din. Vi sørger for at alt er konfigurert riktig slik at du kan kjøre Python-kode lokalt. Vi går gjennom hvordan du laster ned Python, samt hvordan du setter opp en enkel utviklingsmiljø på datamaskinen din. (www.python.org)

2. Hva er programmering?

En introduksjon til hva programmering er, og hvordan det brukes til å løse problemer ved å instruere en datamaskin til å utføre bestemte oppgaver. Vi snakker om grunnleggende konsepter som koder, algoritmer, og programmeringsspråk.

3. Hvordan fungerer det?

Her forklarer vi det grunnleggende prinsippet bak hvordan datamaskiner tolker og kjører kode. Vi diskuterer hvordan programmer oversettes fra høyere nivå programmeringsspråk (som Python) til maskinkode som datamaskinen kan forstå.

4. Filsystem

Navigering ved hjelp av terminal vindu

Lær hvordan du navigerer i filsystemet på datamaskinen ved hjelp av terminalen. Dette inkluderer grunnleggende kommandoer som `cd` (change directory), `ls` (list directory contents), og `pwd` (print working directory).

5. Første python data program

Notisblokk -> skriv kode

Vi begynner med å skrive vårt første Python-program ved hjelp av en enkel teksteditor som Notisblokk. Du vil skrive en enkel kode og lære hvordan du lagrer den som en `.py` fil.

Starte fra terminal

Her lærer vi hvordan man kjører Python-koden du har skrevet direkte fra terminalen. Dette gir deg en praktisk forståelse av hvordan man interagerer med Python utenfor et IDE (Integrated Development Environment).

6. Pycharm intro

Introduksjon til PyCharm, et populært IDE for Python. Vi ser på hvordan PyCharm kan brukes til å skrive, teste, og feilsøke Python-kode mer effektivt. Vi går gjennom grunnleggende funksjoner som kode fullføring, debugging, og prosjektstruktur.

7. Python programmering

Formatering (PEP 8 – Style Guide for Python Code)

Vi diskuterer viktigheten av god kodeformatering og hvordan dette påvirker lesbarhet og vedlikehold av koden. Vi går gjennom PEP 8, den offisielle stilguiden for Python-kode.

- **Innrykk:** Korrekt bruk av innrykk i Python for å strukturere koden.
- **Linje lengde:** Viktigheten av å holde linjelengden til maks 79 tegn for bedre lesbarhet.

Print() funksjonen

Lær hvordan du bruker `print()`-funksjonen til å skrive ut informasjon til skjermen. Dette er et av de mest grunnleggende verktøyene i Python for å visualisere resultater og feilsøke kode.

Variabler

En gjennomgang av hva variabler er og hvordan de brukes i Python. En variabel er en navngitt plass i datamaskinens minne hvor data kan lagres for senere bruk.

- **Eksempel:** `x = 5` lagrer verdien 5 i variabelen `x`.

Input() funksjonen

Vi lærer hvordan man bruker `input()`-funksjonen til å ta inn data fra brukeren. Dette er nyttig for å lage interaktive programmer hvor brukerens input kan påvirke programmets oppførsel.

Datatyper

Vi utforsker ulike datatyper i Python:

- **String (str):** Tekst, f.eks. `"Hei, verden!"`
- **Number:**
 - **Int:** Heltall, f.eks. `10`
 - **Float:** Desimaltall, f.eks. `3.14`
- **Boolean:** Representerer sannhetsverdier, `True` eller `False`.

Casting

Vi kan endre datatypen på variabler/data

Regne-operasjoner

- `+` (addere)
- `-` (substrahere)
- `/` (divisjon)
- `*` (multiplisere)
- `//` (heltall-divisjon)
- `%` (modulus, rest)
- `**` (exponent)

Betingelser

En introduksjon til betingede utsagn i Python ved hjelp av `if`, `elif`, og `else`-setninger. Vi ser på hvordan man kan styre programflyten basert på ulike betingelser.

- **Eksempel:**

```
if x > 10:  
    print("x er større enn 10")  
elif x == 10:  
    print("x er lik 10")  
else:  
    print("x er mindre enn 10")
```