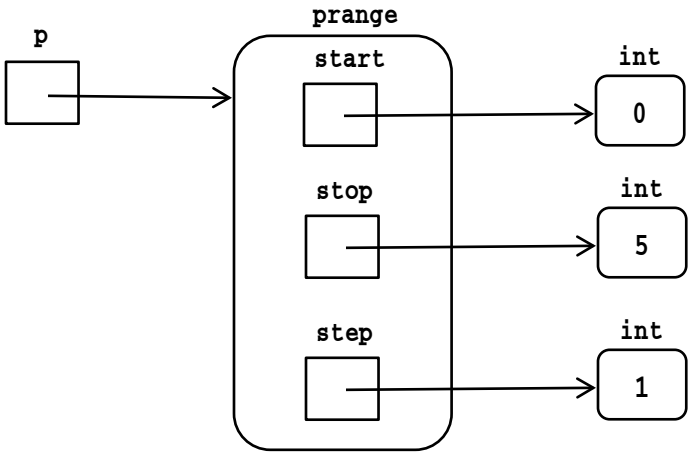
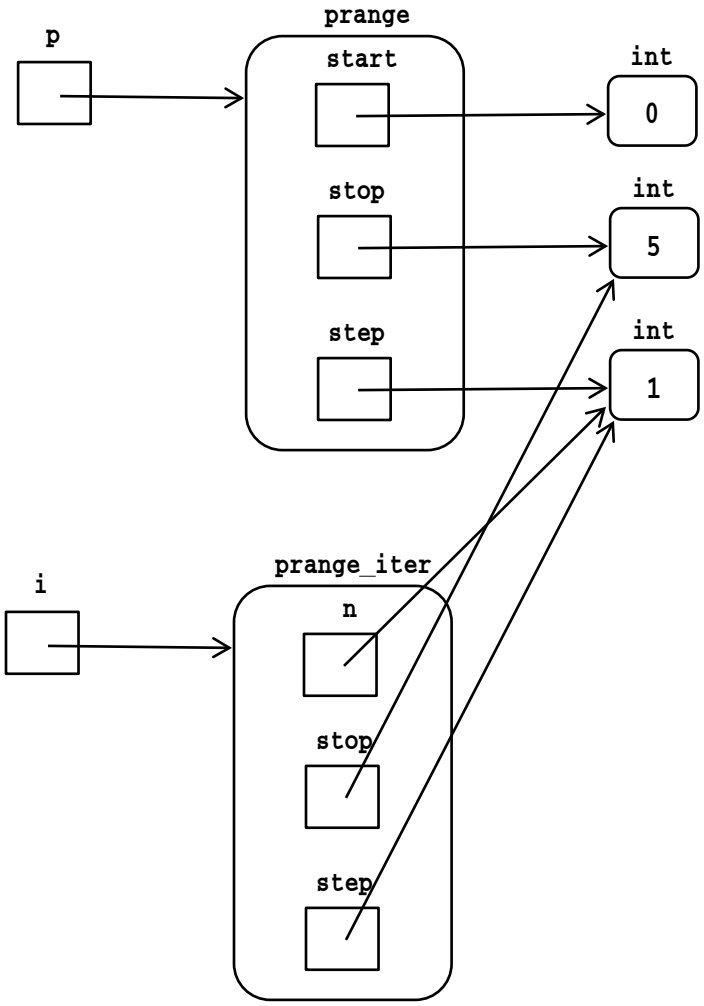
prange with nested prange_iter class

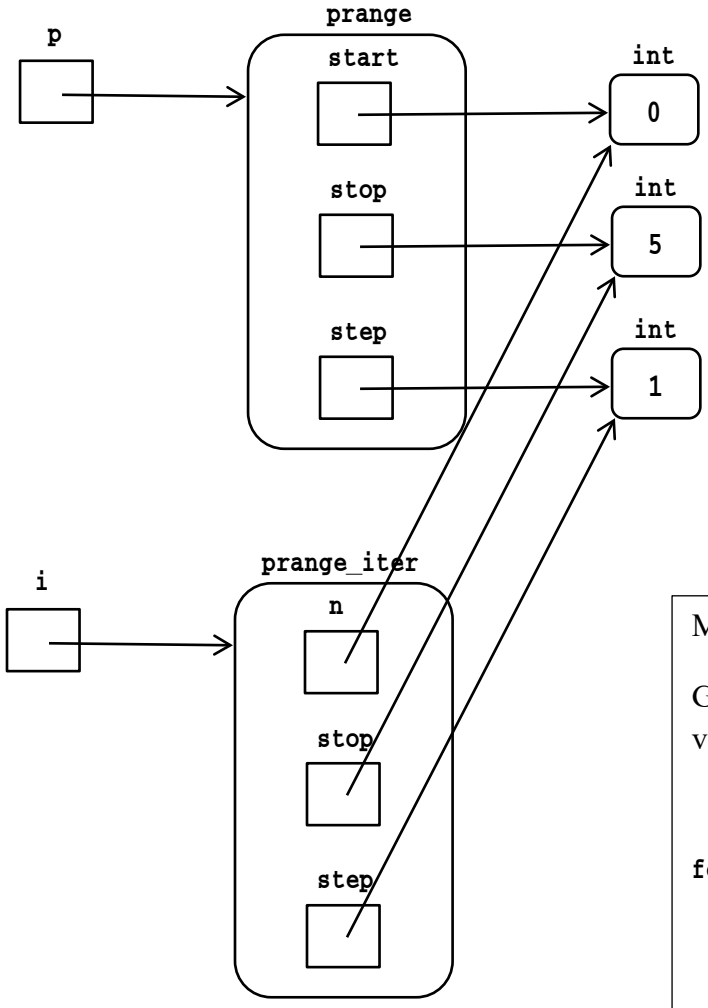
```
p = prange(0,5,1)
```




```
print(next(i)) # prints 0
```




```
i = iter(p)
```



Multiple iterators will produce **different** `prange_iter` objects.

Given this behavior, the code on the left will print the same values as the code on the right.

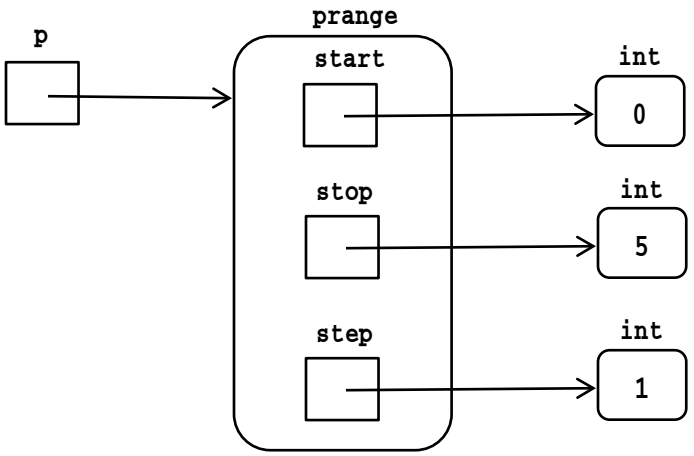
```
for a in prange(0,5,1):
    for b in prange(0,5,1):
        print(a,b)
```

```
p = prange(0,5,1)
for a in p:
    for b in p:
        print(a,b)
```

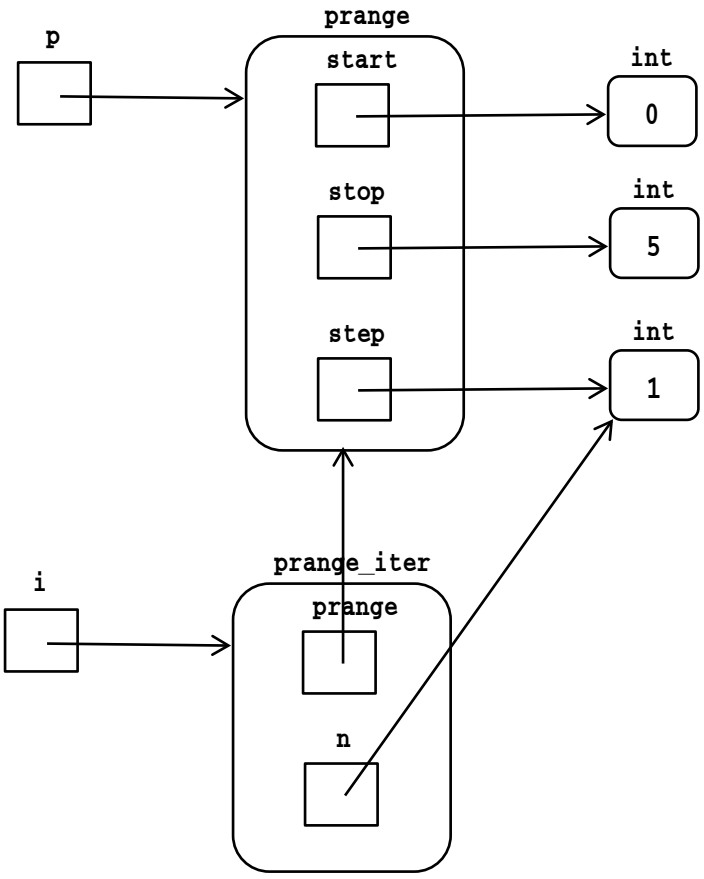
prange with nested alternative prange_iter class

(here the prange attribute in a prange_iter object refers to the prange object being iterated over: so its start/stop/step can be indirectly accessed via the prange attribute in the prange_iter object)

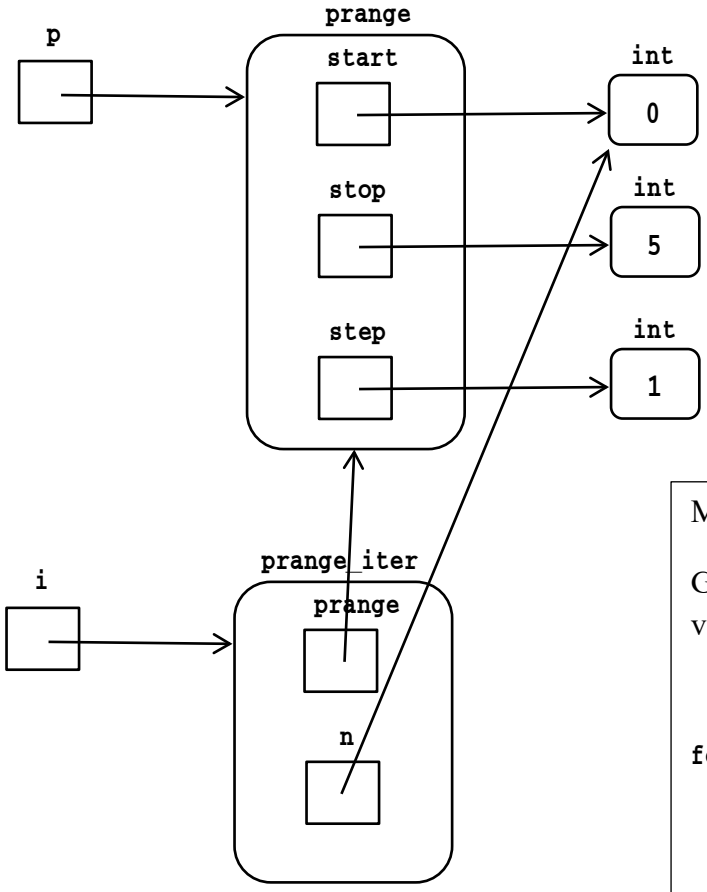
```
p = prange(0,5,1)
```




```
print(next(i)) # prints 0
```




```
i = iter(p)
```



Multiple iterators will produce **different** `prange_iter` objects.

Given this behavior, the code on the left will print the same values as the code on the right.

```
for a in prange(0,5,1):
    for b in prange(0,5,1):
        print(a,b)
```

```
p = prange(0,5,1)
for a in p:
    for b in p:
        print(a,b)
```