TO IMPROVE YOUR PYTHON, LEARN RUST

a very promising partnership

... RUST?

YOU MIGHT BE ALREADY USING IT!

ruff: *black* but faster

Pydantic 2.0: *Pydantic* but faster

polars: *Pandas* but faster

tiktoken: tokenizer for OpenAI models

…

COMPLEMENTARY SKILLSET

two languages that complement each other

Python		Rust
Fast to develop	<->	Fast to run
Flexible	<->	Robust
Large ecosystem	<->	Small artefacts

GETTING STARTED

generate a project skeleton with `maturin`

```
$ maturin new --mixed --bindings pyo3 something
```

↓

```
$ tree something

something
├── Cargo.toml
├── pyproject.toml
├── python
│   └── something
│       └── __init__.py
└── src
    └── lib.rs
```

RUST VIEW

```rust
use pyo3::prelude::*;

#[pyfunction]
fn sum_as_string(a: usize, b: usize) -> PyResult<String> {
    Ok((a + b).to_string())
}

#[pymodule]
fn something(_py: Python, m: &PyModule) -> PyResult<()> {
  m.add_function(wrap_pyfunction!(sum_as_string, m)?)?;
  Ok(())
}
```

PYTHON VIEW

```python
from something import sum_as_string

print(sum_as_string(2, 2))
# 4
```

NOW DO SOMETHING MORE COMPLEX 🚀

Performance-sensitive task...

Concurrent tasks...

Memory pressured tasks...

Systems programming...

Image processing...

💭

LEARN GOOD PRACTICES FROM RUST

Make documentation a first class feature

Put more thought into error handling

User iterators & generators; pattern matching

Prefer immutable variables

... *[your insights here]* ...

Louie, I think this is the beginning of
a beautiful friendship

maturin.rs