

UC Berkeley EECS
Lecturer
Michael Hall

UC Berkeley EECS
Adj. Ass. Prof.
Dr. Gerald Friedland

Computational Structures in Data Science

Lecture 3: Loops and Functions

UC Berkeley | Computer Science 88 | Michael Hall

1

Announcements

- Waitlist: We expect 10% to get off, but it's unknown.
- Attend Labs!

UC Berkeley | Computer Science 88 | Michael Hall

2

Things You Can Do Now

- Write a program that makes a decision.
- Write your own functions
- Use loops so you can process lots of data.

UC Berkeley | Computer Science 88 | Michael Hall

3

Let's Talk About Python

- Expression `3.1 * 2.6`
- Call expression `max(0, x)`
- Variables
- Assignment Statement `x = <expression>`
- Define Function: `def <function name> (<parameter list>):`
- Control Statements: `if ...`
`while ...`

UC Berkeley | Computer Science 88 | Michael Hall

4

Q&A

UC Berkeley | Computer Science 88 | Michael Hall

5

Defining Functions

```
def <function name> (<argument list>) :
```

↓ ↓

return expression

- Abstracts an expression or set of statements to apply to lots of instances of the problem
- A function should *do one thing well*

UC Berkeley | Computer Science 88 | Michael Hall

6

Doctests

- Write the docstring to explain *what* it does
 - What does the function return? What are corner cases for parameters?
- Write doctest to show what it should do
 - Before you write the implementation.
 - `python3 -m doctest [-v] file.py`

UC Berkeley | Computer Science 88 | Michael Hall

7

Returns and Values

- All functions always return SOME value.
- If you don't specify `return`, the value is `None`.

UC Berkeley | Computer Science 88 | Michael Hall

8

Conditional Statement

- Do some statements, conditional on a *predicate* expression

```
if <predicate>:
    <>true statements>
else:
    <>false statements>
```

- Example:

```
if (temperature > 98.6):
    print("fever!")
else:
    print("no fever")
```

UC Berkeley | Computer Science 88 | Michael Hall

9

while Statement – Iteration Control

- Repeat a block of statements until a predicate expression is satisfied

```
<initialization statements>
while <predicate expression>:
    <body statements>

<rest of the program>
```

```
def first_primes(k):
    """ Return the first k primes. """
    primes = []
    num = 2
    while len(primes) < k:
        if prime(num):
            primes = primes + [num]
            num = num + 1
    return primes
```

UC Berkeley | Computer Science 88 | Michael Hall

10

Functions and Arguments

```
>>> x = 3
>>> y = 4 + max(17, x + 4) * 0.5
>>> z = x + y
>>> print(z)
15.5
```

```
def max(x, y):
    return x if x > y else y

def max(x, y):
    if x > y:
        return x
    else:
        return y
```

UC Berkeley | Computer Science 88 | Michael Hall

11