API and Testing Reference

Environment variables

Expression	Purpose
`os.environ['NAME'] = 'value'`	Set a variable for the running program
`os.environ.get('NAME')`	Read its string value, or `None` if absent
`os.getenv('NAME')`	Another spelling of the same lookup
`load_dotenv()`	Load settings from `.env` into the environment

Message roles

Role	Purpose
`system`	Instructions for the model, such as how to respond
`user`	The person's input or text to process
`assistant`	A reply generated by the model

Offline tests

Test tool	Purpose
`MagicMock(return_value=value)`	Return the same value on each call
`MagicMock(side_effect=[a, b])`	Return successive values on successive calls
An exception in `side_effect`	Raise that exception on the corresponding call
`mock.call_count`	Number of times the mock was called
`patch('module.name', mock)`	Replace the named object inside the `with` block
`patch.object(object, 'name', ...)`	Replace that object's named attribute inside the block