Collection Protocols

Robert Smallshire
COFOUNDER - SIXTY NORTH

@robsmallshire

Austin Bingham
COFOUNDER - SIXTY NORTH

@austin_bingham

Collection Protocols

Protocol	Implementing Collections
Container	`str list range tuple bytes dict set frozenset`
Sized	`str list range tuple bytes dict set frozenset`
Iterable	`str list range tuple bytes dict set frozenset`
Sequence	`str list range tuple bytes`
Set	`set frozenset`
Mapping	`dict`
Mutable Sequence	`list`
Mutable Set	`set`
Mutable Mapping	`dict`

Protocols

- To implement a protocol, objects must support certain operations.
- Most collections implement *container, sized*, and *iterable*.
- All except dict, set and frozenset are *sequences*.

Collection Protocols

Protocol	Implementing Collections
Container	`str list range tuple bytes dict set frozenset`
Sized	`str list range tuple bytes dict set frozenset`
Iterable	`str list range tuple bytes dict set frozenset`
Sequence	`str list range tuple bytes`
Set	`set frozenset`
Mapping	`dict`
Mutable Sequence	`list`
Mutable Set	`set`
Mutable Mapping	`dict`

Container Protocol

- Membership testing using the `in` and `not in` operators.

Collection Protocols

Protocol	Implementing Collections
Container	`str list range tuple bytes dict set frozenset`
Sized	`str list range tuple bytes dict set frozenset`
Iterable	`str list range tuple bytes dict set frozenset`
Sequence	`str list range tuple bytes`
Set	`set frozenset`
Mapping	`dict`
Mutable Sequence	`list`
Mutable Set	`set`
Mutable Mapping	`dict`

Sized Protocol

- Determine the number of elements with `len(s)`

Collection Protocols

Protocol	Implementing Collections
Container	`str list range tuple bytes dict set frozenset`
Sized	`str list range tuple bytes dict set frozenset`
Iterable	`str list range tuple bytes dict set frozenset`
Sequence	`str list range tuple bytes`
Set	`set frozenset`
Mapping	`dict`
Mutable Sequence	`list`
Mutable Set	`set`
Mutable Mapping	`dict`

Iterable Protocol

- Can produce an iterator with `iter(s)`

```
for item in iterable:
    do_something(item)
```

Collection Protocols

Protocol	Implementing Collections
Container	`str list range tuple bytes dict set frozenset`
Sized	`str list range`
Iterable	`str list range`
Sequence	`str list range`
Set	`set frozenset`
Mapping	`dict`
Mutable Sequence	`list`
Mutable Set	`set`
Mutable Mapping	`dict`

Sequence Protocol

- Retrieve elements by index

```
item = seq[index]
```

- Find items by value

```
index = seq.index(item)
```

- Count items

```
num = seq.count(item)
```

- Produce a reversed sequence

```
r = reversed(seq)
```

Collection Protocols

Protocol	Implementing Collections
Container	`str list range tuple bytes dict set frozenset`
Sized	`str list range tuple bytes dict set frozenset`
Iterable	`str list range tuple bytes dict set frozenset`
Sequence	`str list range`
Set	`set frozenset`
Mapping	`dict`
Mutable Sequence	`list`
Mutable Set	`set`
Mutable Mapping	`dict`

Set Protocol

Set algebra operations

(methods and infix operators)

- subset
- proper subset
- equal
- not equal
- proper superset
- superset
- intersection
- union
- symmetric difference
- difference

Collection Protocols

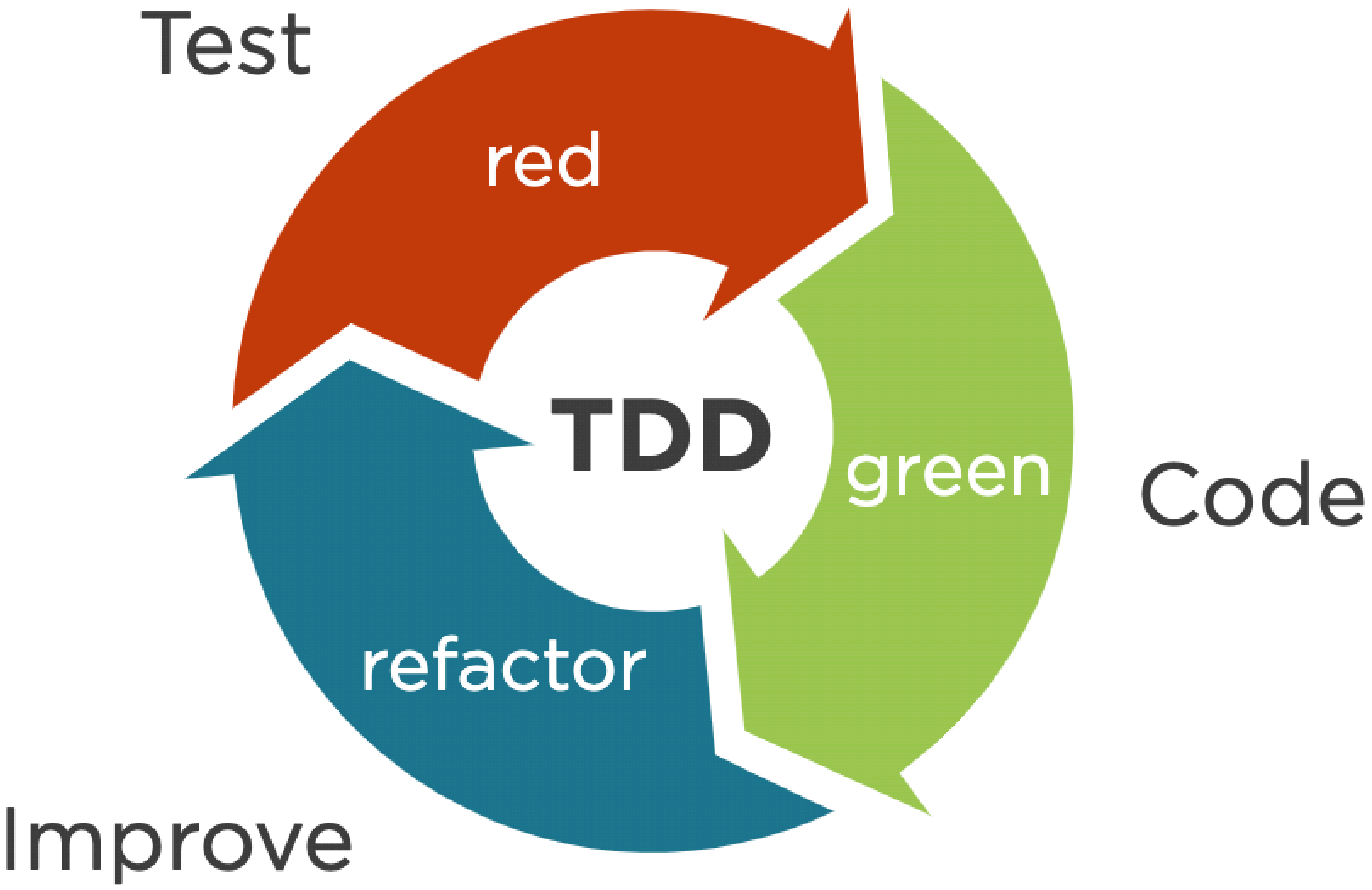
Protocol	Implementing Collections
Container	str list range tuple bytes dict set frozenset
Sized	str list range tuple bytes dict set frozenset
Iterable	str list range tuple bytes dict set frozenset
Sequence	str list range tuple bytes
Set	set frozenset
Mapping	dict
Mutable Sequence	list
Mutable Set	set
Mutable Mapping	dict

Let’s build a

SortedFrozenSet

A collection which is a sized, iterable, sequence container of a set of distinct items, and constructible from an iterable

Test Driven Development


Test
red
TDD
green
Code
refactor
Improve