

Fiche mémo — Tkinter : Raccourcis clavier & gestion des événements clavier

1. Ajouter un raccourci clavier (affichage dans le menu)

Sert uniquement à *afficher le texte du raccourci dans le menu*

Il ne rend pas le raccourci fonctionnel à lui seul

```
menu_fichier.add_command(  
    label="Quitter",  
    accelerator="Ctrl+Q",    # Texte affiché à droite du menu  
    command=fenetre.destroy  
)
```

Affichage :

Quitter Ctrl+Q

2. Activer réellement le raccourci (bind_all / bind)

Associe une combinaison de touches à une action Python

bind_all → actif dans toute l'application

bind → actif seulement dans un widget donné

```
fenetre.bind_all("<Control-q>", lambda e: fenetre.destroy()) # Quitter  
fenetre.bind_all("<F1>", lambda e: afficher_aide())           # Ouvrir l'aide
```

3. Principales syntaxes de touches

Syntaxe	Action
<Control-q>	Ctrl + Q
<Control-Shift-s>	Ctrl + Maj + S
<Alt-F4>	Alt + F4
<F1> à <F12>	Touches de fonction
<Escape>	Échap
<Return>	Entrée

Syntaxe	Action
<space>	Barre d'espace
<Tab>	Tabulation
<BackSpace>	Retour arrière
<Delete>	Suppr
<Up>, <Down>, <Left>, <Right>	Flèches directionnelles
<KeyPress-a>	Touche "a" (minuscule)
<KeyRelease-A>	Relâchement de "A"

4. L'objet "event" (e)

Quand on lie un raccourci, la fonction appelée reçoit un **objet event** contenant des infos sur la touche pressée.

```
def touche_appuyee(e):
    print(f"Touche : {e.keysym} | Code : {e.keycode}")

fenetre.bind_all("<Key>", touche_appuyee)
```

Exemple de sortie :

```
Touche : a | Code : 65
```

5. Liens utiles et astuces

- **Combiner affichage + action :**

```
menu_fichier.add_command(label="Quitter", accelerator="Ctrl+Q",
    command=fenetre.destroy)
fenetre.bind_all("<Control-q>", lambda e: fenetre.destroy())
```

-> cohérence parfaite entre menu et raccourci.

- **Bloquer un raccourci sur un widget précis :**

```
entree.bind("<Return>", lambda e: convertir())
```

Appuyer sur "Entrée" lance la conversion sans cliquer sur le bouton.

- **Désactiver un raccourci temporairement :**

```
fenetre.unbind_all("<Control-q>")
```

6. Bonnes pratiques

- Toujours afficher les raccourcis clavier dans les menus (**accelerator=**).
- Grouper les **bind_all** dans une section dédiée, après la création de la fenêtre.
- Utiliser **lambda e**: quand on n'a pas besoin du paramètre **event**.
- Pour les fonctions complexes → écrire une vraie fonction avec (**event**) en argument.

Exemple complet

```
# Menu
menu_fichier.add_command(label="Quitter", accelerator="Ctrl+Q",
command=fenetre.destroy)
menu_aide.add_command(label="Aide", accelerator="F1", command=afficher_aide)

# Bindings
fenetre.bind_all("<Control-q>", lambda e: fenetre.destroy())
fenetre.bind_all("<F1>", lambda e: afficher_aide())
fenetre.bind_all("<Control-e>", lambda e: effacer())
fenetre.bind_all("<Control-r>", lambda e: convertir())
```

Astuce bonus pour le convertisseur : on pourrait afficher un petit texte d'aide du type :

"Astuce : utilisez **Ctrl+R** pour convertir rapidement ou **F1** pour afficher l'aide."

...dans la zone de messages ou dans le panneau de contexte, pour un rendu encore plus pédagogique