

PGZ - Hangman

Introduksjon

I denne oppgaven skal vi lage vårt eget hangman-spill. Vi har laget litt du [laste ned her](#). Lagre den der du pleier å lagre Python-koden din.

Det kan se ut som mye kode, men det bare den øverste delen du skal som ser slik ut:

```
#####  
# Library code  
#####
```

All koden under denne overskriften kan du bare overse. Dette er kode fra brukeren.

I koden over overskriften har vi laget noen variabler:

- **TRIES** som angir hvor mange forsøk brukeren har. Ettersom det denne være slik den er.
- **WORDS** er en liste over ord som kan gjettes, men ingen av disse
- **state** er en ordbok som inneholder spillets tilstand.

Vi har begynt å lage fire funksjoner for deg. Disse skal du gi innhold i

Steg 0: Test at alt fungerer

Ved å kjøre kommandoen

```
pgzrun hangman.py
```

skal du få opp et vindu som ser slik ut:



Dersom du ikke får opp et vindu som beskrevet over, pass på at du har [hangman.py](#) og er i samme mappe som fila.

Dersom du fremdeles sitter fast bør du snakke med en veileder.

Steg 1: Lage det hemmelig

Nå skal vi skrive koden som skal kjøres når `create_display_string()` `secret_word` (det hemmelige ordet) og `remaining_letters` (en liste

Her er et eksempel på hvordan funksjonen kan fungere:

```
>>> remaining_letters = ['b', 'c', 'd', 'e', 'f', 'g', 'h', 'i',  
                        'm', 'n', 'o', 'p', 'q', 'r', 's', 't',  
                        'x', 'y', 'z'] # alle bokstaver fra b  
>>> secret_word = 'banan'  
>>> create_display_string(secret_word, remaining_letters)  
'_ a _ a _'  
>>> remaining_letters.remove('b')  
>>> create_display_string(secret_word, remaining_letters)  
'b a _ a _'  
>>> create_display_string(secret_word, [])  
'b a n a n'
```

Her er alle bokstaver i `secret_word` som også er i `remaining_letters` mellom hvert tegn.

For å lage denne funksjonen trenger vi noen av tekstfunksjonene vi la være lurt å gå tilbake og lese igjennom alle de gule boksene i oppgav

Du skal nå skrive kode i funksjonen. Dette må du gjøre:

- ☐ For å konvertere `secret_word` til ei liste må du bruke funksjonen `list()`, også returneres en liste av bokstaver.

```
>>> list("ord")  
['o', 'r', 'd']
```

- ☐ Slå sammen bokstavene til en streng igjen, med ett mellomrom

Hint: `s.join(lst)`

- ☐ Bytt ut hver bokstav som er i det nye ordet vårt og i `remaining`

Hint: `s.replace()`

- ☐ Husk å returnere det hemmelige ordet.

Steg 2: Å starte spillet

Nå skal vi kode funksjonen `start_game()`. Denne funksjonen lager nøkkelverdi-parer som forteller hvordan ordbøker fungerer kan det være lurt å repetere [oppgaven om](#)

Nøkklene i `state` er som følger:

- ☐ `"running"` forteller hjelpefunksjonene om spillet kjører
- ☐ `"used_tries"` er hvor mange forsøk brukeren har brukt
- ☐ `"secret_word"` er det ordet brukeren skal gjette
- ☐ `"pressed_button"` er den siste knappen som ble trykket av brukeren
- ☐ `"help_text"` er hjelpetekst for brukeren, som forklarer hva brukeren kan gjøre
- ☐ `"remaining_letters"` er en liste over de bokstavene brukeren ikke har prøvd
- ☐ `"display_string"` er det ordet som vises til brukeren (etter at den er formatert)

Dette må du gjøre:

- ☐ Lag følgende nøkkel/verdi-par:
 - ☐ Nøkkel: `"running"` Verdi: `True`

- ☐ Nøkkel: `"used_tries"` Verdi: `0`
- ☐ Nøkkel: `"pressed_button"` Verdi: `""` (En tom streng)
- ☐ Nøkkel: `"help_text"` Verdi: `"Guess a letter!"`

- ☐ For å velge et tilfeldig ord skal vi bruke `random.choice()`. Ved å returneres et tilfeldig element i lista.
 - ☐ Gi `WORDS` som argument og bruk det returnerte ordet som

- ☐ Vi skal nå lage lista over bokstaver som brukeren ikke har gjettest `string.ascii_letters`, og legger disse til lista.
 - ☐ La den tilhørende verdien til `"remaining_letters"` være `0`
 - ☐ Bruk `enumerate()` og gå igjennom de 26 første bokstaver `state["remaining_letters"]`.
Hint: Bruk `lst.append(elm)`

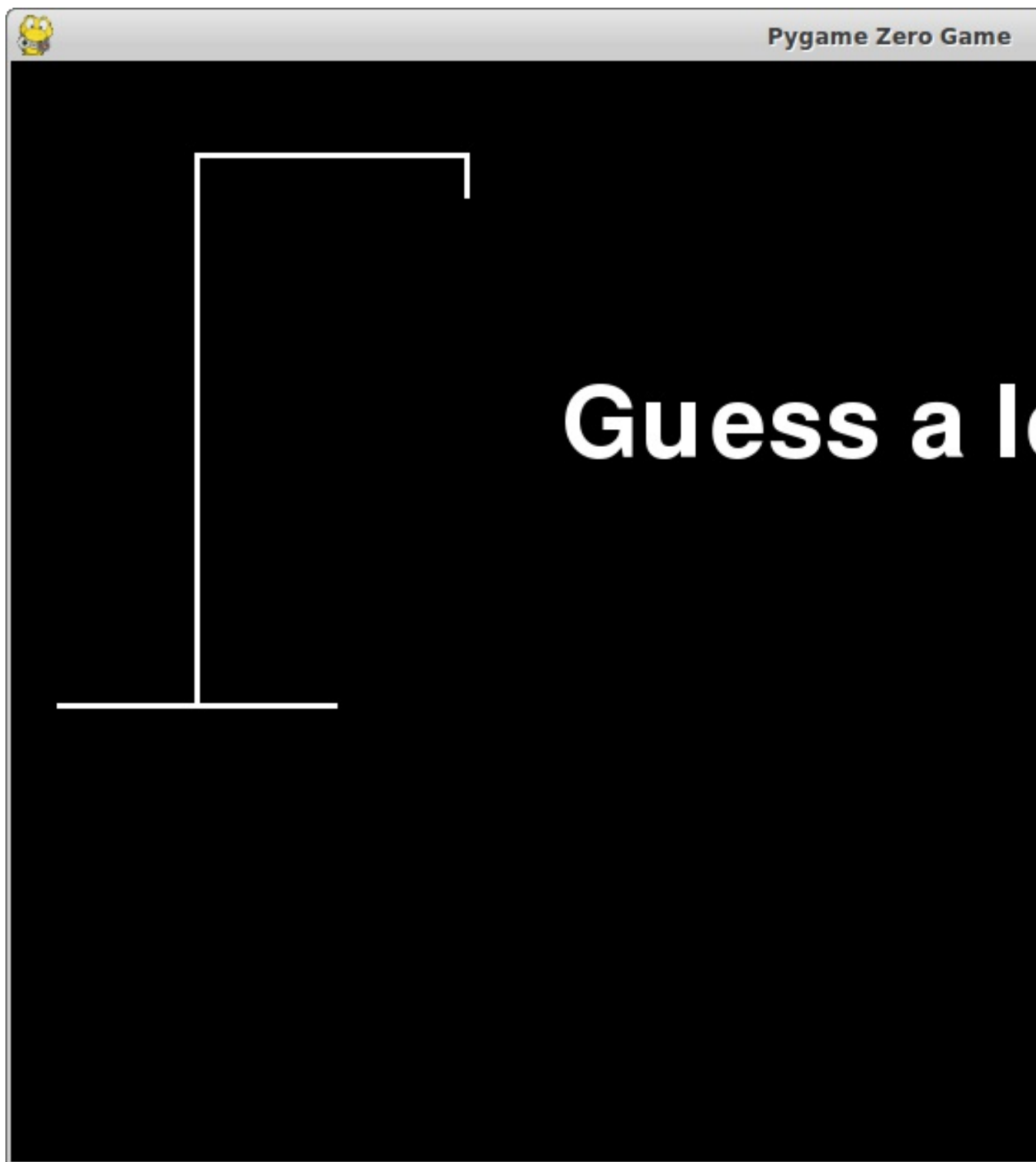
- ☐ For å lage verdien som hører til `"display_string"` må vi bruke `format()` vi gi til funksjonen?

Test spillet ditt

Lagre endringene dine i `hangman.py`, og kjør spillet ditt:

```
pgzrun hangman.py
```

Du skal nå få opp et vindu som ser slik ut:



Steg 3: Å avslutte spillet

I funksjonen `game_over()` skal vi endre `state` slik at både bruker og

Dette må du gjøre:

- ☐ Endre `state["running"]` til `False`.
- ☐ Endre verdien til `state["display_string"]` slik at brukeren kan se ordet. Dette kan du gjøre ved hjelp av funksjonen fra [steg 1](#).

Hint: Hva skal listen vi gir som argument inneholde når vi ønsker å oppdatere display_string?

Steg 4: main()

I dette steget skal vi skrive hovedfunksjonaliteten til spillet. Vi skal skrive funksjonen `main()` som oppdateres. Her skal vi finne ut hvilken bokstav brukeren trykket på, og om den er i ordet vårt. Vi må også oppdatere antall brukte forsøk, og til slutt sjekke om brukeren har gjettet riktig ord.

Dette må du gjøre:

- ☐ Du finner bokstaven brukeren trykket på i `state["pressed_button"]` variabel, så du slipper å skrive `state["pressed_button"]` mange ganger.
- ☐ Dersom bokstaven finnes i `state["remaining_letters"]`, må du oppdatere listen.
 - ☐ Fjern bokstaven fra lista
 - ☐ Dersom bokstaven finnes i det hemmelige ordet vårt må du oppdatere ordet. Hvis brukeren brukt opp ett forsøk, og du må legge til en til `state["used_tries"]`.
- ☐ Sjekk om brukeren har brukt opp alle forsøkene sine. Dersom de har, må du skrive `"You lost!"`. Til slutt må du kalle funksjonen `game_over()`.

Hint: Sjekk om `state["used_tries"]` er større eller lik `TRIES`.

- ☐ Vi må også sjekke om brukeren har gjettet ordet. En måte dette `state["display_string"]` . Bruk `s.count("_")` for å telle antal har brukeren vunnet spillet, og du må da endre `state["help_te`



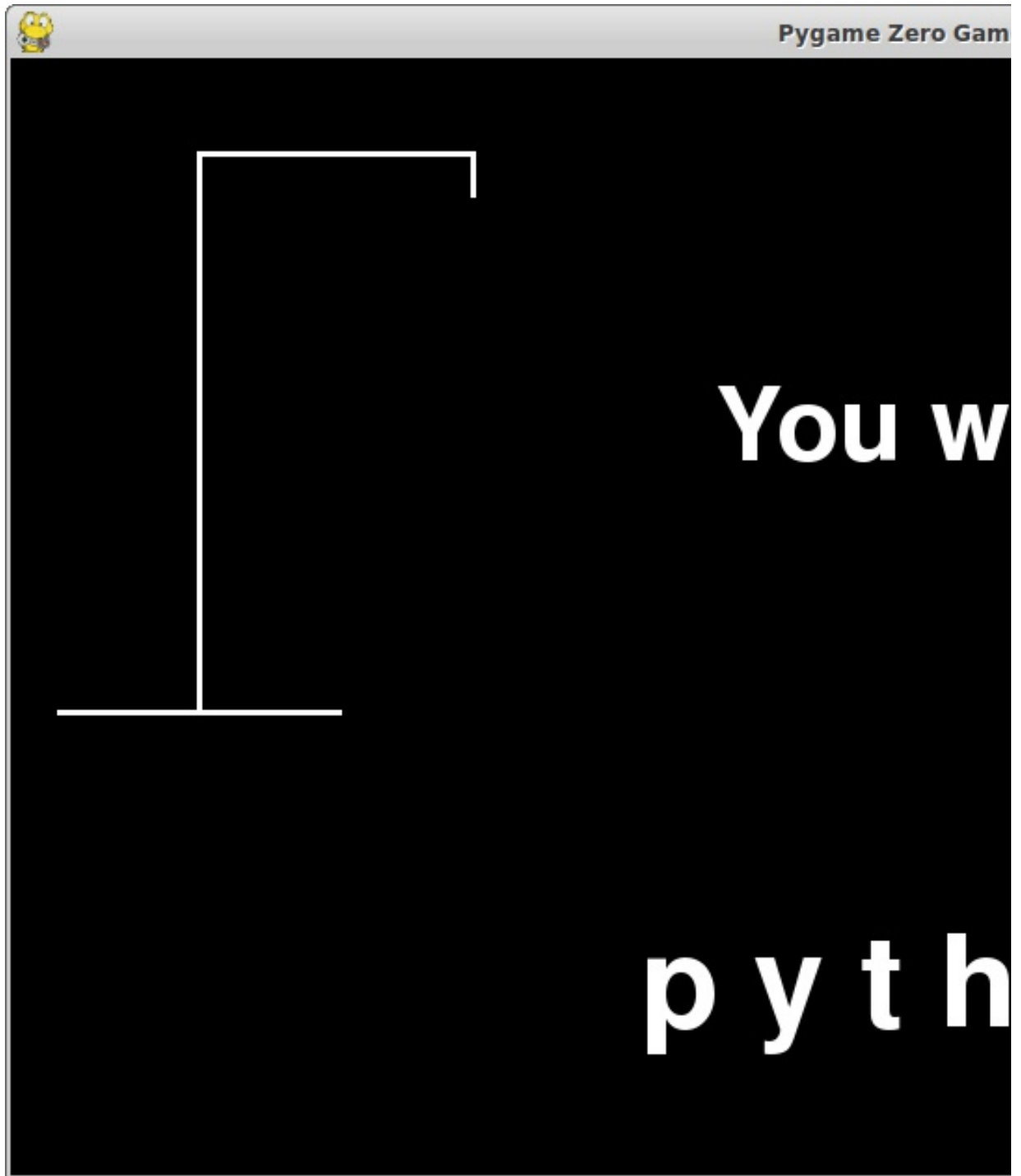
Test spillet ditt

Lagre endringene dine i `hangman.py` , og kjør spillet ditt:

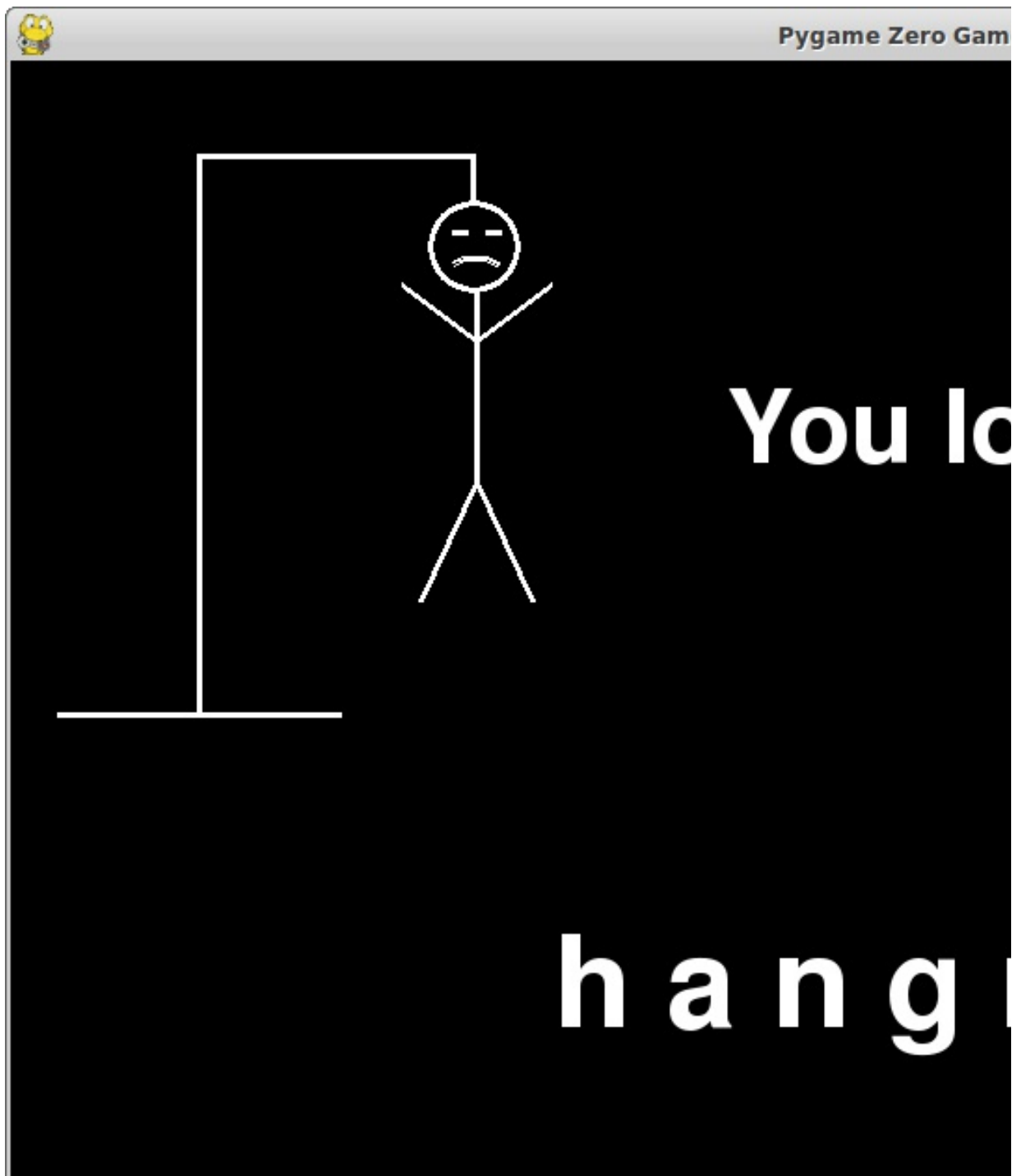
```
pgzrun hangman.py
```

Spillet skal nå fungere fullt og helt. Nå er det noen ting vi må teste:

- ☐ Slik kan det se ut når spilleren vinner.



- ☐ Når spilleren taper kan det se ut omtrent som i bildet under. Pass på at du viser løsningsordet nederst.
- ☐ Det skal ikke være mulig å taste inn flere bokstaver. Dersom du bruker `state["running"] = False` i `game_over()`.
- ☐ Pass på at du viser løsningsordet nederst.



Lisens: [CC BY-SA 4.0](https://creativecommons.org/licenses/by-sa/4.0/) **Forfatter:** Ole Kristian Pedersen, Kodeklubb