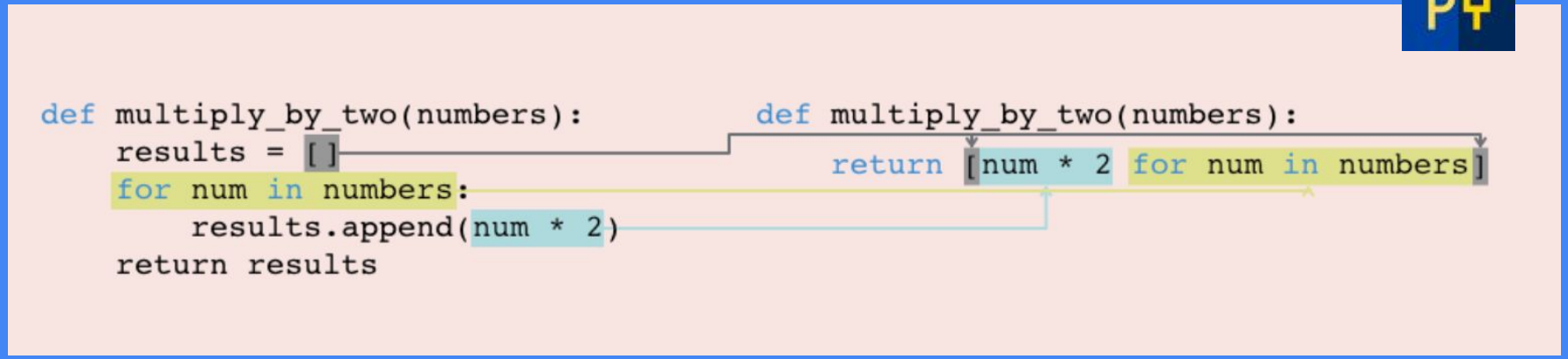
HACK REACTOR

M2 Practice Problems

Week 1 | Day 1 | **Python List Comprehensions**

How to Write a List Comprehension



```
def multiply_by_two(numbers):
    results = []
    for num in numbers:
        results.append(num * 2)
    return results
```

```
def multiply_by_two(numbers):
    return [num * 2 for num in numbers]
```

Step One: _____

- **Define the data structure being returned:**
  - Is it a list? [ ]
  - Is it a list of dictionaries? [{ }, { }]

```
def multiply_by_two(numbers):
    results = []
    for num in numbers:
        results.append(num * 2)
    return results
```

```
def multiply_by_two(numbers):
    return [num * 2 for num in numbers]
```

Step Two:

- Define the operation that is being iterated on each item in the list

```
def multiply_by_two(numbers):
    results = []
    for num in numbers:
        results.append(num * 2)
    return results
```

```
def multiply_by_two(numbers):
    return [num * 2 for num in numbers]
```

Step Three:

- Afterwards, add the iterable expression (for loop)

List Comprehensions

- **Syntactic Sugar**: elegant one-liner
  - Doesn't really change the underlying for loop mechanism or time complexity
- **Returns a new list**
  - Old list is unchanged
- **Python comprehensions work on any iterable:**
  - List, Object, Tuple, Set, range( ), etc.