This challenge focuses on python syntax and classes instances.

First of all, we can look at the print statement:

print(f"{{message}} {final_pattern}".format(message=random.choice(MESSAGES)))

What is easily visible is that final_pattern is never filtered or syntax verified, so if **final_pattern={message}**, the print command effectively becomes

print(f"{{message}} {{message}}".format(message=random.choice(MESSAGES)))

This way, we have access to "message". But what is message exactly? It's a class instance, which shares all of the functions the class has. Meaning, we can access **message.__str__** and any other methods, like init and repr.

Where am I going with this? Every python function has a **__globals__** attribute. This attribute contains the entire environment of the function, including stuff like environment variables. I hope it's clear what I mean by this.

If we consider **final_pattern={message.__str__.__globals__}**, we effectively print **the contents of the environment** of **a function** of **an instance** created in the program we are executing, meaning we print all the environment variables, among other things. This way we exploit the python code to get the flag.

Made with love by: AndreiCat