

Iterators and for..in Loops

Today starts as a Paper + Pencil or Tablet + Pencil day... please keep laptops stowed away!

COMP110 - CL18

2024/04/09

Warm-up: Evaluate and respond to GS questions...

This will be today's attendance submission (CL18). Submit once complete.

```
1  def a(x: int) -> int:
2  |     return x * 2
3
4
5  def b(x: int) -> bool:
6  |     return x % 2 == 0
7
8
9  xs: list[int] = [1, 2, 3, 4, 5, 6]
10 filtered: list[int] = list(filter(b, xs))
11 mapped_a: list[int] = list(map(a, filtered))
12
13 mapped_b: list[int] = list(map(b, xs))
```

Warm-up #2: Fill in the blanks...

```
1  from typing import Callable, TypeVar
2
3  T = TypeVar("T")
4  U = TypeVar("U")
5  Transform = Callable[[T], U]
6
7
8  def compose(f: Transform[int, float], g: Transform[float, str], x: int) -> str:
9      f_rv: float = f(x)
10     return g(f_rv)
11
12
13  def a(x: float) -> str:
14     return f"x is {x}"
15
16
17  def b(x: int) -> float:
18     return x / 2.0
19
20
21  print(compose(_____, _____, 110))
```

Warm-up #3: Trace a Memory Diagram

Reminder: Ignore Imports, TypeVars, and Type Aliases in Diagrams

```
1  from typing import Callable, TypeVar
2
3  T = TypeVar("T")
4  U = TypeVar("U")
5  Transform = Callable[[T], U]
6
7
8  def compose(f: Transform[int, float], g: Transform[float, str], x: int) -> str:
9      |   f_rv: float = f(x)
10     |   return g(f_rv)
11
12
13  def a(x: float) -> str:
14     |   return f"x is {x}"
15
16
17  def b(x: int) -> float:
18     |   return x / 2.0
19
20
21  print(compose(b, a, 110))
```

```
1 from typing import Callable, TypeVar
2
3 T = TypeVar("T")
4 U = TypeVar("U")
5 Transform = Callable[[T], U]
6
7
8 def compose(
9     f: Transform[int, float],
10    g: Transform[float, str],
11    x: int
12 ) -> str:
13     f_rv: float = f(x)
14     return g(f_rv)
15
16
17 def a(x: float) -> str:
18     return f"x is {x}"
19
20
21 def b(x: int) -> float:
22     return x / 2.0
23
24
25 print(compose(b, a, 110))
```

Introducing the range Type

The for..in Loop with range

The for..in Loop with Lists

Exploration of Iterables and Iterators