



Bienvenue dans le monde fantastique où ton ordinateur **te parle** (et ne te juge pas... normalement). Aujourd'hui, tu vas apprendre à créer un programme capable de dialoguer avec un humain. Ton ordinateur deviendra peut-être le prochain meilleur ami de l'humanité, ou juste ton assistant de recherche YouTube. Let's go !

1. Bases du Dialogue Machine/Humain

Pour que la machine comprenne ce que tu veux, il faut établir un langage commun (pas encore le langage des dauphins, on reste sur du texte).

On va se baser sur trois étapes clés :

1. **Compréhension de la phrase** : La machine doit analyser ce que tu dis.
 - o Extraire des mots-clés importants.
 - o Identifier des intentions (par exemple, "cherche", "joue", "salut").
2. **Génération de réponse** : Une fois l'intention comprise, la machine te répond intelligemment (ou tente).
3. **Traitement des actions** : En fonction de l'intention, le programme effectue une action spécifique, comme lancer une recherche sur Google ou jouer une vidéo YouTube.

2. Premiers Pas : La Compréhension des Intentions

Pour commencer, un programme basique qui comprend **trois types de phrases** : un bonjour, une recherche Google, et une commande YouTube.

Voici un exemple minimaliste :

```
def comprendre_intention(phrase):
    """Analyse la phrase et retourne une intention."""
    if "bonjour" in phrase.lower():
        return "salut"
    elif "cherche" in phrase.lower():
        return "recherche"
    elif "youtube" in phrase.lower():
        return "youtube"
    else:
        return "inconnu"
```

Testons :

```
phrase = input("Que veux-tu dire à l'ordinateur ? ")
intention = comprendre_intention(phrase)
print(f'L'ordinateur a compris : {intention}')
```

3. Génération d'une Réponse

Maintenant que la machine a identifié une intention, il faut qu'elle te réponde (de façon sympa).

```
def generer_reponse(intention):
    """Génère une réponse adaptée à l'intention."""
    if intention == "salut":
        return "Salut ! Comment puis-je t'aider ?"
    elif intention == "recherche":
        return "Dis-moi ce que tu veux chercher sur Google !"
```

```

    elif intention == "youtube":
        return "Quelle vidéo veux-tu regarder sur YouTube ?"
    else:
        return "Je n'ai pas compris... Peux-tu reformuler ?"

```

On combine tout ça :

```

phrase = input("Que veux-tu dire à l'ordinateur ? ")
intention = comprendre_intention(phrase)
reponse = generer_reponse(intention)
print(reponse)

```

4. Actions : Recherche sur le Net et Vidéo YouTube

Ajoutons un peu de magie avec des commandes externes :

- **Recherche Google** : On utilise le module `webbrowser` (à installer avant ;)).
- **Lancer une vidéo YouTube** : Même principe, mais en construisant l'URL YouTube.

```

import webbrowser

def traiter_action(intention, phrase):
    """Effectue une action en fonction de l'intention."""
    if intention == "recherche":
        mots_cles = phrase.replace("cherche", "").strip()
        url = f"https://www.google.com/search?q={mots_cles}"
        webbrowser.open(url)
        return f"Recherche Google lancée pour : {mots_cles}"
    elif intention == "youtube":
        mots_cles = phrase.replace("youtube", "").strip()
        url = f"https://www.youtube.com/results?search_query={mots_cles}"
        webbrowser.open(url)
        return f"Recherche YouTube lancée pour : {mots_cles}"
    else:
        return "Aucune action disponible."

```

Testons le tout :

```

phrase = input("Que veux-tu dire à l'ordinateur ? ")
intention = comprendre_intention(phrase)
reponse = generer_reponse(intention)
print(reponse)

if intention in ["recherche", "youtube"]:
    resultat = traiter_action(intention, phrase)
    print(resultat)

```

5. Enrichir la Bibliothèque Automatiquement

Pour rendre la machine plus intelligente, elle doit apprendre des nouvelles phrases. Créons un petit mécanisme d'apprentissage.

```

import json

# Charger ou créer une base de données
try:
    with open("base_de_donnees.json", "r") as f:
        base_de_donnees = json.load(f)

```

```

except FileNotFoundError:
    base_de_donnees = {}

def ajouter_intention(phrase, intention):
    """Ajoute une intention à la base de données."""
    base_de_donnees[phrase] = intention
    with open("base_de_donnees.json", "w") as f:
        json.dump(base_de_donnees, f)
    print(f"Phrase ajoutée : {phrase} -> {intention}")

# Exemple : Apprentissage manuel
ajouter_intention("lance une recherche", "recherche")

```

Avec cette méthode, ton programme peut apprendre de nouvelles intentions grâce à l'utilisateur.

6. Créer son propre module

Pour structurer un programme, il est utile de créer des **modules**. Ici, tu vas apprendre à transformer tes fonctions en module réutilisable. 

6.1. Pourquoi créer un module ?

- Organiser le code (pas tout dans un seul fichier).
- Réutiliser les fonctionnalités dans d'autres projets.
- Faciliter la maintenance.

6.2. Création d'un module "assistant_dialogue"

1. **Créer un fichier assistant_dialogue.py** (ça sera le nom de ton module) dans ton projet.

2. **Déplacer les fonctions dans ce fichier :**

```

# assistant_dialogue.py

import json
import webbrowser

def comprendre_intention(phrase):
    """Analyse la phrase et retourne une intention."""
    if "bonjour" in phrase.lower():
        return "salut"
    elif "cherche" in phrase.lower():
        return "recherche"
    elif "youtube" in phrase.lower():
        return "youtube"
    else:
        return phrase

def recherche(intention):
    phrase = input("Dis-moi ce que tu veux chercher sur Google !")
    return traiter_action(intention, phrase)

def video(intention):
    phrase = input("Quelle vidéo veux-tu regarder sur YouTube ?")
    return traiter_action(intention, phrase)

def generer_reponse(intention):

```

```

"""Génère une réponse adaptée à l'intention."""
if intention == "salut":
    return "Salut ! Comment puis-je t'aider ?"
elif intention == "recherche":
    return recherche(intention)
elif intention == "youtube":
    return video(intention)
else:
    base = charger_base_de_donnees()
    return apprendre_intention(base, intention)

def traiter_action(intention, phrase):
    """Effectue une action en fonction de l'intention."""
    if intention == "recherche":
        mots_cles = phrase.replace("recherche", "").strip()
        url = f"https://www.google.com/search?q={mots_cles}"
        webbrowser.open(url)
        return f"Recherche Google lancée pour : {mots_cles}"
    elif intention == "youtube":
        mots_cles = phrase.replace("youtube", "").strip()
        url = f"https://www.youtube.com/results?search_query={mots_cles}"
        webbrowser.open(url)
        return f"Recherche YouTube lancée pour : {mots_cles}"
    else:
        return "Aucune action disponible."

def charger_base_de_donnees(nom_fichier="base_de_donnees.json"):
    """Charge une base JSON existante ou en crée une vide."""
    try:
        with open(nom_fichier, "r") as f:
            base = json.load(f)
    except FileNotFoundError:
        base = {}
    return base

def sauvegarder_base_de_donnees(base, nom_fichier="base_de_donnees.json"):
    """Sauvegarde une base JSON."""
    with open(nom_fichier, "w") as f:
        json.dump(base, f, indent=4)

def ajouter_intention(base, phrase, intention):
    """Ajoute une nouvelle phrase et intention à la base."""
    base[phrase] = intention
    print(f"Ajouté : {phrase} -> {intention}")
    return base

def apprendre_intention(base, phrase):
    """Propose d'apprendre une intention si la phrase est inconnue."""
    if phrase not in base:
        print("Je ne connais pas cette phrase.")
        print(phrase)
        intention = input(f"Quelle intention associer à '{phrase}' ? ")
        base = ajouter_intention(base, phrase, intention)

```

```
sauvegarder_base_de_donnees(base)
else:
    print(f"Phrase déjà connue : '{phrase}' -> {base[phrase]}")
return base
```

6.3. Utiliser le module dans un autre script

Crée un nouveau fichier `main.py` pour tester ton module.

```
import assistant_dialogue as ad

# Charger la base
base = ad.charger_base_de_donnees()

while True:
    try:
        phrase = input("Que veux-tu dire à l'ordinateur ? ")
        intention = ad.comprendre_intention(phrase)
        reponse = ad.generer_reponse(intention)
        print(reponse)
    except:
        print("problème")
```

7. TP Final : Mon Premier Assistant Virtuel

Objectif : Créer un programme capable de répondre à des demandes courantes :

1. Dire bonjour et demander comment ça va, avec une réponse adéquate.
2. Proposer à l'utilisateur ce qu'il veut faire et enrichir la liste de ce qu'il peut faire (pour l'instant recherche internet et vidéo YouTube).

Énoncé TP :

- Implémente les fonctions `comprendre_intention`, `generer_reponse`, et `traiter_action`.
- Ajoute une fonction `ajouter_intention` qui permet d'enrichir la base de données.
- Propose des tests pour voir si ton assistant apprend bien au fil du temps.