

#07 - PyLM Meetup at Técnico 🎓🐍

Apr 2 2026

PYTHON LISBON MEETUP



Announcements

- We have a [Code of Conduct](#)
- We are now sponsored by the PSF ❤️
- PyCon PT CFP is open until midnight on April 30th

<https://python-lisbon-meetup.github.io/>

PYTHON LISBON MEETUP



Lazy imports

Python Lisbon Meetup

Anže, 2 Apr 2026

imports

The problem

my_module.py

```
import time
```

```
time.sleep(3)
```

```
def my_func():  
    pass
```

```
from my_module import my_func
```

app on lazy [?!] via desktop-linux is v1.0.0 via v24.13.0 via v3.13.11 took 3s
) time ./manage.py check

app2 on main [?!] via desktop-linux is v1.0.0 via v24.13.0 via v3.13.11 took 12s
) time ./manage.py check

I

python -X importtime

Inline imports

```
from pathlib import Path
```

```
def get_path():
```

```
    return Path(".")
```

```
def get_path():  
    from pathlib import Path  
    return Path(".")
```

Type annotations

```
from pathlib import Path
```

```
def get_path() -> Path:
```

```
    return Path(".")
```

```
from typing import TYPE_CHECKING
```

```
if TYPE_CHECKING:
```

```
    from pathlib import Path
```

```
def get_path() -> Path:
```

```
    from pathlib import Path
```

```
    return Path(".")
```

Circular import errors

Lazy imports

PEP 690 – Lazy Imports

PEP 690 - Lazy Imports

[REJECTED]

PEP 810 – Explicit lazy imports

PEP 810 – Explicit lazy imports

ACCEPTED

lazy import json

lazy from json import dumps

```
__lazy_modules__ = ["json"]
```

```
import json  
from json import dumps
```

```
-X lazy_imports=<mode>
```

- "normal" (or unset): Only explicitly marked lazy imports are lazy
- "all": All module-level imports (except in try blocks and import *) become potentially lazy
- "none": No imports are lazy, even those explicitly marked with lazy keyword

```
sys.set_lazy_imports_filter(func)
```

```
import sys

def exclude_side_effect_modules(importer, name, fromlist):
    """
    Filter function to exclude modules with import-time side effects.

    Args:
        importer: Name of the module doing the import
        name: Name of the module being imported
        fromlist: Tuple of names being imported (for 'from' imports), or None

    Returns:
        True to allow lazy import, False to force eager import
    """
    # Modules known to have important import-time side effects
    side_effect_modules = {'legacy_plugin_system', 'metrics_collector'}

    if name in side_effect_modules:
        return False # Force eager import

    return True # Allow lazy import

# Install the filter
sys.set_lazy_imports_filter(exclude_side_effect_modules)

# These imports are checked by the filter
lazy import data_processor # Filter returns True -> stays lazy
lazy import legacy_plugin_system # Filter returns False -> imported eagerly
```

Playground!