

# Discord Bot

## Del 2: Reactions

Sidste gang lavede vi en Discord-bot der kan læse beskeder og svare på dem på forskellige måder. I denne opgave skal du lave en bot der bruger discord's reaction-system.

HUSK AT GEM din kode, så du kan kigge på den senere, når vi skal lave det endelige discord-bot-projekt.

## Reactions

Kig igen på `on_message`-funktionen fra `echobot.py`:

```
@client.event
async def on_message(message):
    contents = message.content
    if contents.startswith("!echo "):
        rem = contents[5:]
        reply = "Du sendte: " + rem
        await message.channel.send(reply)
```

Input-parameteret `message` er et objekt af typen `Message`. På `Message`-objekter findes en funktion der hedder `add_reaction`. Ligesom `send`, så er `add_reaction` en `async`-funktion, og man skal derfor huske at bruge `await` når man kalder den. Altså:

```
await message.add_reaction("\N{EYES}")
```

Her er "`\N{EYES}`" kode for eyes-emojien (se <https://www.fileformat.info/info/unicode/char/1f440/index.htm>).

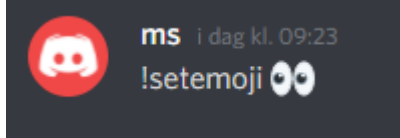
Der er flere måder at repræsentere en emoji på. Der er en fin beskrivelse på <https://gist.github.com/scragly/b8d20aece2d058c8c601b44a689a47a0>.

## Opgave: En virkelig irriterende bot

Dagens første opgave er at lave en Discord-bot der sætter reactions på *alle* beskeder. For at gøre det lidt nemmere at udholde denne irriterende bot, skal det være muligt for hver bruger at vælge en yndlings-emoji, som botten skal bruge når den sætter reactions på netop den brugers beskeder.

1. Start med at lave botten så den sætter den samme reaction på alle beskeder. Kør din kode og tjek at det virker.
2. Tilføj en kommando der tillader en bruger at vælge sin yndlingsemoji. Du kan gemme brugernes valgte emojis i en dictionary, som har bruger-IDer som nøgler (`message.author.id`).

Hvis kommandoen er `!setemoji`, kunne en besked f.eks. se sådan her ud:



Kør din kode, og tjek at der ikke er nogle fejlbeskeder når man forsøger at sætte sin yndlings-emoji.

3. Lav om i din `on_message`-funktion, så din bot bruger den emoji brugeren har valgt. Hvis brugeren ikke har valgt en emoji, skal botten bruge den emoji du brugte i trin 1.

Kør din kode, og tjek at det virker.

## Ny funktion: `on_reaction`

En bot kan også opdage når andre laver reactions på beskeder. Det gør den ved hjælp af et event der hedder `on_reaction`.

Definer funktionen `on_reaction` i din kode. Ligesom `on_message` skal det være en async-funktion. `on_reaction` skal have to input-parametre – det første er den reaction der er blevet sat, og det andet er brugeren der har sat den. Altså:

```
@client.event
async def on_reaction(reaction, user):
    pass
```

## Opgave: Emoji-tæller

Dagens anden opgave er at lave en bot der tæller hvor mange gange forskellige emojis bliver brugt som reaction.

- I `on_reaction`-funktionen kan du bruge `reaction.emoji` til at få fat på den emoji der er blevet sendt.

- Lav en kommando `!reacts` der sender en liste over emojis og hvor mange gange de er blevet brugt som reaction. Det kan f.eks. se sådan her ud:



### Bonus-opgave: Dynamisk yndlings-emoji

Kombiner din virkelig irriterende bot fra den første opgave med din emoji-tæller bot: I stedet for at brugeren selv bestemmer hvad deres yndlings-emoji er, skal botten nu gætte det ud fra hvilken emoji brugeren oftest reacter med.

Det vil altså sige at emoji-tælleren nu skal holde styr på hver bruger, og hvor mange gange de har reactet med hvilke emojis.

Hint: Man kan godt have dictionaries inde i dictionaries.