Réalisation d'une calculatrice en python avec tkinter
Explications du programme calculatrice_01.py
Amélioration du programme → calculatrice_02.py

Le programme calculatrice_01.py contient quelques détails qui méritent d'être éclaircis

Table des matières

1. Que signifie tk.END ?....1
a. Exemple avec Entry (champ de saisie)....2
Exemple 1 : Insérer du texte à la fin....2
Exemple 2 : Supprimer tout le contenu....2
Exemple 3 : Récupérer tout le contenu....2
b. Et dans un widget Text ?....2
c. À noter :....2
2. Comment fonctionne la fonction calculer() qui évalue le résultat ?....2
a. Contexte....2
b. Code de la fonction calculer()....2
c. Détail ligne par ligne....3
resultat = eval(affichage.get())....3
affichage.delete(0, tk.END)....3
affichage.insert(tk.END, str(resultat))....3
except Exception: (en cas d'erreur)....3
3. Amélioration possible : calculatrice_02.py....4
a. Modification de l'interface graphique pour ajouter le bouton '<'....4
b. Ajout de la fonction press_retour() associée au bouton '<'....4
Comment utiliser affichage.delete() pour supprimer le dernier caractère....4

1. Que signifie tk.END ?

La constante **`tk.END`** (ou `END` tout court si on importe `from tkinter import *`) est utilisée dans **Tkinter** pour indiquer **la fin du contenu** d'un **widget textuel** — comme un champ `Entry` ou `Text`. Noter la syntaxe utilisée :

- END en majuscule pour indiquer une constante ;
- tk. pour pour indiquer que cette constante est contenu dans le module tk

`tk.END` représente **la dernière position possible** dans un widget textuel, généralement utilisée pour :

1. **Ajouter du texte à la fin**
2. **Supprimer du texte jusqu'à la fin**
3. **Lire du texte jusqu'à la fin**

a. Exemple avec `Entry` (champ de saisie)

Exemple 1 : Insérer du texte à la fin

```
entry.insert(tk.END, "Bonjour")
```

👉 Cela ajoute le texte `"Bonjour"` **après tout ce qui est déjà présent** dans le champ.

Exemple 2 : Supprimer tout le contenu

```
entry.delete(0, tk.END)
```

- `0` = position de départ (début du champ)
- `tk.END` = position de fin

👉 Cela efface **tout le texte** dans le widget entry.

Exemple 3 : Récupérer tout le contenu

```
saisie = entry.get(0, tk.END)
```

- `0` = position de départ (début du champ)
- `tk.END` = position de fin

👉 Récupère **tout le texte** contenu dans le widget entry et le place dans la variable saisie. Pour récupérer tout le texte on peut aussi se contenter de `saisie = entry.get()` , les valeurs 0 et tk.END sont les valeurs par défaut de ces deux paramètres.

b. Et dans un widget `Text` ?

Dans un widget multi-ligne `Text`, `tk.END` correspond à **la fin du texte + 1 caractère**, souvent une nouvelle ligne automatique. Exemple :

```
text_widget.insert(tk.END, "Ligne ajoutée")
```

c. À noter :

- `tk.END` est une **chaîne de caractères** (type str), équivalente à `"end"`.
- Tu peux aussi écrire `"end"` à la place, mais `tk.END` est plus clair et recommandé pour éviter les erreurs de frappe.

2. Comment fonctionne la fonction calculer() qui évalue le résultat ?

a. Contexte

La fonction `calculate()` est appelée quand l'utilisateur appuie sur le bouton =.

b. Code de la fonction `calculer()`

```
def calculer():
    try:
```

```
        resultat = eval(affichage.get())
        affichage.delete(0, tk.END)
        affichage.insert(tk.END, str(resultat))
    except Exception:
        affichage.delete(0, tk.END)
        affichage.insert(tk.END, "Erreur")
```

c. Détail ligne par ligne

`resultat = eval(affichage.get())`

- `affichage.get()`
  ➤ Récupère **le texte contenu dans le champ `Entry`**, c'est-à-dire l'expression mathématique que l'utilisateur a saisie.
  - Exemple : `"2+3*4"`
- `eval(...)`
  ➤ La fonction **Python intégrée `eval()` évalue une chaîne de caractères comme une expression Python**.
  - `"2+3*4"` devient **`14`** (car `3*4 = 12`, puis `2+12 = 14`).
- `resultat = ...`
  ➤ Le résultat du calcul est stocké dans la variable `resultat`.

⚠️ **Attention avec `eval()`** : elle exécute du code Python arbitraire. C'est pourquoi elle ne doit **jamais être utilisée avec des données non fiables ou externes**. Dans une calculatrice locale pour débutant, c'est acceptable, mais pas dans un programme en production.

`affichage.delete(0, tk.END)`

➤ Efface **tout le texte** dans le champ `Entry`, du caractère à la position `0` jusqu'à la fin (`tk.END`).

`affichage.insert(tk.END, str(resultat))`

➤ Affiche le **résultat du calcul** dans le champ `Entry`.

- `str(resultat)` convertit le résultat (par exemple `14`) en chaîne `"14"`.
- `tk.END` indique qu'on veut insérer le texte **à la fin** du champ (ici, il est vide après l'effacement, donc ça revient à l'insérer au début).

`except Exception:` (en cas d'erreur)

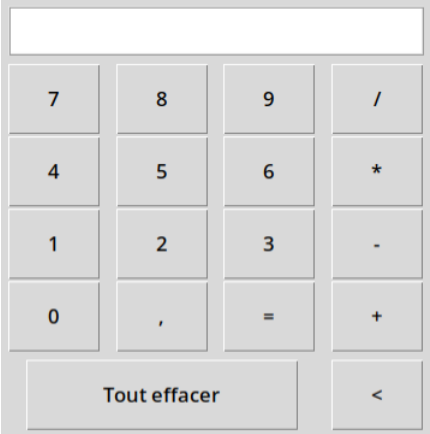
- Si une **erreur** survient pendant `eval()` (par exemple si l'utilisateur tape `2+*3`), Python lèvera une exception.

- Dans ce cas :
    1. Le champ est vidé.
    2. Le message `"Erreur"` est affiché dans le champ.

3. Amélioration possible : calculatrice_02.py

Objectif : ajouter un bouton '<' pour effacer le dernier caractère entré.

a. Modification de l'interface graphique pour ajouter le bouton '<'



Ce nouveau bouton sera placé sur la dernière ligne, à droite du bouton 'Tout effacer'. Enregistrer le programme calculatrice_01.py sous calculatrice_02.py puis effectuer les modifications nécessaires pour obtenir ce rendu visuel.

b. Ajout de la fonction press_retour() associée au bouton '<'

Cette fonction doit supprimer le dernier caractère entré. Cette fonction fera appel à la méthode delete() appliquée à l'objet affichage. Voyons la syntaxe associée à cette méthode.

Comment utiliser affichage.delete() pour supprimer le dernier caractère

Syntaxe générale :

```
affichage.delete(position_debut, position_fin)
```

- Les positions sont des **indices de caractères**, où `0` est le premier.

Pour supprimer **le dernier caractère**, il faut cibler la **position de l'avant-dernier caractère** jusqu'à la **fin**. Pour cela nous utiliserons la méthode index pour rechercher l'indice correspondant à tk.END dans la chaîne de caractère associée à l'objet affichage. Nous soustrairons 1 à cet indice pour avoir l'indice du dernier caractère visible.

```
indice_dernier_caractère_visible = affichage.index(tk.END) - 1
affichage.delete(indice_dernier_caractère_visible, tk.END)
```