



Exercises – Python Getting Started.

Exercise 1

Hello world.

Lav et program der skriver "Hello World" udNår en variabel du har indlæst er større end 3.

Du skal køre dit program fra Pycharm, men også direkte fra cmd prompt.

Exercise 2

Definition af lister og loops.

Lav en liste med navnene (efternavnet er nok) på dine 5 yndlingsfattere - tildel denne liste til en variabel (f.eks. forfattere =)

Derefter så skriv listen ud med en for løkke sådan at hver forfatter kommer til at stå på sin egen linje.

Modifikation, tilføjelse og sletning fra lister.

Tilføj en forfatter mere med append og skriv den nye liste ud igen.

Derefter så prøv at slette forfatter nr 2, sådan at du igen kun har 5 forfattere på listen. Skriv listen ud igen.

Længden af en liste

Brug len kommandoen til at få længden og på din liste og gem denne værdi i en ny variabel. Udskriv denne variabel.

Reverse

Vend din liste af forfattere om, sådan at den nu bliver udskrevet i omvendt rækkefølge.

Exercise 3

I grupper af 2-3.

Du skal nu lave et program med en Gui i Python.

Bemærk. Dette er **ikke** et kursus i at lave Gui i Python.

Men ->

Ideen med denne opgave er kun at I kort stifter bekendtskab Tkinter, og derefter laver en opgave, hvori du anvender nogle af dagens sprog Python konstruktioner, samt prøver at anvende Tkinter.

Helt overordnet:

Du skal lave et Python program hvori brugeren kan angive et tal, hvor programmet så svarer tilbage om det er primtal eller ej. Let nok?

Jo, jo. Men Python programmet skal også have en brugergrænseflade i tkinter

(Som eksempel på anvendelse af et Python bibliotek).

Hvordan kommer du så i gang med at lave en basal Gui i tkinter?

Jo, der findes et utal af glimrende tutorials på nettet, du kan vælge at følge.

Du kan også vælge at prøve at se på nogle enkle grund begreber, og så på baggrund af det lave din egen gui.

Gennemgå først filen

Gui_Selvstudium.py (Canvas)

Ikke nok? Ok, så kig evt. i filen

Exercise3_ExampleCode (Canvas), således at I forstår hvad der sker. Brug dette som inspiration til jeres eget program.

Og kig så evt. på denne mikro-Python tkinter tutorial (nedenfor) for at komme i gang med Tkinter. Og få eksperimenteret med grundbegreber med tkinter. Brug Google – hvis du sidder fast undervejs.

Men som sagt – du kan frit vælge andre ressourcer fra nettet her.

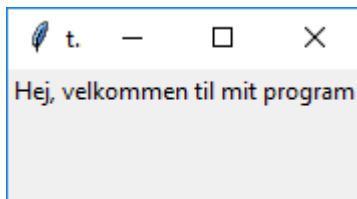
Hovedopgaven her er at få lavet et Python program, med tkinter gui, der kan løse et simpelt problem med at afgøre om et tal er et primtal eller ej.

Basale begreber i tkinter – en mikro rundtur.

Lav et tkinter program som viser følgende string:

Hej, velkommen til mit program\n\n\n

Dit UI skal se sådan her ud:



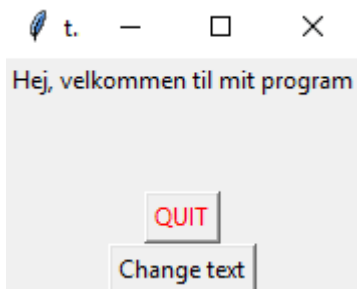
Derefter:

Du skal indsættes en Quit knap som har rød tekst.

Derudover skal du have en "Change Text" knap med almindelig tekst - denne knap skal være under den anden knap.

Begge knapper skal centreres (default faktisk)

Dit UI skal se sådan her ud:



Bemærk, at vi nu har flere UI elementer, så derfor kan det være en fordel at bruge en Frame til at "pakke" det hele sammen. Brug Google for yderligere forklaring.

Det vil sige at du skal ændre din tekst UI og dine knapper til at referere til en Frame i stedet for root. Vi kan nøjes med 1 frame her. Så din tekst fra opgave 1 skal bruge en frame

#Her laver vi en frame og packer den ned i root vinduet.

```
frame = tk.Frame(root)
frame.pack()
```

#nu skal label referere til frame

```
w = tk.Label(frame, text="Hej, velkommen til mit
program\n\n\n")
w.pack()
```

Så dine 2 knapper skal også pakkes ned i frame og ikke root nu.

Dine to knapper skal også referere til en funktion, som bliver udført når du trykker på knappen.

Det gør du ved at sætte command til et funktionsnavn:

```
command=quit
```

quit er faktisk prædefineret - så der skal du IKKE definere din egen funktion.. Men den anden knap skal have en funktion defineret (command=change):

```
def change():  
    pass
```

#pass betyder at funktionen ikke gør noget!

Du skal nu ændre change() funktionen (dvs. er erstat pass med andet kode) sådan at den erstatter den label vi allerede har "Hej velkommen til...." til følgende tekst, når vi trykker på knappen : "Du har trykket på change knappen!\n"

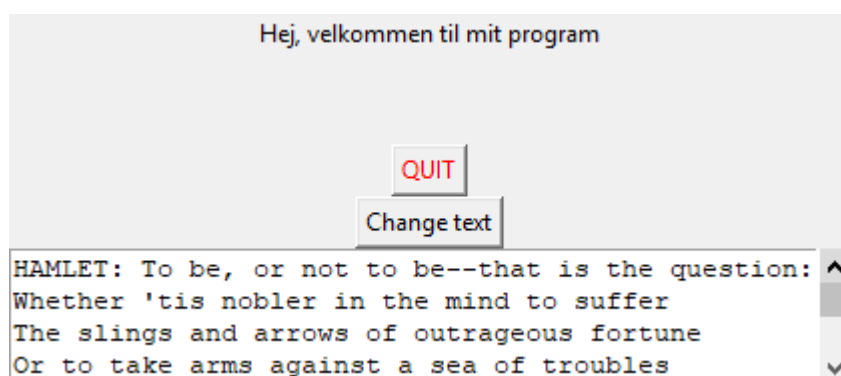
Læg mærke til hvad der sker med placeringen af knapperne efter du har trykket på din change knap?

Derefter:

Text widget

Text widget er en mere sofistikeret udgave af label - vi kan f.eks. indsætte flere linjer, indsætte billeder og/eller html styling af vores tekst.

Dit UI bør se ca. sådan her ud til sidst:



Bemærk - i din kode vil du have dette her:

```
S.pack(side=tk.RIGHT, fill=tk.Y)
```

```
T.pack(side=tk.LEFT, fill=tk.Y)
```

Normal vil pack sætte UI elementer under hinanden, ligesom med knapper

tidligere, men nu siger vi at scrollbaren (S) skal være til højre og Text (T) skal være til venstre (og så i øvrigt fylde ud sådan at de er lige høje - det sørger `fill=tk.Y` for).

Derefter:

Entry widget er et input felt hvor brugeren kan indtaste noget tekst, som vi så kan læse og bruge i vores program.

Reference guide gør det lidt anderledes - den bruger grid manager og ikke pack manager (se sidste opgave), så du får her noget hjælpe kode til at lave et input felt:

#bemærk at vi nu laver en ny frame - fordi vi skal have noget UNDER den oprindelige frame

```
frame2 = tk.Frame(root)
frame2.pack()
```

```
label2 = tk.Label(frame2, text="Indtast navn:")
```

```
#vi angiver padx og pady for at få lidt plads
```

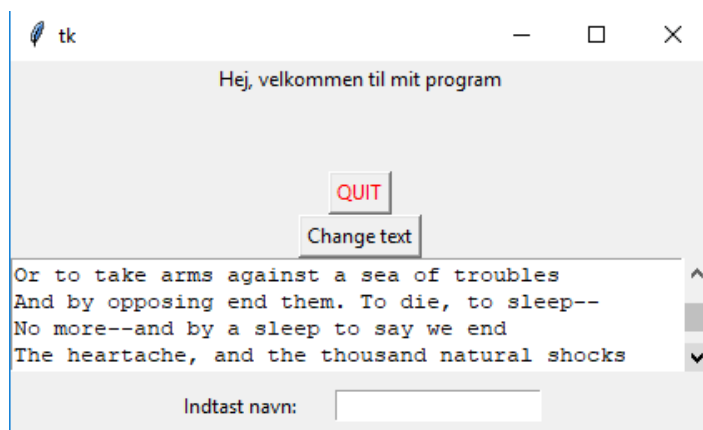
```
label2.pack(side=tk.LEFT, padx= 10, pady=10)
```

```
#så skal vi have vores input felt ind
```

```
entry = tk.Entry(frame2)
```

```
entry.pack(side=tk.LEFT, padx= 10, pady=10)
```

Du bør få dit UI til at ligne ca. dette her:



Nu skal vi så bruge input fra brugeren til noget.

Så du skal modificere din `change()` funktion, sådan at når vi nu trykker på change tekst knappen, så skal du ændre din overskrifts label til det som brugeren har indtastet.

Pt ser din change funktion sådan her ud:

```
def change() :
```

```
label.config(text=str("Du har trykket på change  
knappen! \n"))
```

Vi kan nemt få teksten som brugeren har tastet ind. Vores input felt hedder jo "entry" i dette tilfælde så følgende kode vil hente den tekst, som brugeren har indtastet:

```
input = entry.get()
```

Så skal du bare have ændret label til den nye tekst som inputtet har.

Derefter:

Layout Management

Der er 3 forskellige måder at lave layouts på i Tkinter - dvs. der er 3 forskellige layout managers:

- pack
- grid
- place

Her vil vi kun bruge pack.

Prøv at løbe den kode du indtil nu har set på igennem, og prøv at forstå hvorfor pack placerer din UI som den gør. Google for yderligere forklaringer.