Python and Rust, a perfect pairing

Miikka Koskinen

https://miikka.me/talks/python-and-rust/

```c
static PyObject* my_sum_sum(PyObject* self, PyObject* args) {
    PyObject* input_list;
    if (!PyArg_ParseTuple(args, "O!", &PyList_Type, &input_list)) {
        return NULL;
    }

    Py_ssize_t len = PyList_Size(input_list);
    long long total = 0;

    for (Py_ssize_t i = 0; i < len; i++) {
        PyObject* item = PyList_GetItem(input_list, i);
        if (!PyLong_Check(item)) {
            PyErr_SetString(PyExc_TypeError, "List elements must be integers");
            return NULL;
        }
        long long value = PyLong_AsLongLong(item);
        if (PyErr_Occurred()) {
            return NULL;
        }
        total += value;
    }

    return PyLong_FromLongLong(total);
}
```

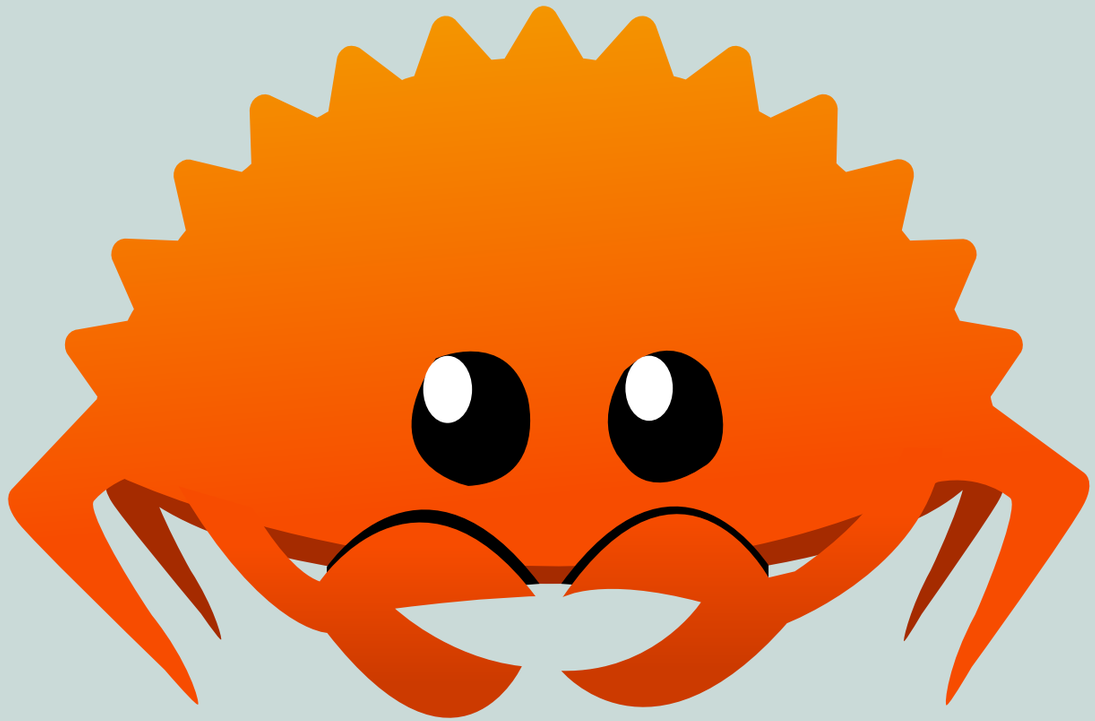
```rust
#[pyfunction]
fn my_sum(numbers: Vec<i64>) -> i64 {
    numbers.iter().sum()
}
```

Agenda

1. Why Rust?
2. How to Rust?

https://miikka.me/talks/python-and-rust/

Part 1: Why Rust?



Unlike Python, Rust is fast

https://miikka.me/talks/python-and-rust/

Like Python, Rust is nice to use

- Expressive: structs, enums, pattern matching, traits, powerful type system
- Enforces memory safety at compile time
- Helpful compiler and ergonomic tools
- Active community

Examples of Python libraries with Rust inside them:

- cryptography
- Polars
- Pydantic
- orjson

A pattern is emerging:

Python library with its core written in Rust

Part 2: How to Rust?

https://miikka.me/talks/python-and-rust/

https://www.rust-lang.org/learn

Learn Rust

Get started with Rust

Affectionately nicknamed “the book,” *The Rust Programming Language* will give you an overview of the language from first principles. You’ll build a few projects along the way, and by the end, you’ll have a solid grasp of the language.

READ THE BOOK!

Alternatively, Rustlings guides you through downloading and setting up the Rust toolchain, and teaches you the basics of reading and writing Rust syntax, on the command line. It's an alternative to Rust by Example that works with your own environment.

DO THE RUSTLINGS COURSE!

If reading multiple hundreds of pages about a language isn’t your style, then Rust By Example has you covered. While the book talks about code with a lot of words, RBE shows off a bunch of code, and keeps the talking to a minimum. It also includes exercises!

CHECK OUT RUST BY EXAMPLE!

https://miikka.me/talks/python-and-rust/

Tools you need:

- A library for interacting with Python: PyO3
- A build tool: Maturin or rustimport

PyO3

Rust bindings for Python

https://miikka.me/talks/python-and-rust/

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

/// Sums the integers in a list.
#[pyfunction]
fn sum(numbers: Vec<i64>) -> i64 {
    numbers.iter().sum()
}

/// A Python module implemented in Rust.
#[pymodule]
fn my_package(m: &Bound<'_, PyModule>) -> PyResult<()> {
    m.add_function(wrap_pyfunction!(sum, m)?)?;
    Ok(())
}
```

Maturin

A build tool for Python packages implemented in Rust.

```
$ maturin new my_package
$ eza --tree my_package
my_package
├── Cargo.toml
├── pyproject.toml
└── src
    └── lib.rs
```

```
$ maturin build
# ....
📦 Built wheel for CPython 3.13 to
/Users/miikka/mess/2025-38/my_package/target/wheels/my_package-0.1.0-cp313-
cp313-macosx_11_0_arm64.whl
```

Maturin has a build backend, so you can bring your own build frontend

```toml
# pyproject.toml
[build-system]
requires = ["maturin>=1.9,<2.0"]
build-backend = "maturin"
```

rustimport

Import Rust code from Python directly with `import`

https://miikka.me/talks/python-and-rust/

`my_rust_module.rs`:

```rust
// rustimport:pyo3

use pyo3::prelude::*;

#[pyfunction]
fn sum(numbers: Vec<i64>) -> i64 {
    numbers.iter().sum()
}
```

`my_python_script.py`:

```python
import rustimport.import_hook
import my_rust_module
print(my_rust_module.sum(range(100))
```

Maturin or rustimport?

Publishing a package? Maturin

Experimenting? rustimport

Otherwise? Both are great!

Will this make my code fast?

Maybe, but you have to benchmark.

Thanks for coming!

Scan the QR code for a page with slides and extra links.

miikka.me/talks/python-and-rust/

Blog: quanttype.net