

[Python 102: Discord Bot for Server Functions]

Tuesday, February 11th | 12:30 PM - 1:30 PM | HUB Games Arena

Agenda: 12:30-12:50 PM (lesson), 12:50 PM - 1:20 PM (hacking in groups)

Details:

1) Review of what we have done

We have covered downloading:

- PyCharm, Python, Discord.py, setting up server, and everything is in the prior lesson
- If you need any reference, the documentation is in the server where we can continue talking about developing a discord bot together there
- This is what we previously coded up - the token is dependent on your bot

```
import discord
from discord.ext import commands

bot_channel = 66418613223322517
client = commands.Bot(command_prefix='&')
TOKEN = TOKEN

@client.event
async def on_ready():
    post_time.start()
    print('Bot is ready.')

@client.event
async def on_member_join(member):
    print(f'{member} has joined the server.')

@client.command()
async def ping(ctx):
    await ctx.send('Pong!')

client.run(TOKEN)
```

Tasks & Loops, Errors & Administrative server functions

Tasks & loops:

Overview: Tasks are able to have loops that run scheduled based off of what time increment you set within it. You will be able to start and stop the command through different functions. For a `post_time` method, you these are the primary commands for controlling it: `post_time.stop()`, `post_time.start()`.

Importing: You will need to import tasks from `discord.ext`

- Depending on what you want to send, you may want to consider import `random` for a random generator

Example: Reference the `post_time` method and the header.

Errors:

Overview: User errors such as inputting the wrong number of inputs, having the wrong input type, or any other errors can be used using `method_name.error`. This function will run only when an error has been given where you can then use this to prompt the user.

Example: Reference the `load_error` method below

Administrative Server Functions:

Overview: Administrative server functions focuses around having permissions. Every user has a different permission within discord and checking these will section off which functions are applicable to which user.

Example: Reference the `post_time_status` method header

```
@client.command()
@commands.has_permissions(manage_messages=True)
async def post_time_status(ctx, command: int):
    if command == 1:
        post_time.stop()
        await ctx.send('You have stopped the automatic loop.')
    elif command == 2:
        post_time.start()
        await ctx.send('You have started the automatic loop')
    else:
        await ctx.send('You must enter in \'%post_time_status <1/2>\'.\n1 is to stop the loop.\n2 is to start the loop.')
```

```
@post_time_status.error
async def load_error(ctx, error):
    if isinstance(error, commands.MissingRequiredArgument):
        await ctx.send(f'**Error:** You did not enter a number following \'&post_time_status\'.')
```

```
@tasks.loop(seconds=10)
async def post_time():
    channel = client.get_channel(int(bot_channel))
    await channel.send(random.randint(1, 10))
    print('Ran through loop')
```