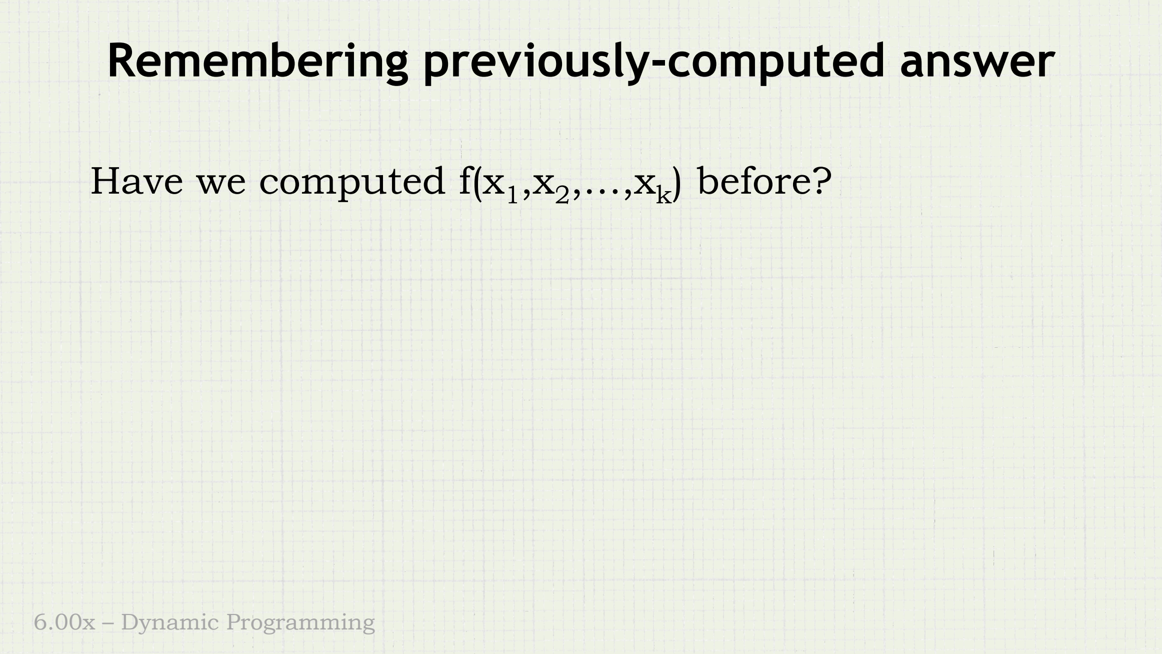
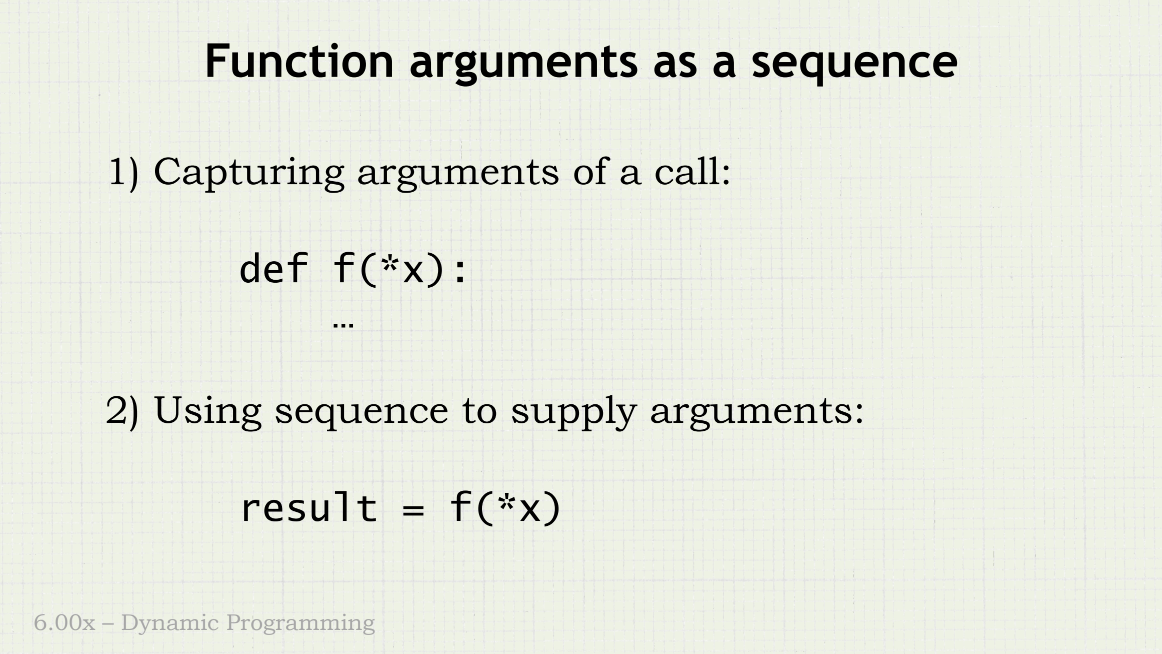


Remembering previously-computed answer

Have we computed $f(x_1,x_2,\ldots,x_k)$ before?



Function arguments as a sequence

1) Capturing arguments of a call:

```
def f(*x):
    …
```

2) Using sequence to supply arguments:

```
result = f(*x)
```

Wrapper function

```
# works for functions with hashable (immuatble) arguments
# Example usage: fib = memoize(fib)
def memoize(f):

    def memf(*x):
        if x not in memf.cache:
            memf.cache[x] = f(*x)
        return memf.cache[x]

    memf.cache = {}
    return memf
```

“Memoization”

```
# works for functions with hashable (immuatble) arguments
# Example usage: fib = memoize(fib)
def memoize(f):

    def memf(*x):
        if x not in memf.cache:
            memf.cache[x] = f(*x)
        return memf.cache[x]

    memf.cache = {}
    return memf
```