

Poké-Bot

Version 3.7



DIVERSITY
by EPITECH

I. Introduction

Pour la première fois depuis des années, l'hiver atteint des températures particulièrement basses dans le monde Pokémon. C'est pour cette raison que les habitants de Pokéville sont obligés de rester enfermés chez eux et de flâner au coin du feu afin d'éviter de finir complètement congelés ! Cependant, même s'ils sont au chaud et en sécurité, l'ennui commence à se faire sentir...

Le maire connaît un outil qui pourrait leur permettre de communiquer entre eux et de s'occuper par la même occasion : Discord. Il sait aussi qu'il est possible de créer des robots en Python, mais n'y connaissant rien à la programmation, il requiert vos compétences de développeur. Il souhaite mettre en place beaucoup de fonctionnalités en très peu de temps et connaissant votre détermination, il vous fait entièrement confiance et sait que vous serez prêt à tout pour obtenir rapidement de bons résultats.

Etant donné qu'il fait froid, vous vous dites que choisir un Pokémon de type feu permettrait aux gens de se réchauffer ! Mais les gens aiment aussi ce qui est mignon, votre choix s'oriente donc vers *Salamèche*.

Votre outil principal sera la documentation pour les bots en Python qui se trouve sur [ce site](#). Pour le bon déroulé de ce document, il vous faudra installer Python ainsi que les dépendances nécessaires précisées dans le document d'installation de Python et ses outils. Il vous faudra aussi installer FFmpeg, voir le document dédié à ce propos.

II. Consignes

Pour ce projet-là, il vous sera demandé de créer un repository avec le nom : ***cc_bot_discord_python***.

N'oubliez pas de push régulièrement !

En cas de question, pensez à demander de l'aide à votre voisin de droite. Puis de gauche. Ou inversement. Puis demandez enfin à un Cobra si vous êtes toujours bloqué(e).

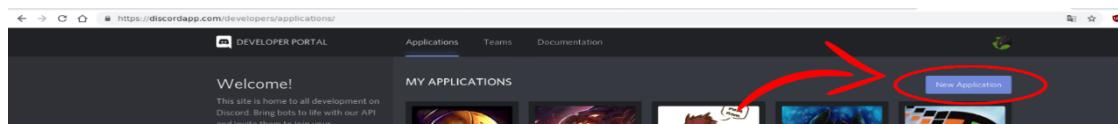
Pensez à faire valider chaque partie que vous réaliserez à un Cobra lorsque vous l'aurez terminée.

N'hésitez pas à faire des bonus et à ajouter des fonctionnalités lorsque votre projet sera terminé et validé.

Vous avez tout à fait le droit d'utiliser internet pour trouver des réponses ou pour vous renseigner.

III. Naissance de votre Poké-bot

Pour créer votre bot, connectez-vous [ici](#) à votre compte Discord ou inscrivez-vous si vous n'en avez pas. Ensuite, rendez-vous sur [le portail des développeurs](#). Cliquez sur « New Application ».



Aperçu du bouton «New Application»

Nommez votre Poké-Bot comme vous le souhaitez et cliquez sur « Create ».

A screenshot of a modal window titled 'CREATE AN APPLICATION'. Inside the window, there is a 'NAME' input field containing the text 'Charamander Bot'. Below the input field, there is a note: 'By creating an API application, you agree to the Discord API Terms of Service'. At the bottom of the window, there are two buttons: 'Cancel' on the left and a blue 'Create' button on the right. The 'Create' button is highlighted with a red circle and a red arrow pointing to it from the left.

Fenêtre pour créer une application

Allez dans l'onglet « Bot » et cliquez sur « Add Bot ».

Bots

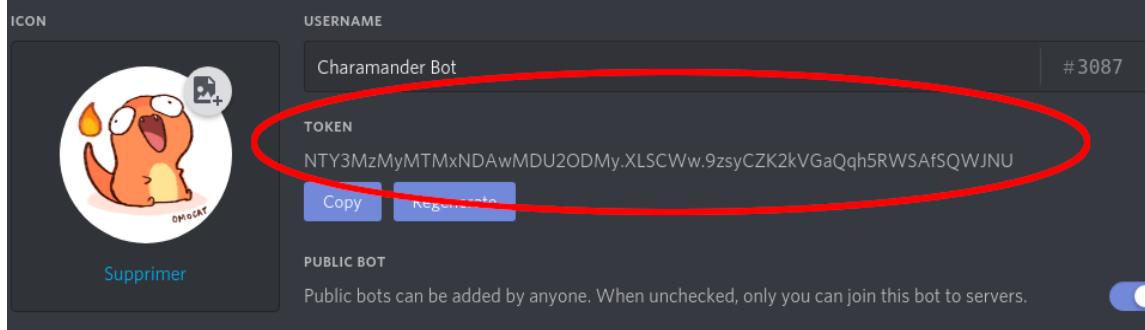
Bring your app to life on Discord with a Bot user. Be a part of chat in your users' servers and interact with them directly.

[Learn more about bot users](#)

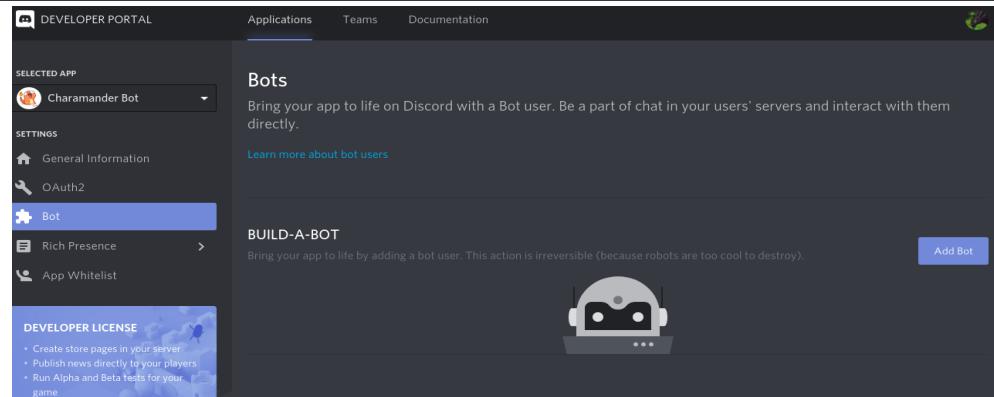
A wild bot has appeared!

BUILD-A-BOT

Bring your app to life by adding a bot user. This action is irreversible (because robots are too cool to destroy).



The screenshot shows the 'BOTS' section of the Discord Developer Portal. On the left, there's a placeholder icon for a bot with the name 'OMOCAT'. To its right, the bot details are shown: Username 'Charamander Bot', ID '#3087', Token 'NTY3MzMyMTMxNDAwMDU2ODMy.XLSCWw.9zsyCZK2kVGaQqh5RWSAfSQWJNU', and two buttons 'Copy' and 'Regenerate'. Below this, under 'PUBLIC BOT', it says 'Public bots can be added by anyone. When unchecked, only you can join this bot to servers.' A toggle switch is shown as checked. A red oval highlights the 'Token' field.



The screenshot shows the 'Bot' settings page. The sidebar on the left has 'Charamander Bot' selected under 'SELECTED APP'. The main area shows the 'BOTS' section with the same information as the previous screenshot: 'Charamander Bot' as the bot name, a placeholder icon, and a public toggle switch. A red oval highlights the 'Token' field in the 'TOKEN' section. At the bottom, there's a large 'Add Bot' button and a small robot icon.

Aperçu de votre Poké-Bot

Mis en valeur en rouge, votre Token ultra secret !

⚠️ Attention : Votre token correspond à votre clé d'identification, c'est-à-dire qu'elle va vous servir à

connecter votre programme au compte bot. Une personne mal intentionnée pourrait faire n'importe quoi

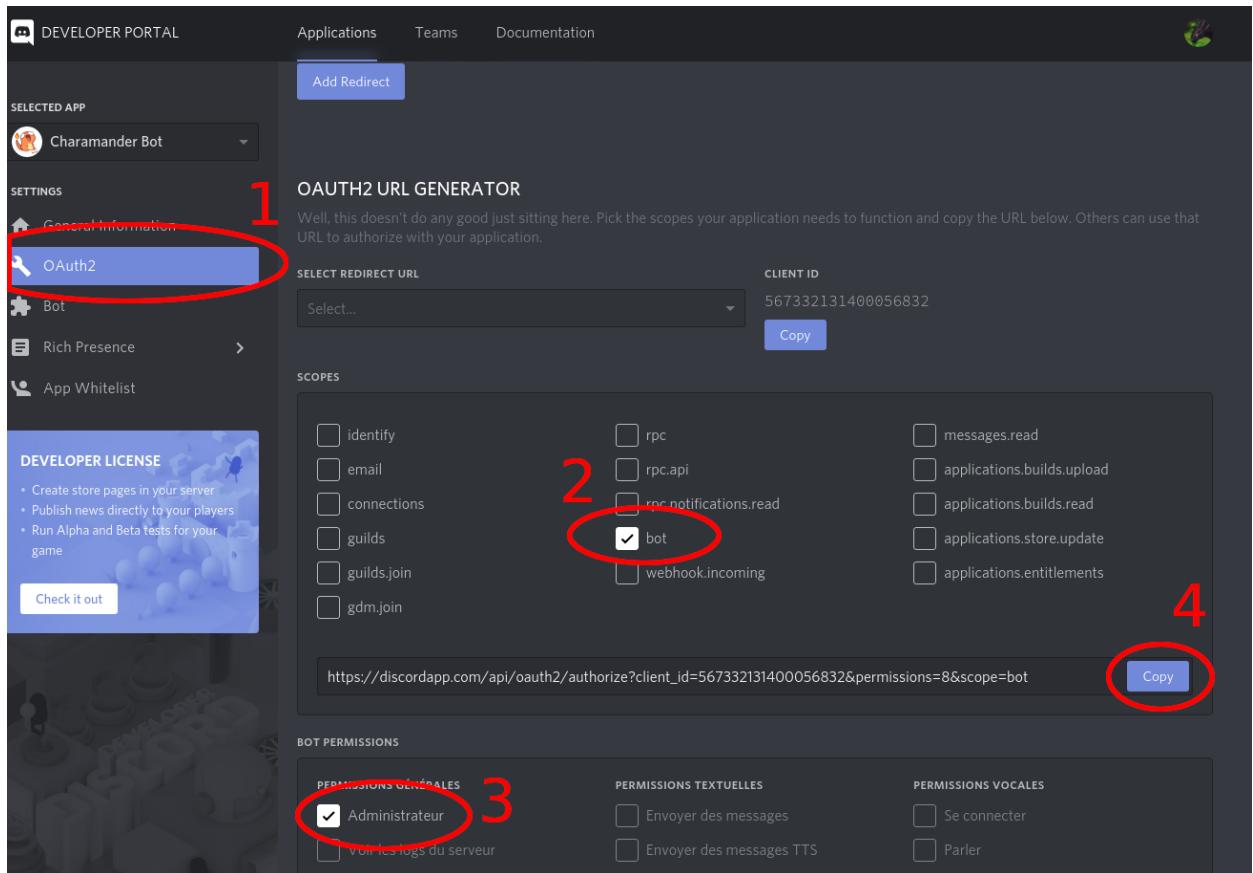
sur les serveurs sur lesquels se trouve votre bot ! Il ne faut donc absolument JAMAIS LE PARTAGER.

Cette chaîne de caractères sera à mettre à la place des endroits indiqués « VOTRE TOKEN » dans le code plus tard. Ensuite, créez un nouveau serveur pour tester le bot ou utilisez celui de votre choix que vous administrez.



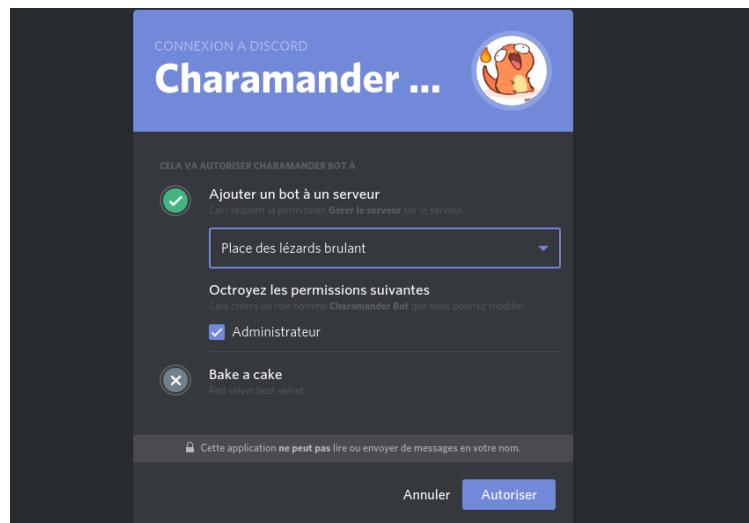
À gauche, votre bouton pour ajouter un nouveau serveur / À droite la fenêtre de création

Retournez sur la page précédente, et créer l'URL Oauth en suivant ces étapes :



Modification de vos périmètres d'action (scopes) et des permissions

Collez l'URL dans votre navigateur. Vous allez arriver sur la page suivante, sélectionnez le serveur que vous venez de créer et cliquez sur « Autoriser ».



Fenêtre de sélection d'ajout du bot à un serveur

À présent, le bot est enfin sur votre serveur, il apparaît déconnecté et c'est normal puisque le programme n'est pas lancé.



Votre Poké-Bot est bel et bien hors ligne

Pour connecter le bot, il faut écrire un programme qui va se connecter avec le token que vous avez généré plus haut. Et c'est ce que vous allez faire !



Félicitations, vous avez atteint le niveau 1 !

You have successfully created a bot and put it on a server.

IV. Hello World!

Avant de coder quoi que ce soit, créez un fichier **.py** dans votre dossier de travail. Ce fichier vous servira justement à pouvoir coder.

a. En avant Salamèche !

Votre première tâche sur ce bot est de réussir à le faire fonctionner un tout petit peu pour voir si vous arriver au moins à récupérer les messages qui passent sur le serveur de Pokéville. Pour cela, vous avez trouvé le petit bout de code suivant qui permettrait d'après le Pokénet de bien débuter pour créer le bot :

```
import discord
import asyncio

client = discord.Client()

@client.event
async def on_ready():
    print("Capitaine Salamèche à l'écoute !")

@client.event
async def on_message(message):
    print(message.content)

client.run("VOTRE_TOKEN")
```

Vous êtes perdu(e) tel un petit lézard dans les bois ? Pas de panique !

La partie la plus importante qu'il faut comprendre, c'est que la fonction `on_message` est exécutée à chaque fois que votre bot voit un message. Dans cet exemple, il se contentera d'afficher le contenu du message reçu dans la console.

b. Les premiers mots de votre Poké-Bot

Si une erreur comme ci-dessous apparaît, il n'y a rien de grave, ne vous inquiétez pas ! C'est probablement que vous n'avez pas indenté votre code. Le Python n'utilisant pas les accolades pour délimiter ses « blocks » mais les indentations (touche de tabulation, n'utilisez pas la barre espace).

```
File "bootstrap.py", line 9
    print(message.content)
          ^
IndentationError: expected an indented block
```

Si tout s'est bien passé, vous devriez voir dans la console les messages que vous envoyez sur votre serveur. C'est un bon début, mais vous allez faire encore mieux !



Vous allez écrire votre première commande !

Vous avez pu voir que vous étiez capable de récupérer le contenu d'un message, mais vous pouvez aussi utiliser cette fonction dans votre programme. Par exemple, vous pouvez comparer si le message correspond à une commande, pour ça vous allez dire au bot : *Si le contenu du message est égal à « !hello » : alors vous envoyez dans le canal de provenance du message « Bonjour humain ! ».*

En termes de code, ça donne ça :

```
@client.event
async def on_message(message):
    if message.content == "!hello":
        await message.channel.send("Bonjour humain !")
```

Le « **await** » permet à votre programme d'attendre une réponse de la commande suivante pour passer à la prochaine étape. En attendant, sachez qu'il faut l'utiliser uniquement quand votre programme envoie des informations sur Discord.

Vous avez maintenant un bon Poké-Bot capable de traiter des messages et de répondre en conséquence. Vous pouvez utiliser plusieurs conditions si vous voulez plusieurs comportements, par exemple :

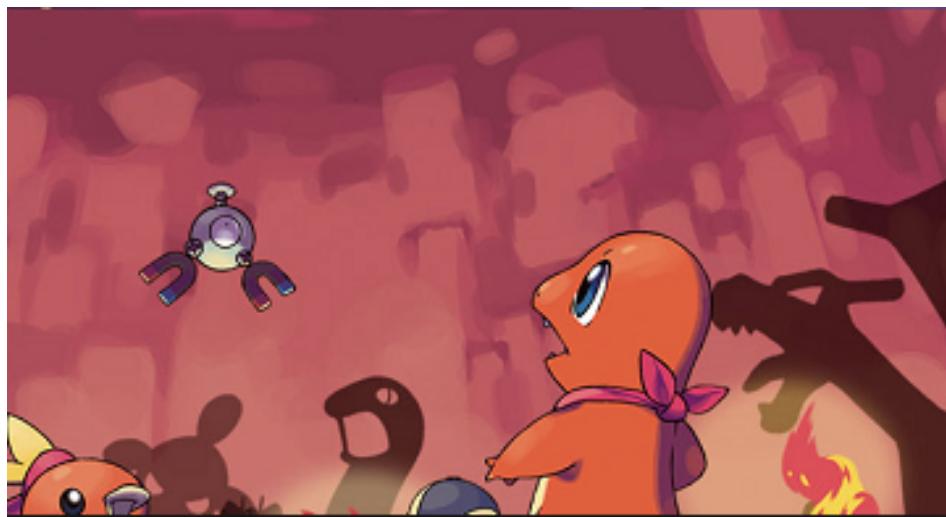
```
@client.event
async def on_message(message):
    if message.content == "!hello":
        await message.channel.send("Bonjour humain !")
    if message.content == "!carapuce":
        await message.channel.send("Beurk")
```



Félicitations, vous avez atteint le niveau 2 !

Vous avez fait toutes les installations pour le bot.

V. Un objet bien mystérieux ...



Stupéfaction chez les habitants de Pokéville !

Durant vos recherches sur les différentes fonctionnalités et comment les utiliser, vous avez beaucoup entendu parler (ou plutôt lu) à propos des objets.

- a. Un objet ? Mais qu'est-ce que c'est que cette bête-là ?

Un objet, pour résumer, c'est quelque chose qui contient des valeurs : ça peut être une chaîne de caractères, aussi appelé « *string* » ou « *str* », comme votre `message.content`, une structure plus complexe comme une liste ou un dictionnaire ou bien un autre objet.

Par exemple, votre objet `message`, qui vous a été donné par la fonction `on_message`, contient plusieurs attributs intéressants :

`Content` : Donne le contenu du message.

`Channel` : Donne les informations du salon souhaité.

`Guild` : Donne les informations sur le serveur discord.

`Author` : Donne les informations de la personne ayant écrit le message.

Si vous souhaitez connaître l'objet `message` sur le bout des griffes, vous pouvez vous renseigner sur cette partie de la [documentation officielle](#).

b. Exploitons toute la puissance de l'objet

Vous avez tout ce qu'il vous faut dans votre boîte à outils à présent, donc vous pouvez ajouter quelques fonctionnalités au bot ! Le Maire de Pokéville aimerait que ses concitoyens puissent retrouver facilement les informations disponibles, dans la même idée qu'ils pourraient trouver des affiches dans la rue.

Pour vous aiguiller sur l'utilisation générale des objets voici quelques points importants :

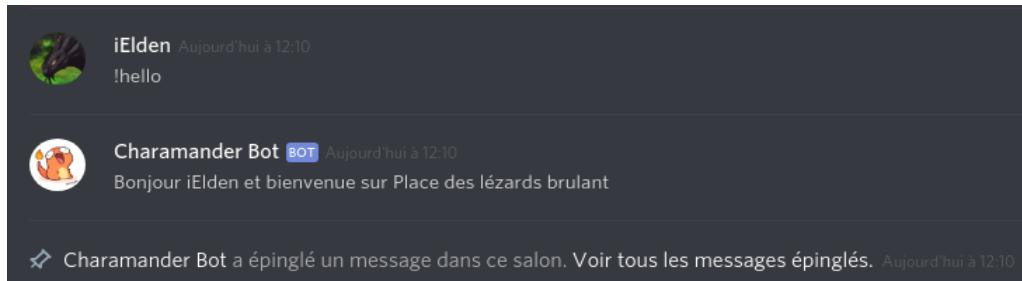
Les objets ont des attributs (comme `.channel`, `.guild` ou `.author`), ce qui vous permet d'avoir d'autres objets, généralement pour récupérer des informations.

Les objets ont des fonctions (comme `.send()` ou `.pin()`) ce qui vous permet d'effectuer des actions avec un bot comme envoyer un message, le supprimer, gérer un membre, un rôle etc....

Vous pouvez, par ailleurs, voir que `.send()` renvoie un nouvel objet `Message`, qui est ensuite utilisable comme celui que l'on reçoit au début.

Vous allez pouvoir, par exemple, améliorer votre `!hello` pour qu'il salue la personne en disant son nom. Pour faire cela, vous allez utiliser l'attribut `name` de l'objet `message.author`.

L'objet `message` est très complet, vous pouvez vraiment faire énormément de choses avec. Par exemple, rien ne vous empêche de saluer votre interlocuteur en lui souhaitant la bienvenue sur votre serveur, puis ensuite d'épingler le message.



Votre Poké-Bot vous appelle enfin par votre pseudo !



Félicitations, vous avez atteint le niveau 3 !

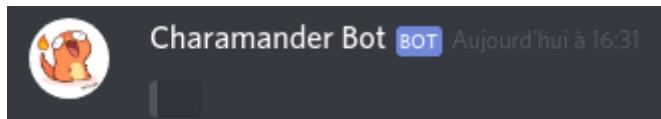
Vous savez maintenant utiliser et interpréter les messages avec votre Poké-Bot.

VI. Mettez un peu de décos dans votre vie

Vous souvenez-vous de la dernière requête du maire ? Il parlait d'affiches. Quoi de mieux que d'utiliser un format « **embed** » pour que les informations transmises soient plus jolies que dans un simple message ? Ce format permet de poster des messages possédant un titre, un corps et plus en encore ! Pour faire un embed minimaliste, suivez ces lignes de codes :

```
em = discord.Embed()  
await message.channel.send(embed=em)
```

Ce qui donnera :



Début de création prometteur de votre embed !

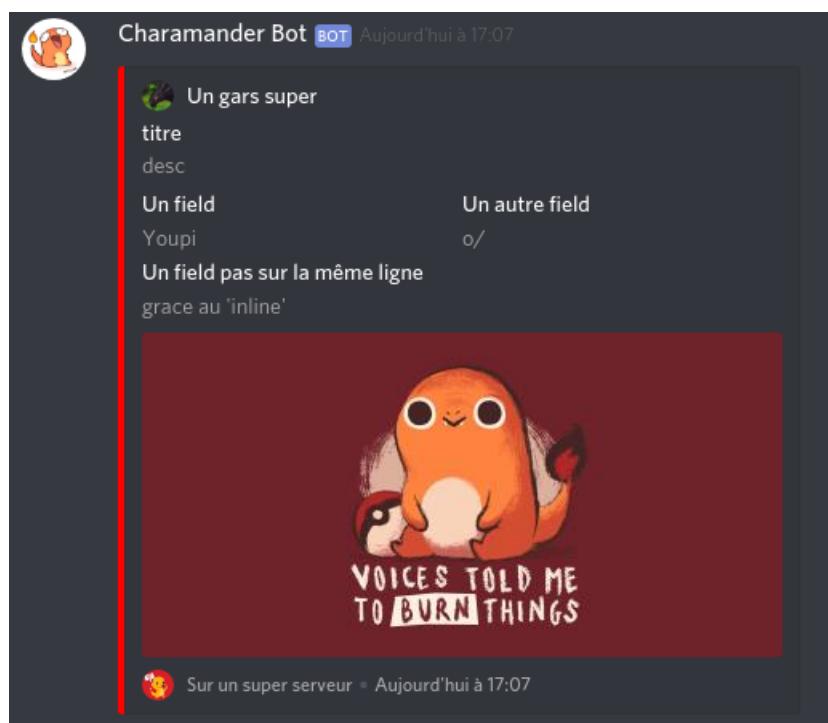
Bravo, c'est un bon début !

En Python, les fonctions peuvent prendre des paramètres optionnels, ce sont des **Keyword Arguments** (abrégés en **kwargs**). Par exemple, la fonction `.send()` que vous pouvez voir ci-dessus prend un paramètre optionnel appelé `embed` qui vous permet de rajouter une mise en page en plus du message. Ce qu'il y a de bien avec ce format, c'est qu'ils prennent PLEIN de kwargs.

Essayez par exemple avec ce code :

```
em = discord.Embed(title="titre", description="desc",
colour=0xFF0000,
                           timestamp=message.created_at)
em.add_field(name="Un field", value="Youpi", inline=True)
em.add_field(name="Un autre field", value="o/", inline=True)
em.add_field(name="Un field pas sur la même ligne", value="grâce
au 'inline'", 
                           inline=False)
em.set_author(name="Un gars super",
icon_url=message.author.avatar_url)
em.set_footer(text="Sur un super serveur",
icon_url=message.guild.icon_url)
em.set_image(url="https://cdn.discordapp.com/attachments/"
+
"567630033246617621/581496185424969749/1733852_0.jpg")
await message.channel.send(embed=em)
```

Ce qui donnera :



En voilà une belle mise en page !

Ça ressemble quand même plus à quelque chose comme ça, non ? On ne va pas trop s'attarder sur les embeds, mais si vous voulez plus d'information dessus et connaître toutes leurs possibilités, renseignez-vous sur internet.



Félicitations, vous avez atteint le niveau 4 !

Vous savez maintenant manier la mise en page comme un chef.

VII. Poké-Bot the Robocop

A la mairie de Pokéville, une nouvelle loi a été votée interdisant à la population de prononcer certains mots jugés contrariants. Le service courrier Bekipan a distribué la liste à tous les habitants de Pokéville mais la mairie n'a pas assez de poké\$ pour engager assez d'administrateurs sur son serveur Discord pour faire respecter la nouvelle loi. Il a besoin d'une solution, c'est pour ça que l'on fait appel à vous !



Découverte de la liste des mots interdits par les habitants de Pokéville

- Les listes de mots à bannir à tout jamais de votre Pokébouche

Une liste en Python est une structure qui permet de contenir plusieurs variables. Ici, vous allez vous en servir pour contenir les mots interdits par la nouvelle loi.

Déclarez, en haut de votre code, une liste de mots interdits comme par exemple :

```
BAN_WORDS = ["pokeball", "carapuce", "dresseurs", "javascript", "giratina"]
```

En Python, les variables sont écrites en majuscules pour informer les autres programmeurs que ce sont des constantes. C'est-à-dire que le programme doit consulter les variables mais pas les modifier.

b. Bannez moi ça les modos !

Une boucle for en Python, va exécuter le code compris dans le bloc autant de fois qu'il y a d'éléments dans la liste. A chaque tour de boucle, la variable va prendre la valeur d'un élément de la liste.

Par exemple, le code ci-dessous :

```
for word in BAN_WORDS:  
    print("La variable word vaut : " + word)
```

Va produire ce résultat :

```
La variable word vaut : pokeball  
La variable word vaut : carapuce  
La variable word vaut : dresseurs  
La variable word vaut : javascript  
La variable word vaut : giratina
```

On va donc vérifier un par un si chaque mot est présent dans le contenu du message. Et pour ça, il vous faut utiliser le mot-clé `in`.

```
if word in message.content:  
    print("ALERTE : mot interdit utilisé")
```

En combinant cela et ce que vous avez vu dans le [Chapitre III](#), vous pouvez donc effectuer la procédure suivante :

Supprimer le message de la personne incriminée.

Envoyer un avertissement contenant le mot interdit utilisé et le message d'origine.

On souhaite ici que le résultat soit équivalent au suivant :



Le robot Salameche fait régner l'ordre, tout en bienveillance !

c. Triomphe de l'ordre et de la paix

Avez-vous regardé dans la documentation ce que faisaient les fonctions suivantes ? Il y a fort à parier que cela vous servirait !

`delete()`

`send("Quelque chose")`

`lower()`

Le saviez-vous ? `break` est un mot-clé qui « casse » la boucle lorsque vous l'utilisez. Ce qui peut vous éviter que le bot veuille supprimer deux fois le message si plusieurs mots bannis sont utilisés.

Dans votre algorithme, avez-vous pensé à regarder qui était l'auteur du message ? Ce serait dommage que *Salamèche* se retrouve à bannir ses verbalisations...

Par ailleurs, voici le message du maire pour *Salamèche* :

Toutes mes félicitations agent Salamèche ! Grâce à vous et à votre maîtrise sans égal des listes, la paix et l'ordre ont été rétablis sur le serveur de Pokéville, loin de tous les mots barbares que l'on peut rencontrer.

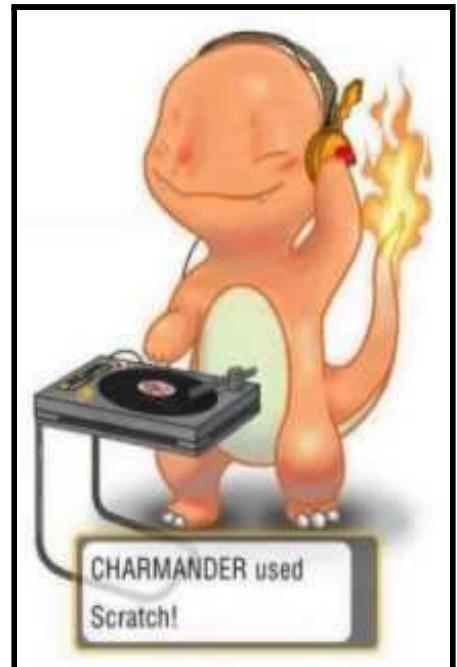


Félicitations, vous avez atteint le niveau 5 !

Vous avez réussi à sécuriser le serveur de Pokéville et à protéger les habitants de ces horribles mots !

VIII. DJ Salamèche est aux platines

Queulorior est venu vous voir, en panique, pour vous parler de son problème. En effet, *Rondoudou* ne veut plus animer la soirée discothèque à la maison car il est agacé que les gens s'endorment pendant ses prestations. Il compte sur vous pour animer la soirée d'aujourd'hui !



Salameche est prêt!

a. Mise en place du matériel pour mixer

Queulorior ayant déjà commencé à préparer la soirée a trouvé quelque chose qui pourrait vous servir. Allez savoir par quel hasard ça porte de nom de *SalamecheMusic.py*... Cependant il semblerait que ce soit un module dont vous pourriez avoir besoin. Donc dans un premier temps, déplacez le module nommé *SalamecheMusic.py* dans votre dossier de travail. Puis, dans votre fichier .py, importez le module en ajoutant cette ligne en haut du fichier :

```
import SalamecheMusic
```

⚠️ Attention : il faut marquer import *SalamecheMusic* et non *SalamecheMusic.py* car Python ajoute tout seul le .py

- b. Couper et recouper pour arriver à ses fins

Une dernière chose avant de terminer, la méthode *split*. Elle va vraiment vous être utile. Elle permet de séparer une chaîne de caractère en une liste de chaîne de caractères, en d'autres mots : changer une phrase en liste de mot.

Voici quelques exemples :

```
>>> message.content
!play http://youtube.com/randomvideo
>>> message.content.split(' ')
['!play', 'http://youtube.com/randomvideo']
>>> message.content.split(' ')[0]
!play
>>> message.content.split(' ')[1]
http://youtube.com/randomvideo
```

Maintenant, vous avez tous les outils en mains pour pouvoir assurer la soirée ! Essayez donc de faire une commande « `!play {lien youtube}` » pour que le bot joue la musique envoyée dans le canal vocal.

Regardez les autres commandes auxquelles vous avez accès dans le module. Elles pourraient vous être utiles. 😊



Félicitations, vous avez atteint le niveau 6 !

Vous avez sauvé de nombreuses soirées endiablées, vous êtes vraiment le roi de la fête !

IX. Toujours plus loin, toujours plus haut, toujours plus fort !

Le maire de Pokéville est très heureux du travail que vous avez fourni ! Le serveur devrait donc pouvoir tourner avec de bonnes fonctionnalités maintenant ! Bien que ce soit déjà la fin de cette histoire, ce n'est que le début de votre épopee !

Les possibilités de création sont infinies, voici quelques petits défis à réaliser :

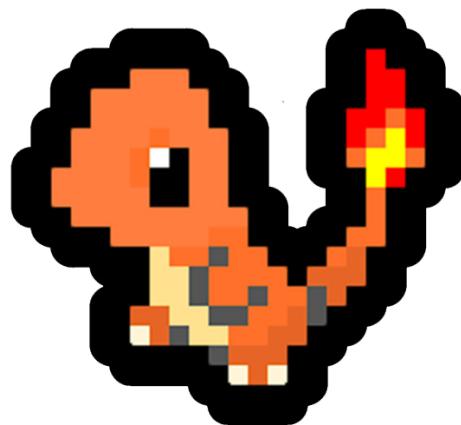
Reprendre l'exercice précédent et rajoutez des fonctionnalités, comme une *queue* pour les musiques par exemple.

Refaire l'exercice précédent sans utiliser le module *SalamecheMusic*.

Créer une nouvelle fonctionnalité comme un questionnaire interactif.

Faire une image personnalisée avec la photo de profil de chaque nouvel arrivant sur un serveur (manipulation d'images).

Un des grands avantages de Python, c'est qu'il existe des tonnes de modules pour pouvoir faire facilement tout ce dont vous avez envie, vous avez vu *youtube-dl* pour *Youtube*, mais il en existe plein d'autre, comme *PIL* pour la manipulation d'image, *datetime* pour encore plus gérer les heures et dates ou bien l'*API* de votre jeu vidéo favori pour récupérer des informations ! En espérant vous avoir donner envie de développer encore plus !



Salameche vous remercie !