Mojo🔥 Is All You Need

Daniil Sherki, Oleg Sautenkov, Mariia Kalinina

What is Mojo🔥?

- First introduced in **2023.**
- **Faster alternative to Python**, especially in the context of **DS and ML.**
- **Superset of Python**, addresses the performance limitations of Python while maintaining its simplicity and extensive ecosystem.
- Shows significant speed improvements, being **68000** times **faster** than Python.*

*According to the developers

01 Python Accelerator

- **Parallel processing across multiple cores.**
- **Achieve performance on par with C++ and CUDA**
- **Mojo does not have a garbage collector, instead it uses “move semantics” very similar to Rust.**
- **The language can interact with SIMD commands.**

LANGUAGES	TIME (S) *	SPEEDUP VS PYTHON
PYTHON 3.10.9	1027 s	1x
PYPY	46.1 s	22x
SCALAR C++	0.20 s	5000x
MOJO 🔥	0.03 s	68000x

02 Usability

PROGRESSIVE TYPES

Leverage types for better performance and error checking

ZERO COST ABSTRACTIONS

Storage control by inline-allocating values into structures

PORTABLE PARAMETRIC ALGORITHMS

Leverage compile-time meta-programming to write hardware-agnostic algorithms and reduce boilerplate

```
def exp[dt: DType, elts: Int]
    (x: SIMD[dt, elts]) -> SIMD[dt, elts]:
  x = clamp(x, -88.3762626647, 88.37626266)
  k = floor(x * INV_LN2 + 0.5)
  r = k * NEG_LN2 + x
  return ldexp(_exp_taylor(r), k)
```

03 Access to Python ecosystem

Intermix arbitrary libraries like **Numpy** and **Matplotlib** and the custom code with Mojo.

```
from python import Python

fn use_array() raises:
    # This is equivalent to Python's `import numpy as np`
    let np = Python.import_module("numpy")

    # Now use numpy as if writing in Python
    let array = np.array([1, 2, 3])
    print(array)
```

```
use_array()
```

```
[1 2 3]
```