

A photograph of a tall building at dusk. The building's facade is illuminated with the text "AFRO PIX 2026" in large, glowing purple letters. At the top of the building is a silhouette of a person's head with a crown. The sky is a deep purple. Other skyscrapers are visible in the background. A yellow school bus is partially visible at the bottom left.

# AFRO PIX 2026

COMMUNITY CENTER

FEBRUARY 28, 2026

REGISTRATION OPEN NOW



# Reference Types & Practice with Lists

# Announcements

## Re: Quiz 01

- Great job!! Median was ~85%
- Graded quizzes will be available on Gradescope soon! Once they're released:
  - *Please review what you missed ASAP*; we will build on the topics covered in Quiz 01 throughout the course, and these foundational concepts are vital!
  - Don't understand a particular question/part of a memory diagram? Please visit us in Office Hours or Tutoring! Full list of hours on the site's [support page](#)
  - Please submit a regrade request *if you believe your quiz was not graded correctly according to the rubric*

# Primitive Types vs. Reference Types

```
1  """An example of primitive vs. reference types."""
2
3  a: int = 0
4  b: int = a
5  b += 1
6  print(f"a is: {a}")
7  print(f"b is: {b}")
8
9  c: list[int] = [b, 4]
10 d: list[int] = c
11
12 d.append(13)
13 c.pop(1)
14 d[0] += 1
15
16 print(f"c is: {c}")
17 print(f"d is: {d}")
```

# Takeaways

```
1  """An example of primitive vs. reference types."""
2
3  a: int = 0
4  b: int = a
5  b += 1
6  print(f"a is: {a}")
7  print(f"b is: {b}")
8
9  c: list[int] = [b, 4]
10 d: list[int] = c
11
12 d.append(13)
13 c.pop(1)
14 d[0] += 1
15
16 print(f"c is: {c}")
17 print(f"d is: {d}")
```

Your turn: write a function called `gen` that takes as input an int called `n`, and returns a `list` of integers from 0 up to, but not including `n`

```
1 def gen(stop: int) -> list[int]:
2     """Generate a list from 0 to stop, not inclusive."""
3     i: int = 0
4     acc: list[int] = []
5     while i < stop:
6         acc.append(i)
7         i = i + 1
8     return acc
9
10
11 print(gen(3))
```

# Diagramming a Nested List

```
1  def sum2d(xs: list[list[int]]) -> int:
2      """Calculate the sum of a 2-dimensional list of lists."""
3      total: int = 0
4      row_i: int = 0
5      while row_i < len(xs):
6          col_i: int = 0
7          while col_i < len(xs[row_i]):
8              total += xs[row_i][col_i]
9              col_i += 1
10         row_i += 1
11     return total
12
13
14     values: list[list[int]] = [[1, 2, 3], [3, 4, 5]]
15     print(sum2d(values))
```