

Creating your first agent

Introduction

Once you've [installed](#) the uAgents library, it's quite simple to get a minimal use case running. Let our first agent be a simple initialisation and printing out the agent's name.

The agent

1. Let's create a Python script for this task, and name it by running: touch agent.py
2. We then need to import the Agent
3. and Context
4. classes from the uagents
5. library, and then create an agent using the class Agent
6. :
7. from
8. uagents
9. import
10. Agent
11. ,
12. Context
13. agent
14. =
15. Agent
16. (name
17. =
18. "alice"
19. , seed
20. =
21. "secret_seed_phrase"
22.)
23. It is optional but useful to include a seed
24. parameter when creating an agent to set fixed [addresses](#)
25. . Otherwise, random addresses will be generated every time you run the agent. Your address is kind of important, as this is how other agents will identify you.
26. Let's define a say_hello()
27. function for our agent to print a message periodically saying hello, my name is ...
28. :
29. @agent
30. .
31. on_event
32. (
33. "startup"
34.)
35. async
36. def
37. introduce_agent
38. (
39. ctx
40. :
41. Context):
42. ctx
43. .
44. logger
45. .
46. info
47. (
48. f
49. "Hello, I'm agent
50. {
51. agent.name
52. }
53. and my address is
54. {
55. agent.address
56. }
57. ."

```

58. )
59. if
60. name
61. ==
62. "main"
63. :
64. agent
65. .
66. run
67. ()
68. The.on_event("startup")
69. decorator defines a behavior for this agent when it is run. In this case, the agent will execute thesay_hello()
70. function when the agent starts. TheContext
71. object is a collection of data and functions related to the agent. In this case, we just use the agent'sname
72. ,alice
73. . The agent executes the function and uses thectx.logger.info()
74. method to print the message.
75. Save the script.

```

The overall script should look as follows:

```
agent.py from uagents import Agent , Context
```

agent

```
Agent (name = "alice" , seed = "secret_seed_phrase" )
```

```
@agent . on_event ( "startup" ) async
```

```
def
```

```
introduce_agent ( ctx : Context): ctx . logger . info ( f "Hello, I'm agent { agent.name } and my address is { agent.address } ."
)
```

```
if
```

```
name
```

```
==
```

```
"main" : agent . run ()
```

Run your agent

Make sure to have activated your virtual environment correctly.

Run the script:python agent.py

The output would be:

```
[alice]: Hello, I'm agent alice and my address is Congratulations, you have just created your first Agent!
```

Was this page helpful?

[uAgents Framework installation](#) [Agents address](#)