Context Managers

Advanced Python Programming

—

Valerio Velardo - The Sound of AI

What's a context manager?

*Object that defines a runtime context executing within the **with** statement*

Context manager syntax

```
with context as cont:
    ...
```

Context manager syntax

- *with* statement creates new context manager

```
with context as cont:
    ...
```

Context manager syntax

- *with* statement creates new context manager
- An object may be returned

```
with context as cont:
    ...
```

Context manager syntax

- *with* statement creates new context manager
- An object may be returned
- After the *with* block, context is teared down

```
with context as cont:
    ...
```

Context manager protocol

- To create a custom class-based context manager, implement 2 magic methods: __enter__(), __exit__()

Context manager protocol

- To create a custom class-based context manager, implement 2 magic methods: __enter__(), __exit__()
- __enter__() sets up the context and can return a value optionally (not the context object!)

Context manager protocol

- To create a custom class-based context manager, implement 2 magic methods: __enter__(), __exit__()
- __enter__() sets up the context and can return a value optionally (not the context object!)
- __exit__() tears down the context

Context manager flow

Context manager flow

1. with -> instantiate context manager object

Context manager flow

1. with -> instantiate context manager object
2. __enter__() -> set up context / return object

Context manager flow

1. with -> instantiate context manager object
2. __enter__() -> set up context / return object
3. # do stuff

Context manager flow

1. with -> instantiate context manager object
2. __enter__() -> set up context / return object
3. # do stuff
4. __exit__() -> clean up context / called if exceptions occurs

What are context managers good for?

- Manipulate files (open / close)
- Access databases (connect / disconnect)
- Manage threads (start / stop)
- Create / delete resources
- …