Modern Python From Alternate Universes

Reece Jones

Python is very cool

…but so are other languages.

How can other languages inspire expressive and safe Python?

Overview

- Type hints
  - New minor features
  - Type parameters
  - Parameter specifications
  - Protocols
  - Deduced self type
- Dataclasses
- Return codes + pattern matching

Type Hints

Static typing

C, Java, Haskell, etc.

3.5

Annotate code with type information for static analysis

Highlights

- Detect bugs earlier
- Confidence in code safety
- Enhanced readability

Key uses

- Everywhere!

Lowlights

- Can't static type everything
- Tooling isn't perfect
- More dev effort

Type Hints - New Minor Features

`List[str]` → `list[str]`  *3.9*

`Union[int, str]` → `int | str`  *3.10*

Type Parameters

Generics

Java, C++, Haskell

3.5

Statically parameterise functions and classes by types

✨ Key uses

- Generic data structures
- Generic algorithms

Parameter Specifications

*Variadic generics**

C++

3.10

Static analysis for manipulating arbitrary function signatures

Key uses

- Function wrappers/decorators

Lowlights

- Limited parameter manipulation

*sort of

Protocols

Structural subtyping

Go, OCaml, C++

3.8

Statically analysable ad-hoc interface implementation

Highlights

- Simple interfaces
- Easily extendable interfaces

Key uses

- Widely-used interfaces

Lowlights

- Less explicit

Deduced Self Type

Self type

TypeScript, Rust, Scala

3.5 (3.11)

Statically capture derived type in base class

Key uses

- Method chaining / fluent interfaces
- Factory classmethods

Dataclasses

Record types

Elixir, C++, C#, Kotlin

3.7

Automatically generated record types

Key uses

- Functional programming
- Procedural programming

Highlights

- Simple & concise
- Immutability
- Flexible & customisable
- Statically analysable

Lowlights

- Not 100% statically analysable in practice

Return Codes

Return values

Erlang, Elixir

3.10

Return values and pattern matching for indicating errors

Highlights

- Explicit error modes
- Explicit control flow
- Clean error handling
- Statically analysable

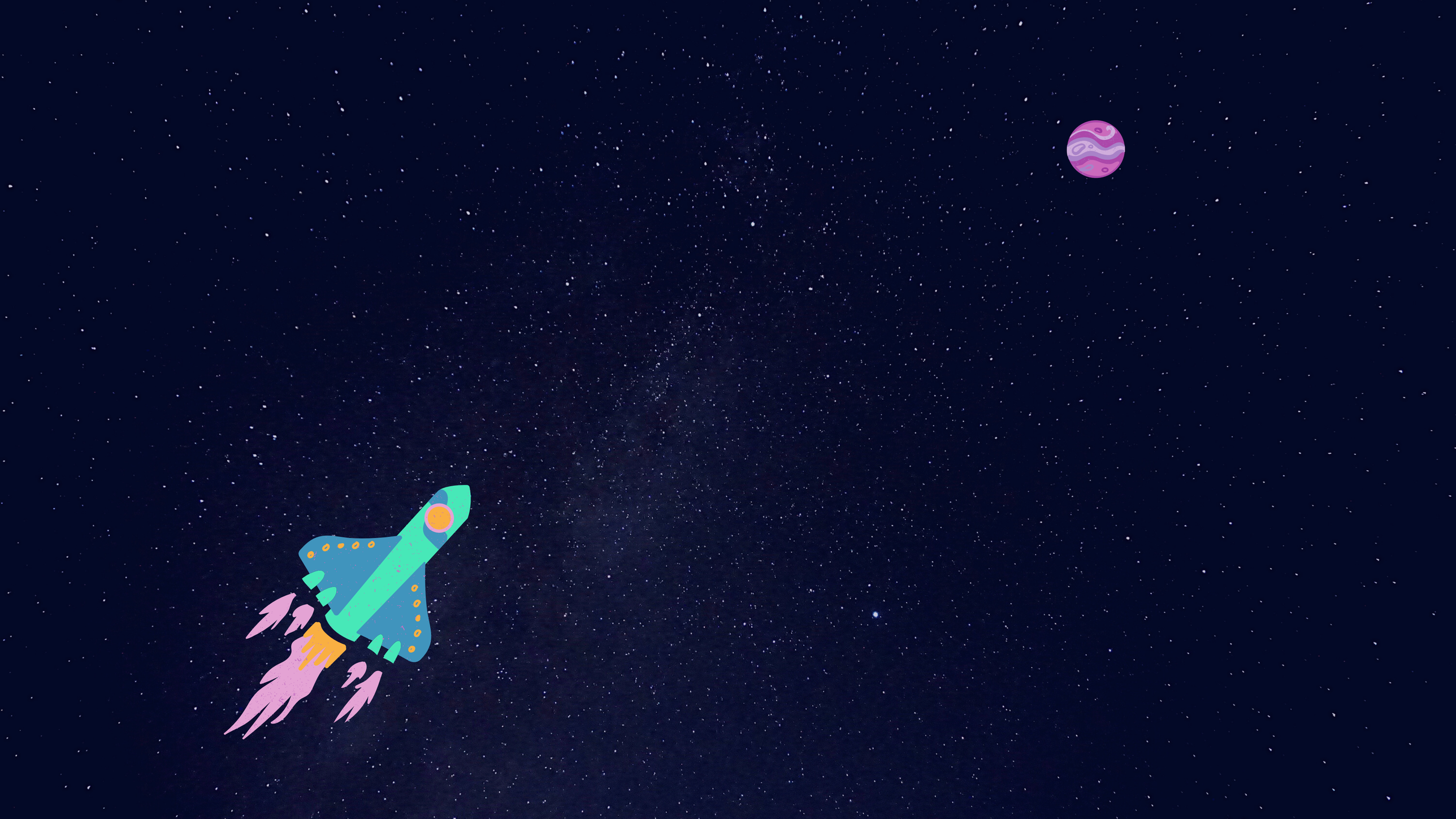
Key uses

- ?

Lowlights

- Weak pattern matching
- No Elixir "with"
- Tooling isn't great

} currently



Code examples repository:

https://github.com/MC-DeltaT/PythonFromAltUniverses

Me:

Email: reece.jones131@gmail.com

GitHub: https://github.com/MC-DeltaT

LinkedIn: https://www.linkedin.com/in/reece-jones-1327261a3/