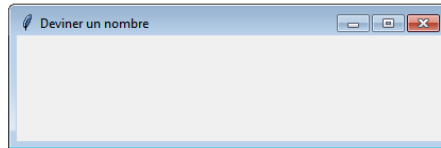


1. Création de la fenêtre deviner un nombre :

On souhaite reprendre le programme qui consiste à deviner un nombre mais cette fois ci en utilisant une interface graphique.

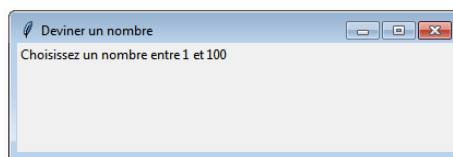
Dans un répertoire **TP4** créer un fichier Python : **PGM1.py**

1.1. Créer une fenêtre (nommée **fenetre** dans le programme) ayant pour titre **Deviner un nombre**



1.2. Ajouter le texte : **Choisissez un nombre compris entre 1 et 100**. Et dimensionner la taille de la fenêtre à 400x100 pixels en utilisant l'instruction `fenetre.geometry('450x100')`

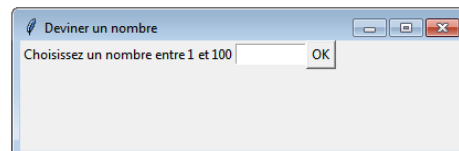
```
lab1 = Label(fenetre, text = "...")  
lab1.grid(row = 0, column = 0)
```



1.3. Ajouter une zone de saisie ainsi qu'un bouton **ok** pour valider le **nombre** saisi.

Attention : Le **nombre** saisi sera de type **StringVar()**

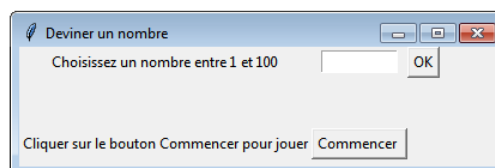
L'action sur le bouton **ok** entraine l'appel d'une nouvelle fonction que l'on nommera : **comparer**



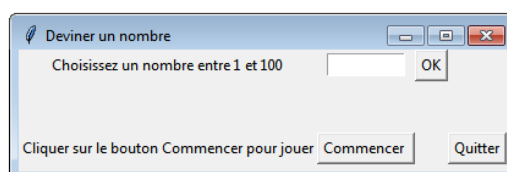
1.4. Vous définirez 2 étiquettes **lab2** et **lab3** sous "Choisissez un nombre entre 1 et 100" sans texte pour le moment (`text=""`) dans lesquelles on communiquera des indications au joueur durant la partie

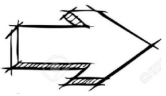
1.5. Ajouter ensuite en bas et à gauche de la fenêtre une instruction destinée commencer la partie.

L'action sur le bouton **Commencer** entraine l'appel d'une fonction que l'on nommera **init**

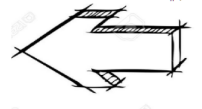


1.6. Ajouter enfin un bouton permettant de quitter le jeu et de détruire/fermer la fenêtre.





Passons maintenant à la définition des fonctions **comparer** et **init**

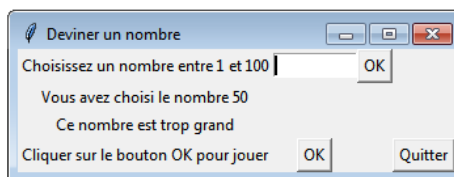


1.7. La fonction **init** permet :

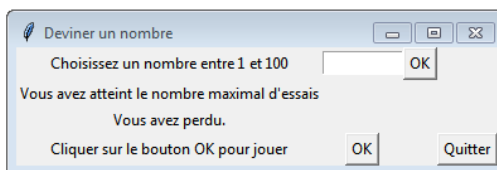
- de choisir le nombre aléatoire que l'on nommera **nba** et qui sera de type **IntVar()** variable déclarée au préalable c'est-à-dire en dehors de la fonction **init**.
- d'afficher dans l'interpréteur Python la valeur de **nba** grâce à l'instruction `print(nba)` et ce pour nous aider à mettre au point le programme.

1.8. La fonction **comparer** permet :

- de comparer (*on s'en serait douté*) le nombre saisi par le joueur **nombre** au nombre aléatoire **nba**.
- de « guider » le joueur en lui fournissant des indications "Vous avez choisi le nombre", "Ce nombre est trop petit", "Ce nombre est trop grand" ou "Bravo, vous avez gagné"



Ou encore : Vous avez atteint le nombre maximal d'essais...



2. La balle rebondissante :

2.1. Toujours dans le répertoire **TP4** créer un fichier Python : **PGM2.py** dans lequel vous copiez/collez le code suivant.

Complétez les pointillés puis exécutez le programme.

```
from tkinter import *
Largeur = .....
Hauteur = .....
root = Tk()
canvas = Canvas(root, width=Largeur, height=Hauteur, background="white")
canvas.pack(fill="both", expand=True)
```

2.2. Vous ajouterez à la suite, le code suivant.

```
# Les coordonnées de départ de la balle et sa vitesse
x0, y0 = 10,10
dx = +12 # Vitesse horizontale
dy = +5 # Vitesse verticale
```

2.3. Rajoutez le code décrivant la balle.

```
# La balle à déplacer
balle = canvas.create_oval(x0,y0,x0+20,y0+20,width=2,fill="red")
```



A quoi correspondent les différentes valeurs indiquées par les flèches, modifier ces valeurs et observez le résultat après exécution du programme.

2.4. On s'intéresse maintenant au déplacement et aux trajectoires de la balle.

On crée une fonction **deplacer**.

Vous ajouterez ce code et le complétez

```
def deplacer():
    .....
    x, y = canvas.coords(balle)[:2]
    x0 = x + dx # Nouvelle abscisse
    y0 = y + dy # Nouvelle ordonnée
    canvas.coords(balle,x0,y0,x0+20,y0+20) # Déplacement

    if .....<0 or .....> Largeur:
        dx = .... # Changement de sens horizontal
    if .....<0 or .....> Hauteur:
        dy = .... # Changement de sens vertical

    canvas.after(50,deplacer) # Appel après 50 millisecondes
    return

deplacer()
root.mainloop()
```