

## 2N6 Programmation 2



avec python<sup>TM</sup>

- GUI (suite)



- > On a vu les:
  - > Les **Frames** pour structurer l'interface.
  - > Cinq widgets :
    - > **Label** pour afficher de l'information.
    - > **Entry** pour saisir une courte chaîne de caractères.
    - > **TextBox** pour entrer/afficher plus d'information.
    - > **Combobox** (liste déroulante) pour permettre de faire des choix parmi une liste.
    - > **Button** pour interagir avec l'interface.

# Aujourd'hui :



- On va voir trois nouveaux widgets :
  - **CheckBox** pour faire des choix True / False
  - **RadioButton** pour sélectionner une option parmi plusieurs.
  - **Image** pour communiquer plus d'informations et rendre le GUI plus beau.



# CheckBox (case à cocher)

- > Les cases à cocher permettent de faire une sélection booléenne, vrai ou faux

```
self.chk_1 = ctk.CTkCheckBox(master=frm_experience,  
.....text="Programmation Python",  
.....onvalue=1, offvalue=0)
```

- > onvalue : lorsque la case est cochée,
- > offvalue : lorsque la case n'est pas cochée
- > On obtient la valeur (onvalue ou offvalue) avec la méthode .get()

```
valeur_chkbox = self.chk_1.get()
```

<input type="checkbox"/>	Programmation Python
<input type="checkbox"/>	Programmation PowerShell
<input type="checkbox"/>	Maintenance des serveurs
<input type="checkbox"/>	Maintien des postes de travail
<input type="checkbox"/>	Maintien du réseau câblé et sans fil
<input type="checkbox"/>	Maintien des autres périphériques
<input type="checkbox"/>	Gestion des licences
<input type="checkbox"/>	Formation usagers
<input type="checkbox"/>	Gestion des accès usagers
<input type="checkbox"/>	Gestion des anti-virus
<input type="checkbox"/>	Gestion des télécommunications
<input type="checkbox"/>	Gestion des routeurs



# RadioButton (boutons radio)

- Permettent de choisir entre plusieurs options.
- Les groupes de boutons radio sont définis par la variable à laquelle ils réfèrent.

```
choix_pizza = tk.StringVar(value="nature")
```

```
pizza_nature = ttk.Radiobutton(frm_pizza_choix, text='Nature', variable=choix_pizza, value='nature')  
pizza_vege = ttk.Radiobutton(frm_pizza_choix, text='Végétarienne', variable=choix_pizza, value='végétarienne')  
pizza_garnie = ttk.Radiobutton(frm_pizza_choix, text='Toute garnie', variable=choix_pizza, value='toute garnie')
```

- On obtient la valeur du groupe de boutons radios à partir de la variable à laquelle ils réfèrent grâce à la méthode .get()

```
choix_pizza_val = choix_pizza.get()
```

# RadioButton (boutons radio)



```
choix_pizza = tk.StringVar(value="nature")
```

Classe importée de tkinter. Elle est similaire à la classe "str" mais fonctionne dans les widgets et possède la méthode .get()

```
pizza_nature = ttk.Radiobutton(frm_pizza_choix, text='Nature', variable=choix_pizza, value='nature')  
pizza_vege = ttk.Radiobutton(frm_pizza_choix, text='Végétarienne', variable=choix_pizza, value='végétarienne')  
pizza_garnie = ttk.Radiobutton(frm_pizza_choix, text='Toute garnie', variable=choix_pizza, value='toute garnie')
```

La même variable → ces boutons font partie du même groupe.  
Un seul RadioButton peut être sélectionné à la fois.

```
choix_pizza_val = choix_pizza.get()
```

La méthode retourne la valeur sous forme de "str" qui peut ensuite être utilisé.

# Images



- > Besoin d'un autre module :

```
!> pip install pillow
```

```
from PIL import ImageTk, Image
```

- > Permet d'ajouter des images dans des labels ou dans des boutons.
- > Par standard, les images sont situées dans un répertoire "images" dans le même emplacement que notre script.

```
R25_Ex2_MyOrder
├── R25_YourOrder.py
└── images
    ├── Logo.jpg
    ├── pizza.jpg
    ├── poutine.jpg
    └── sousmarin.jpg
```

# Images



- Par standard, les images sont situées dans un répertoire "images".
- On va chercher dans l'emplacement de l'image avec le module os.

```
R25_Ex2_MyOrder
├── R25_YourOrder.py
└── images
    ├── Logo.jpg
    ├── pizza.jpg
    ├── poutine.jpg
    └── sousmarin.jpg
```

```
self.image_path = os.path.join(os.path.dirname(os.path.realpath(__file__)), "images")
```

```
self.logo_image = ImageTk.PhotoImage(Image.open(os.path.join(self.image_path, "logomatissoft.jpg")))
```

- On peut ensuite récupérer l'image et la mettre dans un objet. Qu'on utilisera dans un label ou un bouton.

```
# lbl pour le logo de l'entreprise
```

```
self.lbl_logo = tk.Label(master=frm_container, image=self.logo_image)
```