

1 Egenerklæring (med bekreftelse)

Jeg erklærer herved at besvarelsen som jeg leverer er mitt eget arbeid og

- ikke har vært brukt i en annen eksamen eller vært levert eller publisert ved en annen utdanningsinstitusjon innenlands eller utenlands.
- ikke inneholder andres arbeid uten at dette er oppgitt.
- ikke inneholder eget tidligere arbeid uten at dette er oppgitt.

Jeg er kjent med at brudd på disse bestemmelsene er å betrakte som fusk.

Dersom du er usikker på om du kan stille deg bak erklæringen, se [retningslinjer for bruk av kilder i skriftlige arbeider ved Universitetet i Bergen](#) og eventuelt ta kontakt med veileder/emneansvarlig.

Alle innleveringene dine ved UiB kan bli sendt til elektronisk plagiatkontroll.

Merk: Det er ikke anledning til å levere besvarelser som ikke oppfyller kravene i egenerklæringen.

Velg ett alternativ

- ☐ Jeg bekrefter at jeg har lest egenerklæringen og at besvarelsen er mitt eget arbeid

Maks poeng: 0

i Generelle råd og kommentarer:

- Les nøye gjennom oppgavene før du begynner å svare.
- Dersom du ikke klarer å gi fullstendig svar til en oppgave, kan du likevel fortsette. Legg inn en kommentar som beskriver hva du skulle ha gjort i delen som mangler
- Koden din bør være leserlig og enkel å forstå. Velg gode variabelnavn og tydelig oppsett. Lag hjelpefunksjoner der de er nyttige
- Syntes du at oppgaveteksten er uklar eller ufullstendig, må du lage dine egne forklaringer og gi disse i svaret som kommentar.
- Prosentsatsene ved hver deloppgave gir omtrentlig vektlegging ved sensur

Kontakt:

- Se: https://mitt.uib.no/courses/21612/discussion_topics/169436

Lykke til!

Velg den passende datatypen for hvert uttrykk eller *(-error-)* om det ikke er et gyldig uttrykk i Python 3.7.

2

- a = "Hello"
- b = 1.3
- c = 7
- d = True

Choose the data type

	bool	float	int	list	(-error-)	str
a + 'b'	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
a*b	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
b*c	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
f'{d}'	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
a+b	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
[d]	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
len(a)	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
c*a	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
c == 10.3	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
a+a	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>

Maks poeng: 5

Velg de passende alternativene i hvert boks så at koden passer beskrivelsen. Hver oppgave teller med 2%.

3 Velg rett alternativ slik at alle sammenligningene er True. Dict xs ser sånn ut:

```
xs = {  
  'a' : 5,  
  'd' : 'hello',  
  'z' : 3.1415,  
  5 : 7  
}
```

(xs[0], xs['a'], xs[5], xs[a]) == 5

'hello' in

(xs.items(), xs.keys(), xs, xs.values())

(list(xs.keys()), list(xs.values()), list(xs[:,0]), list(xs.items()))== ['a','d','z',5]

(a, 7, 'a', 5) == xs[5]

Maks poeng: 2

4 Be om en tekst og skriv ut lengden av teksten, gjenta helt til brukeren gir en tom tekst

(for x in [], for _ in True:, while True:, while False:)

• text =

(open("Text: "), input("Text: "), print("Text: "), read("Text: "))

• if text == "":

◦ print("Bye")

(None, break, continue, pass)

• print(f"Your text had

{ len text }, len(text), { len(text) }, len{text} characters."

Maks poeng: 2

3/12

5 Velg resultatet av hvert *bool/sk* uttrykk.

	False	True
5 in range(0,10,3)	<input type="radio"/>	<input type="radio"/>
False or True	<input type="radio"/>	<input type="radio"/>
15 // 2 > 7	<input type="radio"/>	<input type="radio"/>
not (not True)	<input type="radio"/>	<input type="radio"/>
True and False	<input type="radio"/>	<input type="radio"/>
5 < 7 and 4 > 5	<input type="radio"/>	<input type="radio"/>
list(range(7))[-1] == 7	<input type="radio"/>	<input type="radio"/>
3 < 7 < 9	<input type="radio"/>	<input type="radio"/>
[x**2 for x in range(3)] == [0,1,4]	<input type="radio"/>	<input type="radio"/>
5 in range(6)	<input type="radio"/>	<input type="radio"/>

Maks poeng: 2

6 Les inn hver linje fra en tekstfil og print den på skjermen baklengs.
Linjen "abcd" skal skrives ut som "dcba"

```
filename = "foo.txt"
[with, open, file, read] [read(filename), with(filename), filename, open(filename)]
[with f:, to f:, as f:, from f:]

• for line in f:
  ○ line = line.strip()
  ○ line = [line[-1], line.lower(), line[::-1], line.split()]
  ○ print(line)
```

Maks poeng: 2

7 Skriv om denne løkken med *while* i stedet for *for*:

for x in xs:

- if x > 5:
 - print(x)

(i = len(xs), i = xs, i = None, i = 0)

while (i < len(xs):, i:, True:, i < xs:)

- (x = xs[i], x = xs, x = xs[0], i = xs[0])
- if x > 5:
 - print(x)
- (i += 1, return x, break, x += 1)

Maks poeng: 2

8 Velg de riktige linjene slik at programmet skriver ut:

- A
C
D

text = "oppvaskmaskin"

(if 'a' not in text:, if 'a' in text:, if a in text:, if a not in text:)

- print('A')

(elif o in text:, if 'o' in text:, else:, elif:)

- print('B')

(elif k in text:, if k in text:, elif 'k' in text:, if 'k' in text:)

- print('C')

(else:, if 'm' in text:, elif 'm' in text:, elif:)

- print('D')

Maks poeng: 2

9 Velg rett alternativ slik at alle sammenligningene er True. Listen xs ser sånn ut: xs = ['a','b','c','d']

(xs[2:3], xs[-1], xs[2], xs[3]) == 'c'

'abc' == (xs[:4], str(xs[0:3]), str(xs[:4]), xs[0:3])

(xs[:::-1], xs[:-1], xs[:::-1], xs[-1]) == ['d','c','b','a']

('b', 'd', 'c', 'a') > xs[-2]

Maks poeng: 2

10 Returner True om alle verdiene i *input* listen er like, ellers returner False

```
def all_same(input):  
    • x = input[0]  
    • for e in input[1:]:  
        ◦ if x != e:  
            ■  (continue, break, return True, return False)  
    • return  (e != x, True, False, e == x)
```

Maks poeng: 2

11 Gitt to tall a og b, returner True om begge er mindre enn 100, ellers False

```
def both_below_100(a,b):  
    •  (return a < 100 and b < 100, return a and b < 100, return a or b < 100, return  
    a < 100 or b < 100)
```

Maks poeng: 2

12 Returner den minste verdien fra *input* listen

```
def min(input):  
    • a = input[0]  
    • for i in input[1:]:  
        ◦ if i < a:  
            ■  (i = a, continue, break, a = i)  
    • return  (input[0], a, i, input[-1])
```

Maks poeng: 2

Du må løse alle tre oppgavene.

Hver oppgave teller med omtrent 25% i vurdering.

- 20% av dette er basert på at programmet ditt fungerer som beskrivelsen i oppgaven tilsier
- 5% deles ut for god programmeringsstil, f.eks gode variabelnavn, bruk av funksjoner hvor det passer, kommentarer og dokumentasjon

13 I løpet av forskningsprosjektet ditt har du samlet inn mange datafiler. Alle filer har .txt format, og de første linjene i hver fil inneholder sted og tidspunkt hvor og når data ble samlet inn. Akkurat nå har filene tilfeldige navn (qwghlm, qwerty, ...), som gjør det vanskelig å finne et datasett uten å åpne alle filene. Du vil gjerne gi bedre navn til filene, basert på informasjonen som finnes i hver fil.

For eksempel, en fil som heter "qwghlm.txt" starter med

- Tromsø
- 2020-05-07
- ... data ...
- ... data ...
- ... data ...

og filen "qwerty.txt" starter med

- Oslo
- 2019-06-01
- ... data ...
- ... data ...
- ... data ...

Skriv et program som

- leser inn hver fil som er nevnt i listen *old_names*
 - tar sted og dato fra de første to linjene
 - lager et nytt filnavn fra dato og sted, med .txt til slutt. Formatet skal være YYYY-MM-DD_PLACENAME.txt
 - lagrer all data fra filen inn i det nye filnavnet

I eksempelet vårt inneholder *old_names* listen ["qwghlm.txt","qwerty.txt"] . Programmet ditt skal gå gjennom listen og lagre en ny fil for hver gammel fil. Innholdet av "qwghlm.txt" skal lagres som "2020-05-07_Tromsø.txt" og innholdet av "qwerty.txt" skal lagres som "2019-06-01_Oslo.txt".

Skriv ditt svar her

1	
---	--

Maks poeng: 25

I spillet "[Rock Paper Scissors](#)" (Stein, saks, papir)

- Rock vinner over Scissors,
- Scissors vinner over Paper og
- Paper vinner over Rock.

Skriv et program som spiller Rock Paper Scissors mot brukeren. Programmet skal telle opp skåren gjennom flere runder.

Datamaskinen skal gjøre tilfeldige valg, ved bruk av *random.randint()* eller *random.choice()* funksjoner. (<https://docs.python.org/3.7/library/random.html#functions-for-integers>, <https://docs.python.org/3.7/library/random.html#functions-for-sequences>)

Bruk *exception handling* for å sjekke at brukerens input er gyldig, og spør en gang til hvis det ikke er det. Programmet skal ikke avbryte med en feilmelding, uansett av hvilken input-streng brukeren skriver.

Slik skal et programkjøringseksempel se ut:

- Let's play round 1
- Your choice (Rock/Paper/Scissors)? *Rock*
- My choicse was Scissors. You win.
- Score: me 0, you 1
- Continue (y/n)? *y*
- Let's play round 2
- Your choice (Rock/Paper/Scissors)? *Paper*
- My choice was Rock. You win.
- Score: me 0, you 2
- Continue (y/n)? *y*
- Let's play round 3
- Your choice (Rock/Paper/Scissors)? *Papp*
- I don't understand Papp. Try again
- Your choice (Rock/Paper/Scissors)? *Paper*
- My choice was Scissors. I win.
- Score: me 1, you 2
- Continue (y/n)? *n*

1	
---	--

Maks poeng: 25

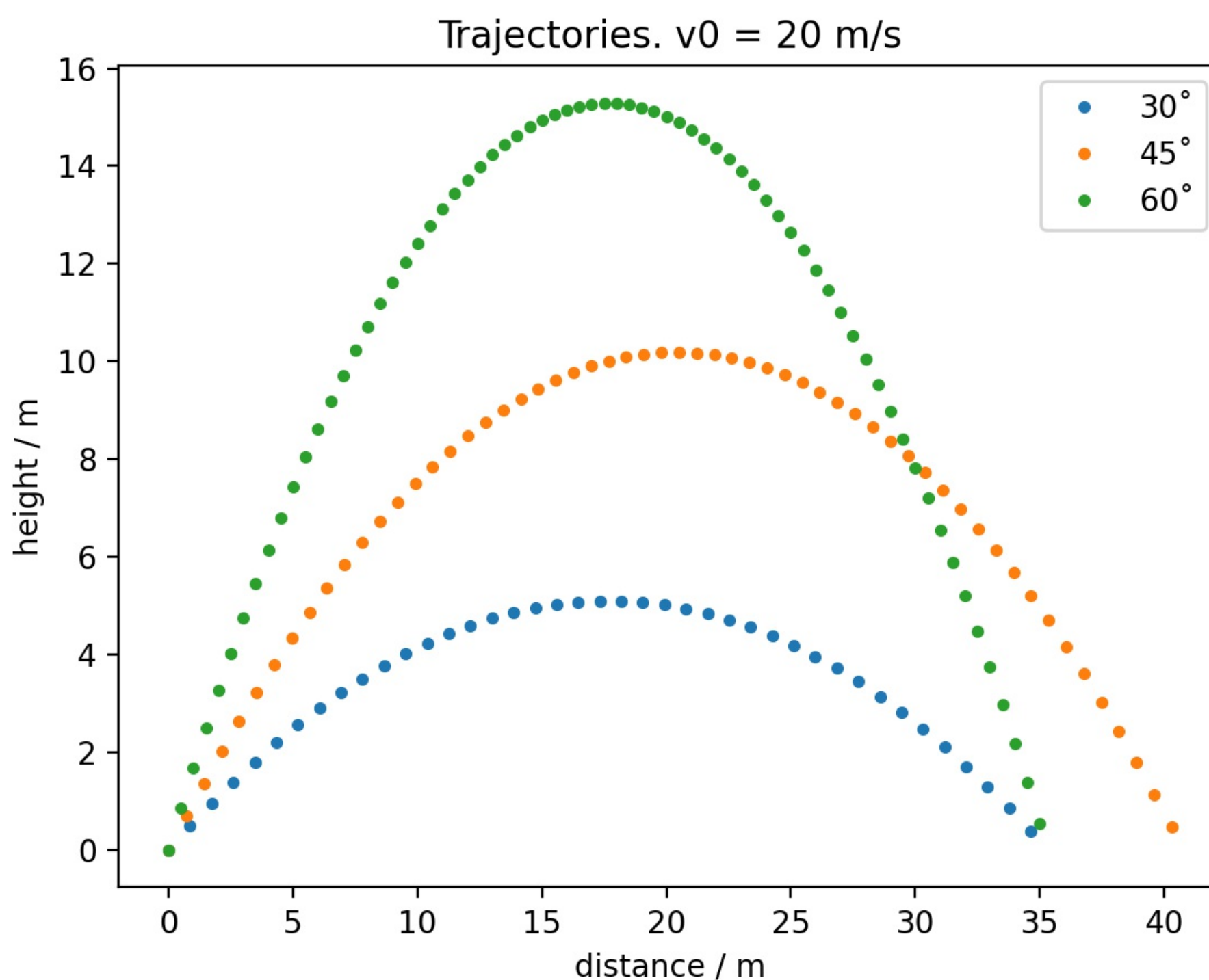
- 15 Når man kaster en ball med vinkelen θ (theta) og fart v_0 uten luftmotstand, så følger ballens flukt en parabel. Posisjonen i x- og y-retning avhenger av tid t som
- $$x(t) = v_0 t \cos \theta$$
- $$y(t) = v_0 t \sin \theta - \frac{1}{2} g t^2$$

I denne oppgaven skal vi bruke

$$g = 9.81 \frac{\text{m}}{\text{s}^2}$$

$$v_0 = 20 \frac{\text{m}}{\text{s}}$$

Vi kan tegne flere baner for ulike vinkler i det samme bildet:



Denne oppgaven kan løses **enten** med numpy arrays **eller** vanlige python lister. Når instruksene nevner "sekvens", kan det være et numpy array eller en python liste.

Lenker til nyttig informasjon finnes etter instruksene.

- Lag en sekvens av tidspunkter fra 0,00 til 5,00 sekunder i skritt av 0,05 s. Det blir 101 verdier her.
- Skriv to funksjoner **calc_x** og **calc_y** som tar en tidssekvens og en vinkel og som returnerer en sekvens av x/y-posisjoner:
 - Begge funksjonene tar to argumenter: en tidssekvens og en vinkel theta
 - calc_x** returnerer en sekvens av x-posjoner i henhold til ligningen $x(t)$ ovenfor
 - calc_y** returnerer en sekvens av y-posjoner i henhold til ligningen $y(t)$ ovenfor
 - sin*- og *cos*-funksjonene og verdien av π er tilgjengelig fra *math* eller *numpy* modulen
- I programmets hoveddelen, skal du skrive en løkke over vinkelverdiene 30, 45 og 60 grader. I løkken skal du
 - konvertere vinkelen fra grader til radianer: $\text{rad} = \frac{\pi}{180} \cdot \text{deg}$
 - bruke **calc_x** og **calc_y** for å få en sekvens av xs og ys
 - finne den delen av sekvensene hvor $y \geq 0$
 - tegne banen ved bruk av *matplotlib.pyplot.plot* med en prikk for hver tidspunkt
- Legg til en fargeforklaring (*legend*), tittel, x- og y-aksetitler. Bildet ditt skal se ut som eksemplene ovenfor
- Til slutt, bruk *matplotlib.pyplot.savefig('trajectory.png')* for å lagre bildet ditt

Nyttige lenker:

<https://numpy.org/devdocs/reference/generated/numpy.linspace.html>
https://matplotlib.org/3.2.1/api/_as_gen/matplotlib.pyplot.plot.html
<https://stackoverflow.com/a/19125863>
<https://pythonprogramming.net/legends-titles-labels-matplotlib-tutorial/>

1	
---	--

Maks poeng: 25