**CS61A Lecture 2**

Amir Kamil
UC Berkeley
January 25, 2013

---

## Announcements
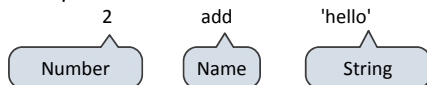
- Reminder: hw0 due tonight!
  - Read html file for instructions
  - Use our .py templates for homework and projects

- Find answers to your questions on Piazza
  - Use search functionality
  - Don't post private question unless it is not of general interest
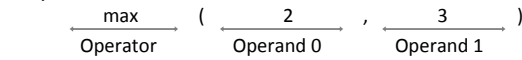
- In-class quiz next Friday
  - Bring a writing implement

---

## Review: Expressions

Primitive expressions:

2 — Number
add — Name
'hello' — String

Call expressions:

max ( 2 , 3 )
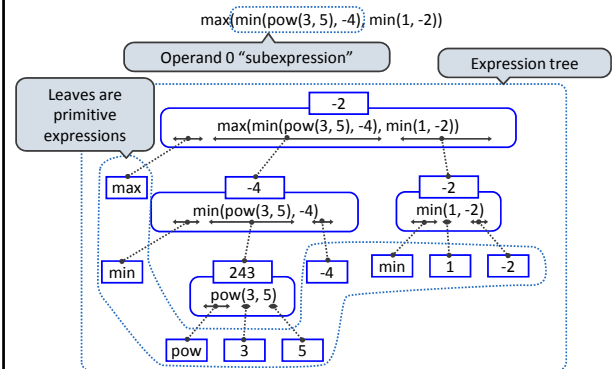Operator — Operand 0 — Operand 1

One big nested call expression

max(min(pow(3, 5), -4), min(1, -2))

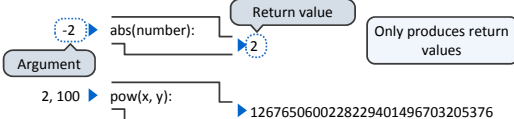Infix operators represent implicit call expressions

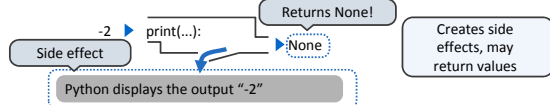2 + 3  →  add(2, 3)

---

## Review: Evaluation Procedure



max(min(pow(3, 5), -4), min(1, -2))

Operand 0 "subexpression"

Expression tree

Leaves are primitive expressions

-2
max(min(pow(3, 5), -4), min(1, -2))

max
-4
min(pow(3, 5), -4)

-2
min(1, -2)

min
243
pow(3, 5)
-4
min 1 -2

pow 3 5

---

## Types of Functions

Pure Functions

-2 ▶ abs(number): ▶ 2
Argument
Return value
Only produces return values

2, 100 ▶ pow(x, y): ▶ 1267650600228229401496703205376

Non-Pure Functions

-2 ▶ print(…): ▶ None
Side effect
Returns None!
Creates side effects, may return values

Python displays the output "-2"

The interactive interpreter displays all return values except None.

---

## Nested Print Expressions

None, None ▶ print(…): ▶ None

display "None None"

```
>>> print(print(1), print(2))
1
2
None None
```

None
print(print(1), print(2))

print
None
print(1)

None
print(2)

print 1
print 2

1 ▶ print(…): ▶ None
2 ▶ print(…): ▶ None

display "1"
display "2"

1
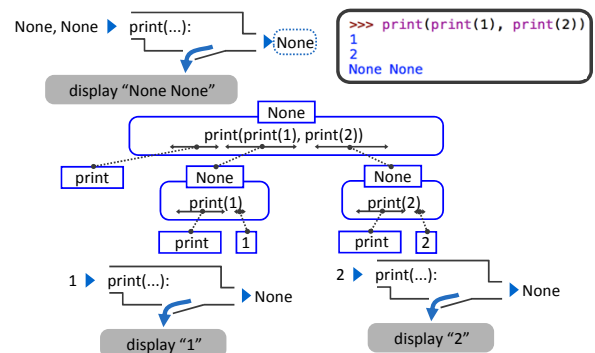
## The Elements of Programming

- Primitive Expressions and Statements
  - The simplest building blocks of a language

- Means of Combination
  - Compound elements built from simpler ones

- Means of Abstraction
  - Elements can be named and manipulated as units