Clojure

Language for old cranky programmers

Facts

- Dynamic functional language (LISP)
- First released 2007 by Rich Hickey
- Hosted language with full iterop
- Dynamic typing

Clojure ~ Python

vector

```clojure
["foo" 1 bar]
```

```python
["foo", 1, bar]
```

Clojure ~ Python

hashmap

```
{
  "foo" 1
  "bar" 2
}
```

```
{
  "foo": 1,
  "bar": 2
}
```

Clojure ~ Python

set

```clojure
#{"foo" 1 bar}
```

```python
{"foo", 1, bar}
```

Clojure × Python

functions

```clojure
(foo "bar" "baz")
```

```python
foo("bar", "baz")
```

Clojure × Python

operators

```
(+ 1 2 3)
```

```
1 + 2 + 3
```

Clojure × Python

keywords

Clojure × Python

```clojure
(every? filled?
        (filter mandatory?
                (:fields form)))
```

```clojure
(->> form
     :fields
     (filter mandatory?)
     (every? filled?))
```

Similarities

- **Simple language**
- **Well thought-out**
- **Dynamic typing**
- **Structural editing** (Parinfer)
- **Great tools**
- **Great community**

Advantages

- **Java** (fast + real threads)
- **Immutable by default** (easy parallelism)
- **REPL** (interactive development)
- **Uberjars** (Docker from the 90‘s)
- **Meta-programming** (powerful macros)
- **BE+FE code sharing**

Disadvantages

- **Java** (slow startup time)
- **Low adoption** (fewer jobs)
- **Learning curve**