

Squaring the Circle

(How) can a single language like **Python** satisfy both simple scripting and serious production use cases?

Python: Obstacles and Opportunities

- The software engineering community is split whether Python is a language suitable for production use cases!
- “I’ve learned a painful lesson that for small programs dynamic typing is great. **For large programs you have to have a more disciplined approach and it helps if the language actually gives you that discipline**, rather than telling you “Well, you can do whatever you want.”

[Guido van Rossum in 2019](#) (emphasis added)

- **You can write production-grade software in Python - but only if you *add additional optional safeguards*.**

Production-grade software in Python

- What makes Python easy to get started makes it hard to scale!
 - ...but rather than having two *separate programming languages* for interactive data science and production software, we can simply have *two different sets of quality standards/tools*
 - By *gradually adding additional safeguards*, we avoid the cost of having to re-write the whole code!
 - E.g., gradual typing (types can be enforced by running static analysis tool, but are ignored at runtime)
- > **Agile**
- > **...but this requires solving the *organizational/management problem* of making sure these additional safeguards are actually added later - which is not an easy problem!**

Production-grade software in Python

- How to avoid the “Billion dollar mistake”
 - The equivalent of nullpointer errors is easily avoided in Python by type hints + static analysis
 - We should never see this: `AttributeError: 'NoneType' object has no attribute ...`

```
def make_api_request(endpoint: str) -> Response | None:
    try:
        response: Response = requests.get(endpoint, timeout=5)
        return response
    except Exception as e:
        return None

response_1: Response | None = make_api_request(endpoint='https://httpbin.org/get')
response_2: Response | None = make_api_request(endpoint='https://httpbin.orgg/get')

response_1.json()
response_2.json()

"json" is not a known member of
"None" PyLance(reportOptionalMemberAccess)
```

```
engineering-standards/best-practices/python/typing/basics/nullpointer_
error.py
Traceback (most recent call last):
  File "/home/thomas-22/repos/engineering-standards/best-practic
es/Python/typing/basics/nullpointer_error.py", line 14, in <modu
le>
    response_2.json()
AttributeError: 'NoneType' object has no attribute 'json'
(Python) thomas-22@MSI:~/repos$
```