Iteration
Announcements
Office Hours: You Should Go!
Office Hours: You Should Go!

You are not alone!
Office Hours: You Should Go!

You are not alone!
Office Hours: You Should Go!

You are not alone!

http://cs61a.org/office-hours.html
Iteration Example: Fibonacci Numbers
The Fibonacci Sequence
The Fibonacci Sequence
The Fibonacci Sequence

0, 1, 1, 2, 3, 5, 8, 13, 21, 34, 55, 89, 144, 233, 377, 610, 987
The Fibonacci Sequence

0, 1, 1, 2, 3, 5, 8, 13, 21, 34, 55, 89, 144, 233, 377, 610, 987
The Fibonacci Sequence

0, 1, 1, 2, 3, 5, 8, 13, 21, 34, 55, 89, 144, 233, 377, 610, 987
The Fibonacci Sequence

0, 1, 1, 2, 3, 5, 8, 13, 21, 34, 55, 89, 144, 233, 377, 610, 987
The Fibonacci Sequence

```python
def fib(n):
    """Compute the nth Fibonacci number."

>>> fib(0)
0
>>> fib(8)
21
"""
```
The Fibonacci Sequence

```python
def fib(n):
    """Compute the nth Fibonacci number."

    >>> fib(0)
    0
    >>> fib(8)
    21
    """
```
The Fibonacci Sequence

def fib(n):
    """Compute the nth Fibonacci number."

    >>> fib(0)
    0
    >>> fib(8)
    21
    """
The Fibonacci Sequence

```python
def fib(n):
    """Compute the nth Fibonacci number."

    >>> fib(0)
    0
    >>> fib(8)
    21
    """
    k, kth, difference = 0, 0, 1
    while k < n:
        kth, difference = kth + difference, kth
        k = k + 1
    return kth

```

The Fibonacci Sequence: 0, 1, 1, 2, 3, 5, 8, 13, 21, 34, 55, 89, 144, 233, 377, 610, 987...
def fib(n):
    """Compute the nth Fibonacci number."

    >>> fib(0)
    0
    >>> fib(8)
    21

    k, kth, difference = 0, 0, 1
    while k < n:
        kth, difference = kth + difference, kth
        k = k + 1
    return kth
def fib(n):
    """Compute the nth Fibonacci number."

    >>> fib(0)
    0
    >>> fib(8)
    21

    k, kth, difference = 0, 0, 1
    while k < n:
        kth, difference = kth + difference, kth
        k = k + 1
    return kth
def fib(n):
    """Compute the nth Fibonacci number."
    >>> fib(0)
    0
    >>> fib(8)
    21
    k, kth, difference = 0, 0, 1
    while k < n:
        kth, difference = kth + difference, kth
        k = k + 1
    return kth

old_diff = difference
difference = kth
kth = kth + old_diff

kth = kth + difference
difference = kth
Return
Return Statements
Return Statements

A return statement completes the evaluation of a call expression and provides its value
Return Statements

A return statement completes the evaluation of a call expression and provides its value $f(x)$ for user-defined function $f$: switch to a new environment; execute $f$'s body.
Return Statements

A return statement completes the evaluation of a call expression and provides its value $f(x)$ for user-defined function $f$: switch to a new environment; execute $f$'s body

return statement within $f$: switch back to the previous environment; $f(x)$ now has a value
Return Statements

A return statement completes the evaluation of a call expression and provides its value for user-defined function $f$: switch to a new environment; execute $f$'s body

`return` statement within $f$: switch back to the previous environment; $f(x)$ now has a value

Only one return statement is ever executed while executing the body of a function
Return Statements

A return statement completes the evaluation of a call expression and provides its value 
\( f(x) \) for user-defined function \( f \): switch to a new environment; execute \( f \)'s body 

\texttt{return} statement within \( f \): switch back to the previous environment; \( f(x) \) now has a value

Only one return statement is ever executed while executing the body of a function

\begin{verbatim}
def end(n, d):
    """Print the final digits of N in reverse order until D is found."

    >>> end(34567, 5)
    7
    6
    5
    """
\end{verbatim}
Return Statements

A return statement completes the evaluation of a call expression and provides its value f(x) for user-defined function f: switch to a new environment; execute f's body

return statement within f: switch back to the previous environment; f(x) now has a value

Only one return statement is ever executed while executing the body of a function

```python
def end(n, d):
    """Print the final digits of N in reverse order until D is found."""

    >>> end(34567, 5)
    7
    6
    5
    """
    while n > 0:
        last, n = n % 10, n // 10
    print(last)
```
Return Statements

A return statement completes the evaluation of a call expression and provides its value \( f(x) \) for user-defined function \( f \): switch to a new environment; execute \( f \)'s body

```
    return statement within f: switch back to the previous environment; \( f(x) \) now has a value
```

Only one return statement is ever executed while executing the body of a function

```
def end(n, d):
    """Print the final digits of N in reverse order until D is found."

    >>> end(34567, 5)
    7
    6
    5
    """
    while n > 0:
        last, n = n % 10, n // 10
        print(last)
        if d == last:
            return None
```
Return Statements

A return statement completes the evaluation of a call expression and provides its value \( f(x) \) for user-defined function \( f \): switch to a new environment; execute \( f \)'s body

```
return
```
statement within \( f \): switch back to the previous environment; \( f(x) \) now has a value

Only one return statement is ever executed while executing the body of a function

```python
def end(n, d):
    """Print the final digits of N in reverse order until D is found."

>>> end(34567, 5)
7
6
5
""
while n > 0:
    last, n = n % 10, n // 10
    print(last)
if d == last:
    return None
(Demo)
```
Self-Reference (Demo)
Returning a Function Using Its Own Name

1. def print_sums(n):
   2.     print(n)
   3.     def next_sum(k):
   4.         return print_sums(n+k)
   5.     return next_sum
   6. 
   7. print_sums(1)(3)(5)
Function Example: Sounds
WAV Files

https://en.wikipedia.org/wiki/Triangle_wave
https://en.wikipedia.org/wiki/Sampling_(signal_processing)
WAV Files

The Waveform Audio File Format encodes a sampled sound wave

https://en.wikipedia.org/wiki/Triangle_wave
https://en.wikipedia.org/wiki/Sampling_(signal_processing)
WAV Files

The Waveform Audio File Format encodes a sampled sound wave

https://en.wikipedia.org/wiki/Triangle_wave
https://en.wikipedia.org/wiki/Sampling_(signal_processing)
WAV Files

The Waveform Audio File Format encodes a sampled sound wave.

A triangle wave is the simple waveform with the most pleasing sound.

https://en.wikipedia.org/wiki/Triangle_wave
https://en.wikipedia.org/wiki/Sampling_(signal_processing)
WAV Files

The Waveform Audio File Format encodes a sampled sound wave.

A triangle wave is the simple wave form with the most pleasing sound.

- Sine
- Square
- Triangle
- Sawtooth

WAV Files

The Waveform Audio File Format encodes a sampled sound wave

A triangle wave is the simple waveform with the most pleasing sound

(Demo)

https://en.wikipedia.org/wiki/Triangle_wave
https://en.wikipedia.org/wiki/Sampling_(signal_processing)