

# Utvikling av "Hangman" Spill med Norske Fornavn

---

## Introduksjon:

I denne oppgaven skal du utvikle et "Hangman" (Galgespill) i Python, hvor spilleren skal gjette et norsk fornavn hentet fra internett. Du vil lære å hente data fra en API og bruke denne informasjonen i spillet ditt. Oppgaven er delt inn i 10 steg som vil veilede deg gjennom prosessen.

---

## Steg 1: Planlegg Spillet

- **Beskrivelse:**

- Før du begynner å kode, planlegg hvordan spillet skal fungere. Hangman er et ordspill der spilleren prøver å gjette et ord ved å foreslå bokstaver innen et begrenset antall forsøk.

- **Oppgave:**

- Skriv ned flyten i spillet:
    - Hent et fornavn fra API-en.
    - Vis antall bokstaver som understreker eller stjerner (f.eks. `*****` for et navn med 6 bokstaver).
    - Spilleren gjetter en bokstav om gangen.
    - Oppdater visningen hver gang spilleren gjetter riktig.
    - Spilleren har et begrenset antall feilforsøk.
    - Spillet avsluttes når spilleren gjetter ordet eller bruker opp alle forsøk.
- 

## Steg 2: Importer Nødvendige Biblioteker

- **Beskrivelse:**

- Du trenger `requests`-biblioteket for å hente data fra API-en, samt `random` for eventuelle tilfeldigheter.

- **Oppgave:**

- Installer `requests`-biblioteket hvis du ikke har det allerede.
  - **Tips:** I terminalen, kjør `pip install requests`.
- Importer bibliotekene i begynnelsen av programmet ditt.
  - **Eksempel:**

```
import requests
import random
```

---

## Steg 3: Hent Data fra API-en

- **Beskrivelse:**

- Du skal hente 10 tilfeldige personer fra API-en og bruke fornavnet til den første personen.

- **Oppgave:**

- Bruk `requests`-biblioteket til å sende en HTTP GET-forespørsel til `https://webapi.no/api/v1/randomPersons/10`.
- Parse JSON-responsen og hent ut fornavnet til den første personen.

- **Eksempel på å sende en GET-forespørsel:**

```
response =  
requests.get('https://webapi.no/api/v1/randomPersons/10')  
data = response.json()
```

- Lagre fornavnet i en variabel for bruk i spillet.
- 

## Steg 4: Forbered Ordet for Spillet

- **Beskrivelse:**

- Gjør ordet klart ved å opprette en representasjon med understreker eller stjerner for hver bokstav.

- **Oppgave:**

- Lag en variabel som inneholder en liste eller streng med en understrek eller stjerne for hver bokstav i ordet.
    - **Eksempel:** Hvis ordet er "Ola", blir representasjonen `***`.
  - Opprett en liste over allerede gjettete bokstaver (start tom).
- 

## Steg 5: Vis Gjeldende Status til Spilleren

- **Beskrivelse:**

- Vis spilleren hvor langt de har kommet i å gjette ordet.

- **Oppgave:**

- Lag en funksjon eller et kodeavsnitt som viser den nåværende representasjonen av ordet med riktige gjettete bokstaver og understreker/stjerner for de som ikke er gjettet.
    - **Eksempel:** Hvis ordet er "Ola" og spilleren har gjettet "a", vis `**a`.
- 

## Steg 6: Sett opp Spilløkken

- **Beskrivelse:**

- Hovedløkken i spillet der spilleren fortsetter å gjette bokstaver til spillet er over.

- **Oppgave:**

- Bruk en `while`-løkke som kjører så lenge spilleren ikke har vunnet eller brukt opp alle forsøk.
  - Definer en variabel for maksimalt antall feilforsøk, og en teller for antall feil spilleren har gjort.
- 

## Steg 7: Ta imot Brukerens Gjetning

- **Beskrivelse:**

- Be spilleren om å gjette en bokstav og håndter input.

- **Oppgave:**

- Bruk `input()` til å ta imot en bokstav fra spilleren.
  - Sørg for at inputen er en enkelt bokstav og ikke allerede er gjettet.
    - **Tips:** Du kan bruke `.lower()` for å standardisere bokstaven.
  - Legg den gjettede bokstaven til listen over gjettede bokstaver.
- 

## Steg 8: Oppdater Spillet Basert på Gjetningen

- **Beskrivelse:**

- Sjekk om den gjettede bokstaven er i ordet og oppdater spillstatusen.

- **Oppgave:**

- Hvis bokstaven er i ordet:
    - Oppdater representasjonen av ordet slik at bokstaven vises på riktig plass.
  - Hvis bokstaven ikke er i ordet:
    - Øk telleren for antall feilforsøk.
  - Vis en melding til spilleren om resultatet av gjetningen.
- 

## Steg 9: Hold Styr på Spillstatus og Forsøk

- **Beskrivelse:**

- Overvåk om spilleren har vunnet eller tapt, og oppdater spillstatusen.

- **Oppgave:**

- Etter hver gjetning, sjekk om spilleren har gjettet alle bokstavene i ordet.
    - Hvis ja, avslutt løkken og gratuler spilleren.
  - Sjekk om spilleren har brukt opp alle feilforsøk.
    - Hvis ja, avslutt løkken og avslør ordet.
- 

## Steg 10: Avslutt Spillet og Vis Resultatet

- **Beskrivelse:**

- Når spillet er over, vis en passende melding til spilleren.

- **Oppgave:**

- Hvis spilleren vant:
    - Gratuler spilleren og vis antall feilforsøk de brukte.
  - Hvis spilleren tapte:
    - Fortell spilleren at de tapte og vis hvilket ord det var.
  - Gi muligheten til å spille på nytt hvis du ønsker å utvide spillet.
- 

### Ekstra Tips:

- **Håndtering av Ugyldig Input:**

- Sjekk at spilleren kun skriver inn en enkelt bokstav og ikke tall eller symboler.
- Hvis input er ugyldig, gi en feilmelding og be om ny input uten å bruke opp et forsøk.

- **Forbedre Brukeropplevelsen:**

- Bruk `time.sleep()` fra `time`-biblioteket for å legge inn små pauser etter meldinger.
  - **Eksempel:** `time.sleep(1)`
- Vis en enkel tegning av galgen som blir mer komplett for hvert feilforsøk.

- **Utvid Spillet:**

- La spilleren velge vanskelighetsgrad som bestemmer antall tillatte feilforsøk.
  - Hold styr på antall seire og tap hvis spilleren ønsker å spille flere runder.
- 

### Oppsummering:

Du har nå veiledning til å utvikle et "Hangman" spill i Python hvor ordet som skal gjettes er et norsk fornavn hentet fra en API. Dette gir deg praksis i å bruke eksterne data i programmeringen din, samt anvende kontrollstrukturer og datastrukturer du har lært.

**Lykke til med programmeringen! Husk å teste programmet ditt grundig for å sikre at alt fungerer som det skal.**