# rpds.py

*Release 0.18.1*

**Julian Berman**

**May 06, 2024**

# CONTENTS

Python bindings to the Rust rpds crate for persistent data structures.

What's here is quite minimal (in transparency, it was written initially to support replacing `pyrsistent` in the referencing library). If you see something missing (which is very likely), a PR is definitely welcome to add it.

The distribution on PyPI is named `rpds.py` (equivalently `rpds-py`), and thus can be installed via e.g.:

```
$ pip install rpds-py
```

Note that if you install `rpds-py` from source, you will need a Rust toolchain installed, as it is a build-time dependency. An example of how to do so in a `Dockerfile` can be found here.

If you believe you are on a common platform which should have wheels built (i.e. and not need to compile from source), feel free to file an issue or pull request modifying the GitHub action used here to build wheels via `maturin`.

CHAPTER

ONE

# USAGE

Methods in general are named similarly to their `rpds` counterparts (rather than `pyrsistent`'s conventions, though probably a full drop-in `pyrsistent`-compatible wrapper module is a good addition at some point).

```
>>> from rpds import HashTrieMap, HashTrieSet, List

>>> m = HashTrieMap({"foo": "bar", "baz": "quux"})
>>> m.insert("spam", 37) == HashTrieMap({"foo": "bar", "baz": "quux", "spam": 37})
True
>>> m.remove("foo") == HashTrieMap({"baz": "quux"})
True

>>> s = HashTrieSet({"foo", "bar", "baz", "quux"})
>>> s.insert("spam") == HashTrieSet({"foo", "bar", "baz", "quux", "spam"})
True
>>> s.remove("foo") == HashTrieSet({"bar", "baz", "quux"})
True

>>> L = List([1, 3, 5])
>>> L.push_front(-1) == List([-1, 1, 3, 5])
True
>>> L.rest == List([3, 5])
True
```

## 1.1 API Reference

**class** rpds.**HashTrieMap**(*value=None*, ***kwds*)

**__getitem__**(*key*, */*)

Return self[key].

**__iter__**()

Implement iter(self).

**__len__**()

Return len(self).

**convert**()

**discard**(*key*)

**fromkeys**(*val=None*)

**get**(*key*, *default=None*)

**insert**(*key*, *value*)

**items**()

**keys**()

**remove**(*key*)

**update**(**maps*, ***kwds*)

**values**()

class rpds.**HashTrieSet**(*value=None*)

**__iter__**()
Implement iter(self).

**__len__**()
Return len(self).

**difference**(*other*)

**discard**(*value*)

**insert**(*value*)

**intersection**(*other*)

**remove**(*value*)

**symmetric_difference**(*other*)

**union**(*other*)

**update**(**iterables*)

class rpds.**List**(**elements*)

**__iter__**()
Implement iter(self).

**__len__**()
Return len(self).

**drop_first**()

**first**

**push_front**(*other*)

**rest**

class rpds.**Queue**(**elements*)

**__iter__**()
Implement iter(self).

**__len__**()

Return len(self).

**dequeue**()

**enqueue**(*value*)

**is_empty**

**peek**

# PYTHON MODULE INDEX

r

# INDEX