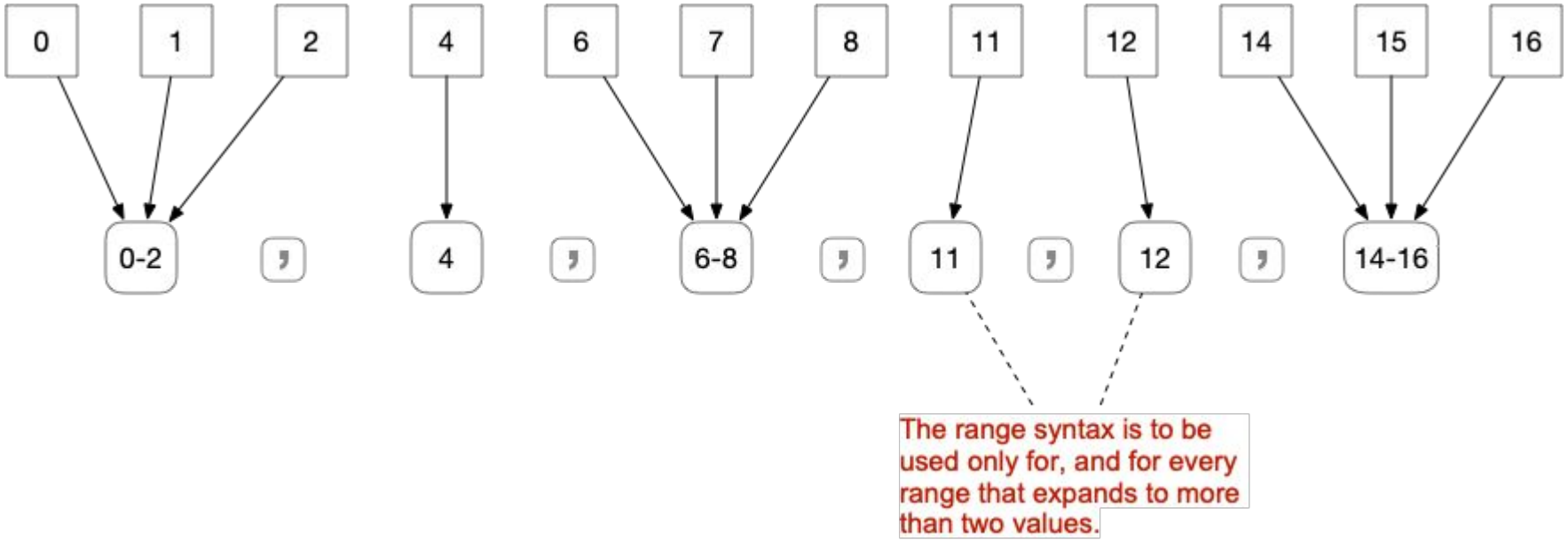
Object-Oriented Programming via Python: Recap Session 01

Stephen Leach, Sept 2021

Why We Care

- Write reusable code units (modules/libraries/services) that provide **behaviour**
- Whose implementation is **local to that unit**
- And you can write other units that **have overlapping behaviour** that are **interchangeable**

Running Example



Procedural

```
def range_extract(lst):
    'Yield 2-tuple ranges or 1-tuple single elements from list of increasing ints'
    lenlst = len(lst)
    i = 0
    while i< lenlst:
        low = lst[i]
        while i <lenlst-1 and lst[i]+1 == lst[i+1]: i +=1
        hi = lst[i]
        if   hi - low >= 2:
            yield (low, hi)
        elif hi - low == 1:
            yield (low,)
            yield (hi,)
        else:
            yield (low,)
        i += 1

def printr(ranges):
    print( ','.join( (('%i-%i' % r) if len(r) == 2 else '%i' % r)
                     for r in ranges ) )

if __name__ == '__main__':
    for lst in [[-8, -7, -6, -3, -2, -1, 0, 1, 3, 4, 5, 7,
                 8, 9, 10, 11, 14, 15, 17, 18, 19, 20],
                [0, 1, 2, 4, 6, 7, 8, 11, 12, 14, 15, 16, 17, 18, 19, 20, 21, 22,
                 23, 24, 25, 27, 28, 29, 30, 31, 32, 33, 35, 36, 37, 38, 39]]:
        #print(list(range_extract(lst)))
        printr(range_extract(lst))
```

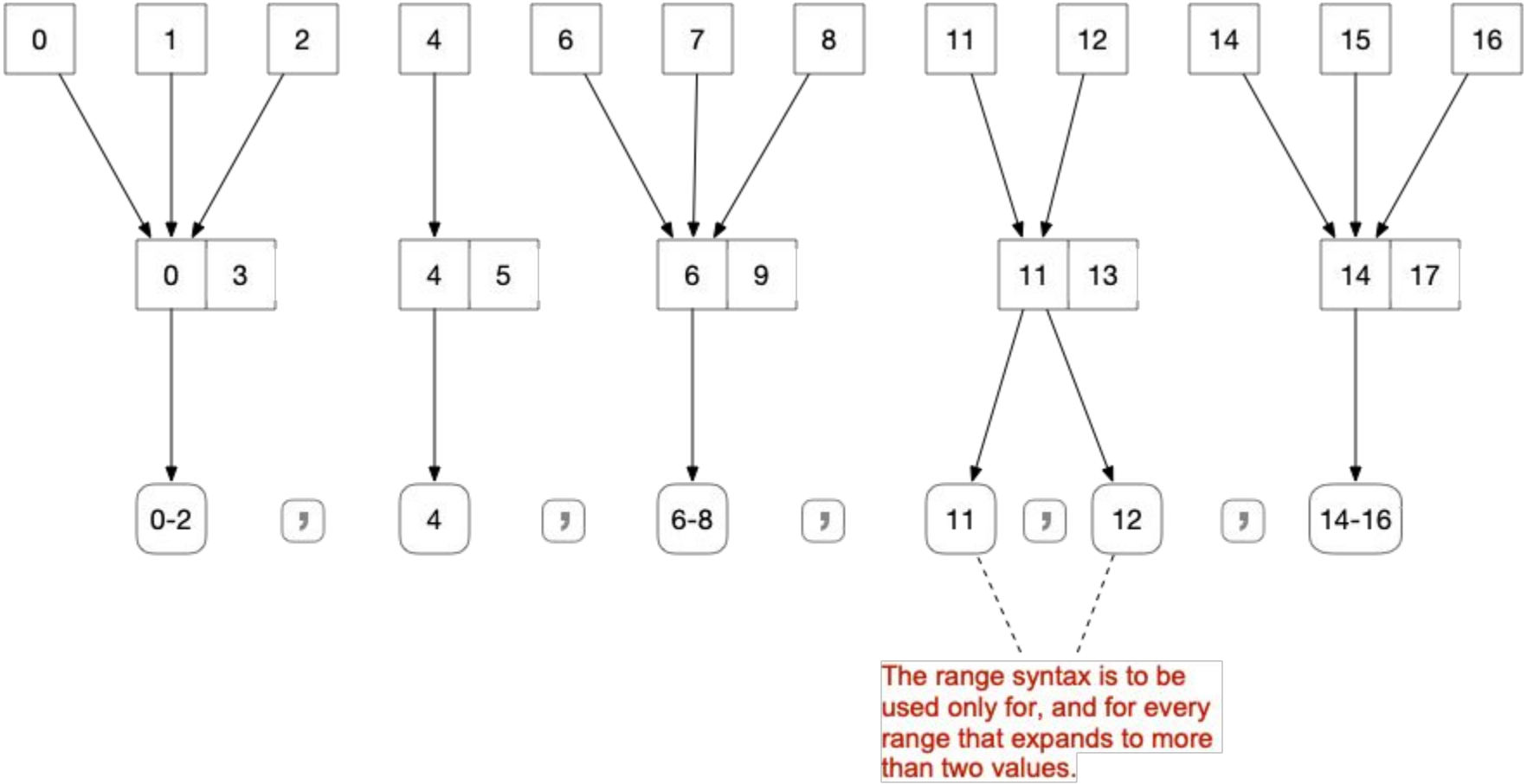
Complex intermediate state is held in fixed variables **lst**, **lenlst** and changing variables **i**, **low**, **hi**

Ranges are implemented as pairs or singles

Printr knows the implementation

Implied Implementation, repurposed built-ins

Behaviour	**Multi-page range**	**Single-page range**
Construction	(a, b)	(a,)
Case distinction	len(x) == 2	len(x) == 1
Low and High	x.iter()	*Error*
Low	*Error*	x.iter()


0
1
2
4
6
7
8
11
12
14
15
16
0
3
4
5
6
9
11
13
14
17
0-2
,
4
,
6-8
,
11
,
12
,
14-16
The range syntax is to be used only for, and for every range that expands to more than two values.

Explicit, Custom Implementation

Behaviour	**RangeOfPages**
Construction	RangeOfPages(a, length=1)
Low	x.start()
High	x.finish()
Try Extend	x.try_add(page: int)

Behaviour vs *Implementation*

Description in the language and terminology of the **Problem Domain** (aka Business Domain)

Description in the language and terminology of our **programming language and tools**