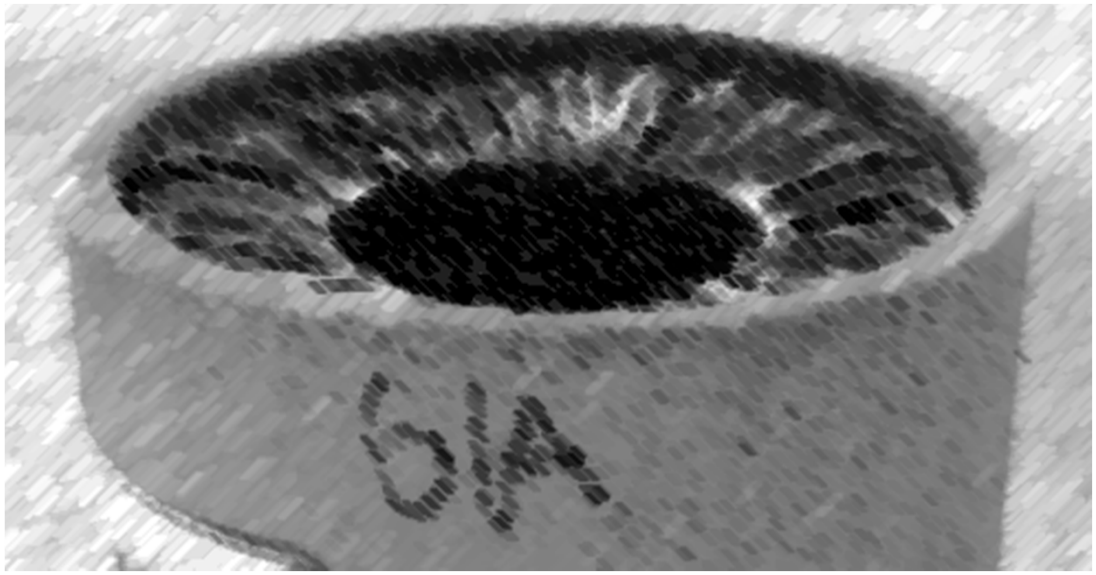


CS61A Lecture 2

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

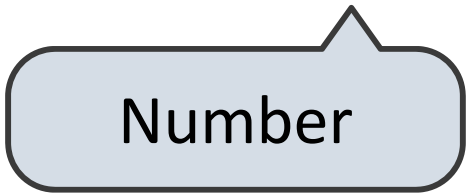
Review: Expressions

Primitive expressions:

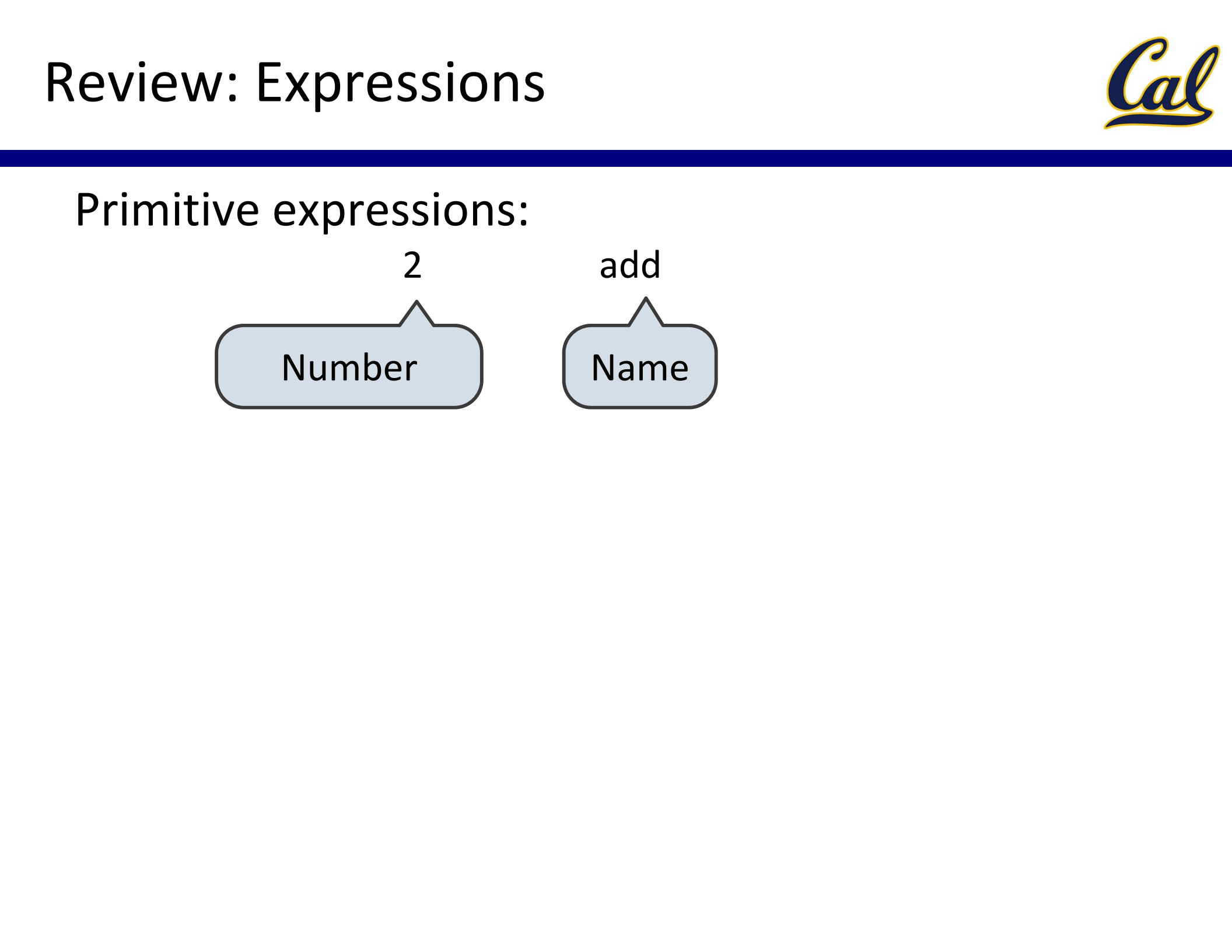
Review: Expressions

Primitive expressions:

2

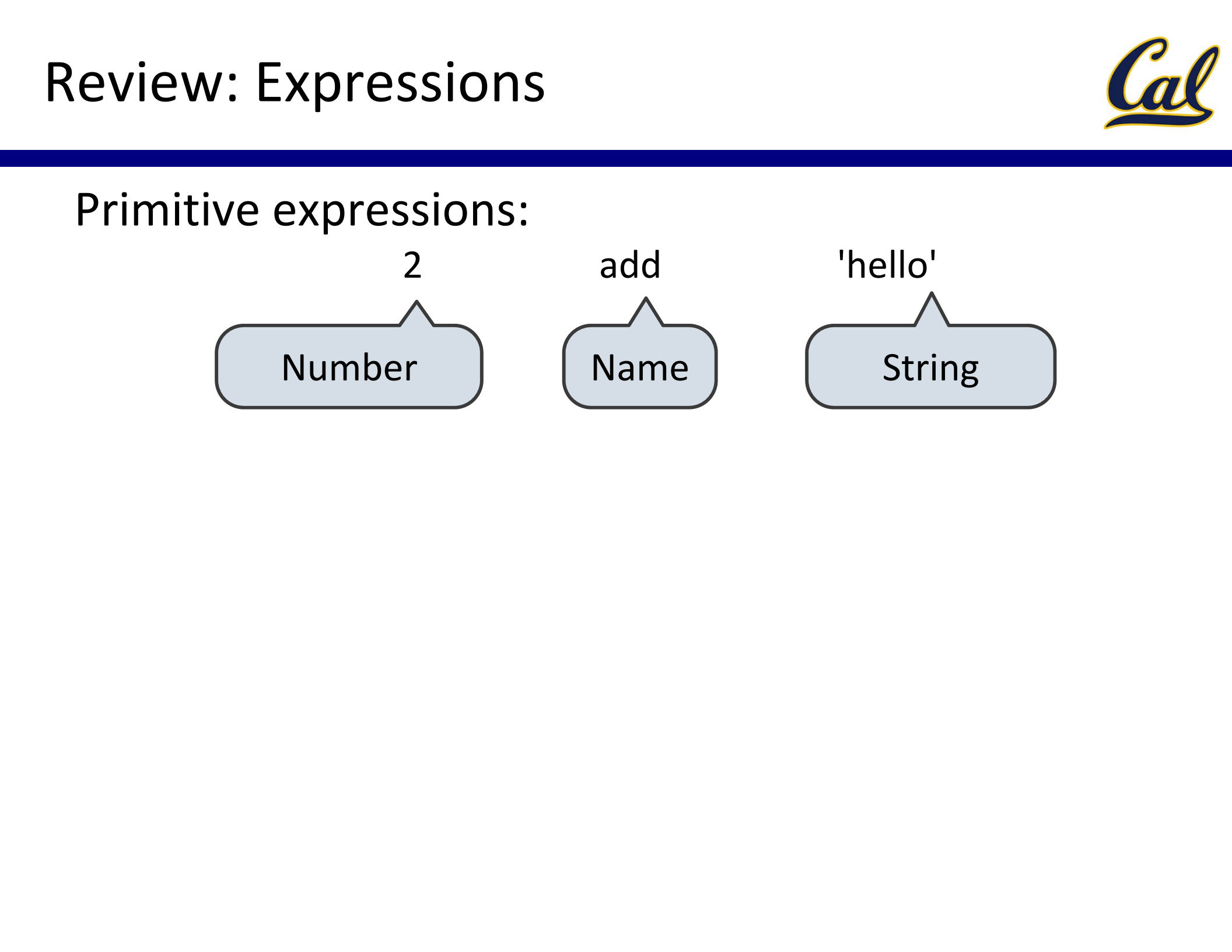


Review: Expressions



Primitive expressions:

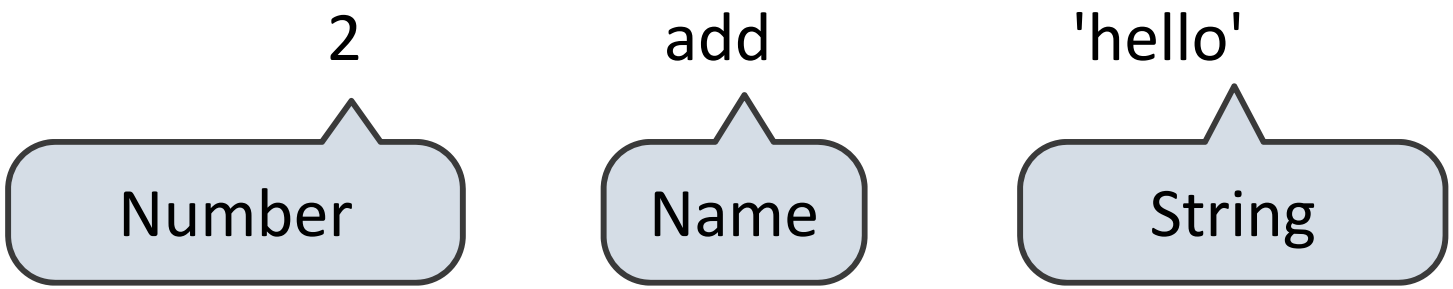
Review: Expressions



Primitive expressions:

Review: Expressions

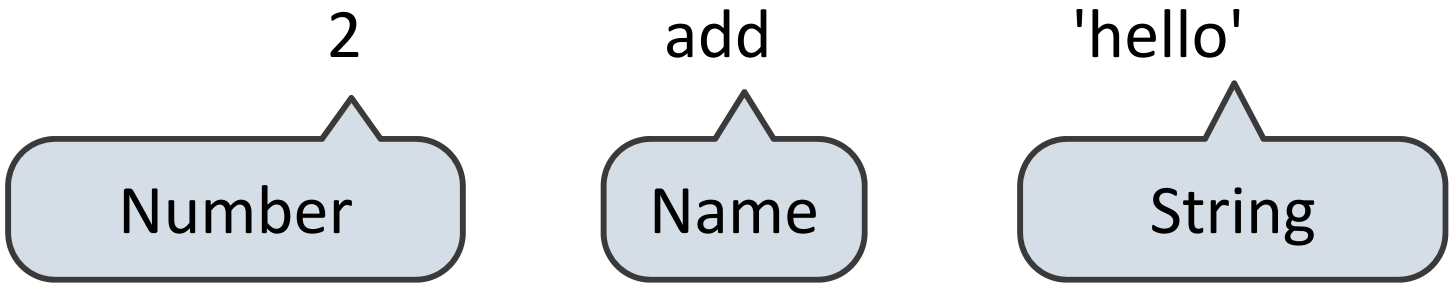
Primitive expressions:



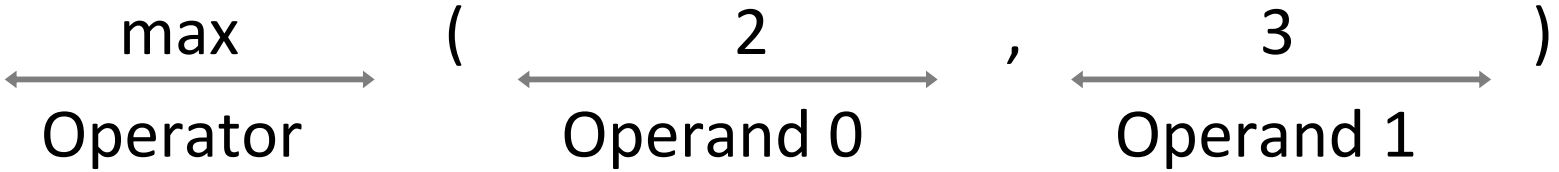
Call expressions:

Review: Expressions

Primitive expressions:

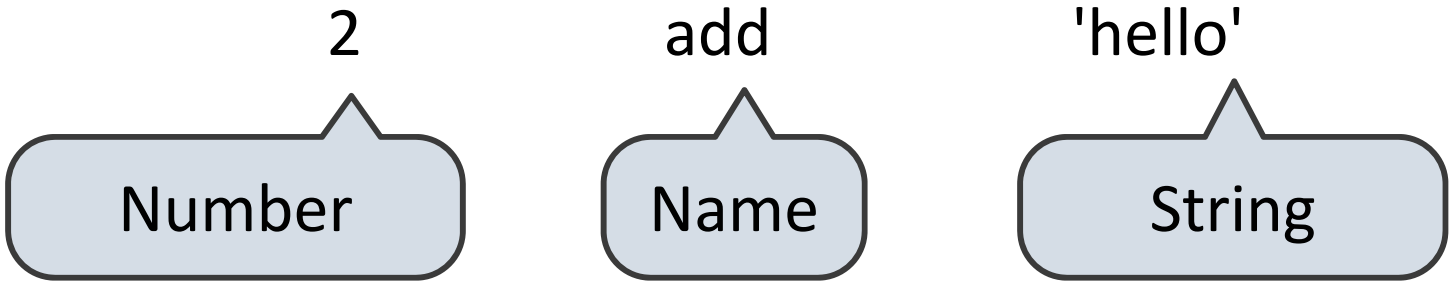


Call expressions:

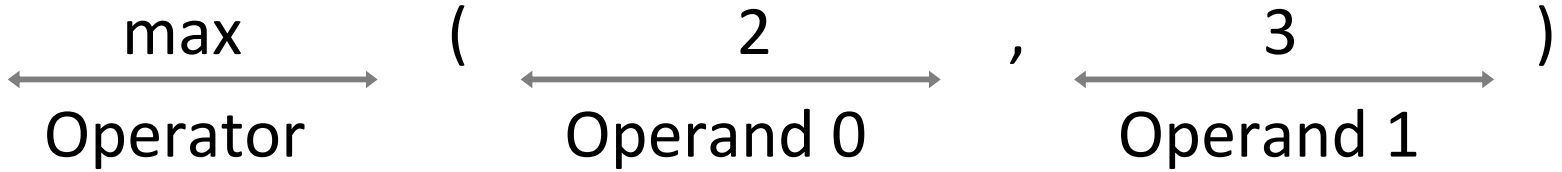


Review: Expressions

Primitive expressions:



Call expressions:

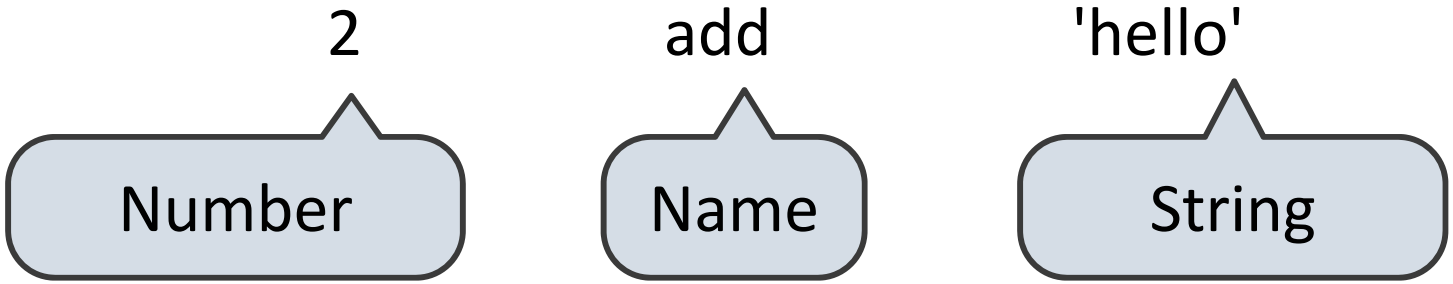


One big nested call expression

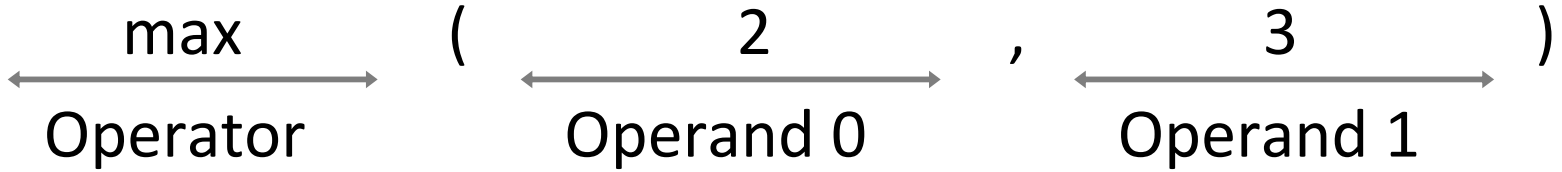
```
max(min(pow(3, 5), -4), min(1, -2))
```

Review: Expressions

Primitive expressions:



Call expressions:



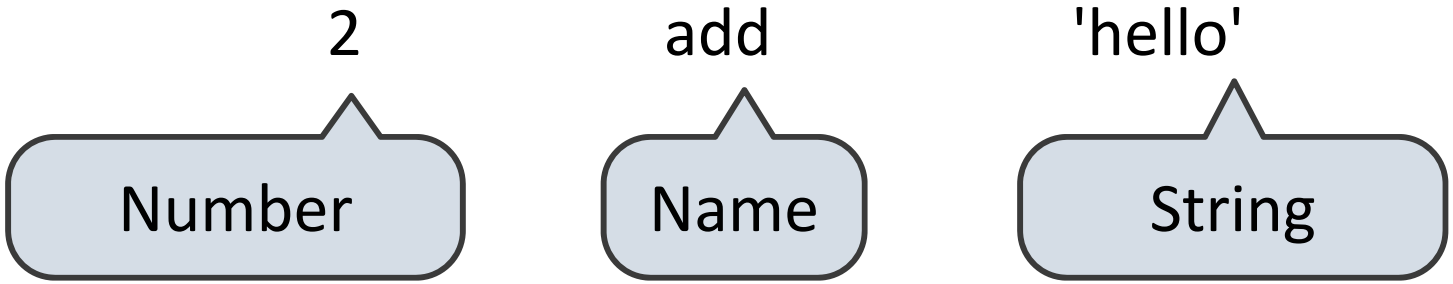
One big nested call expression

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

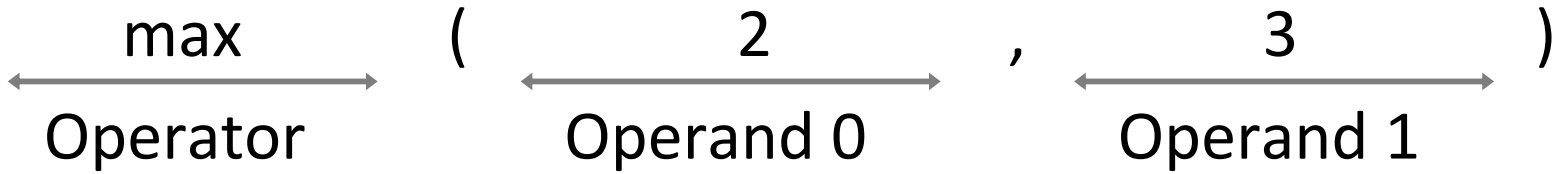
Infix operators represent implicit call expressions

Review: Expressions

Primitive expressions:



Call expressions:



One big nested call expression

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

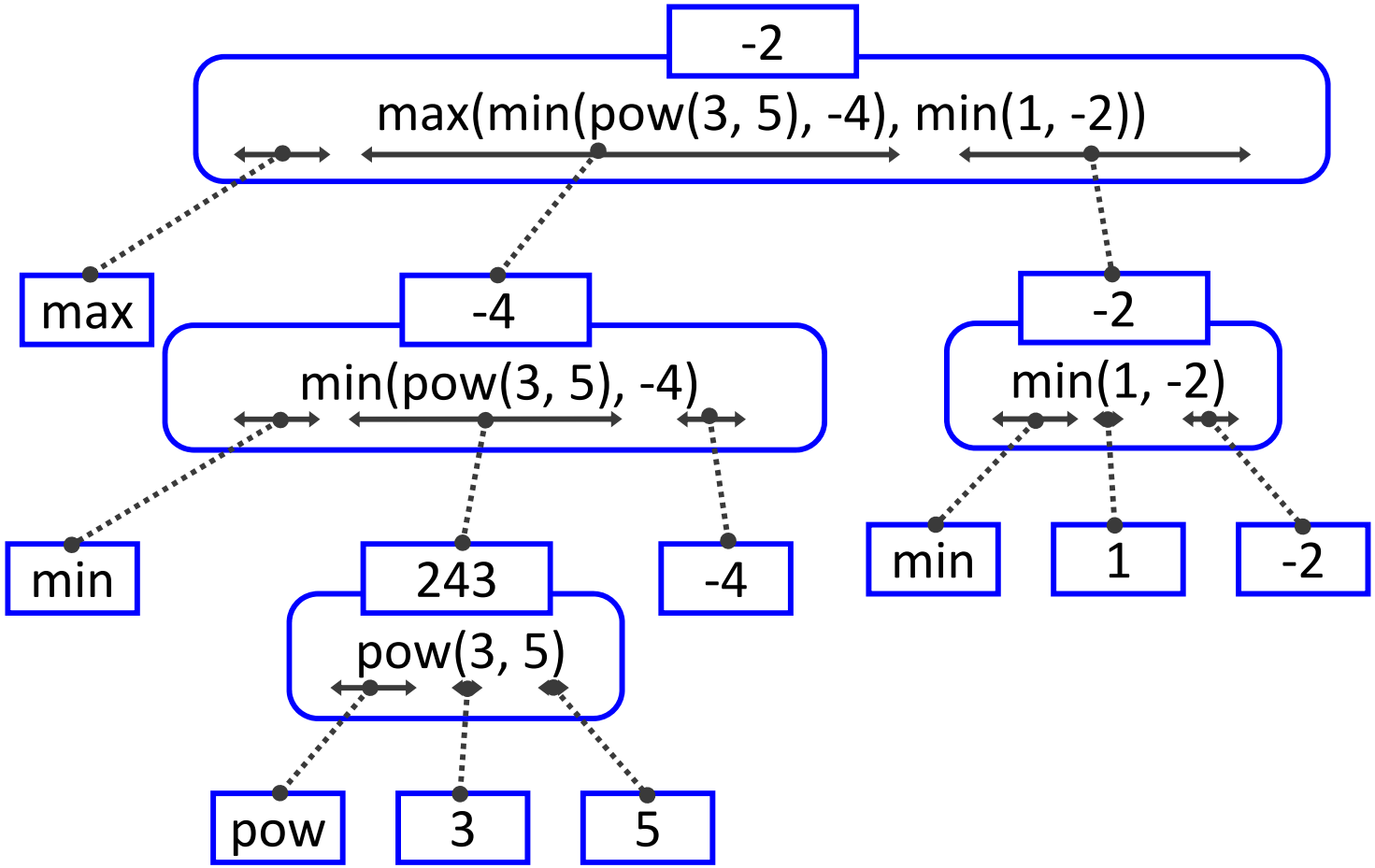
Infix operators represent implicit call expressions

Review: Evaluation Procedure

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

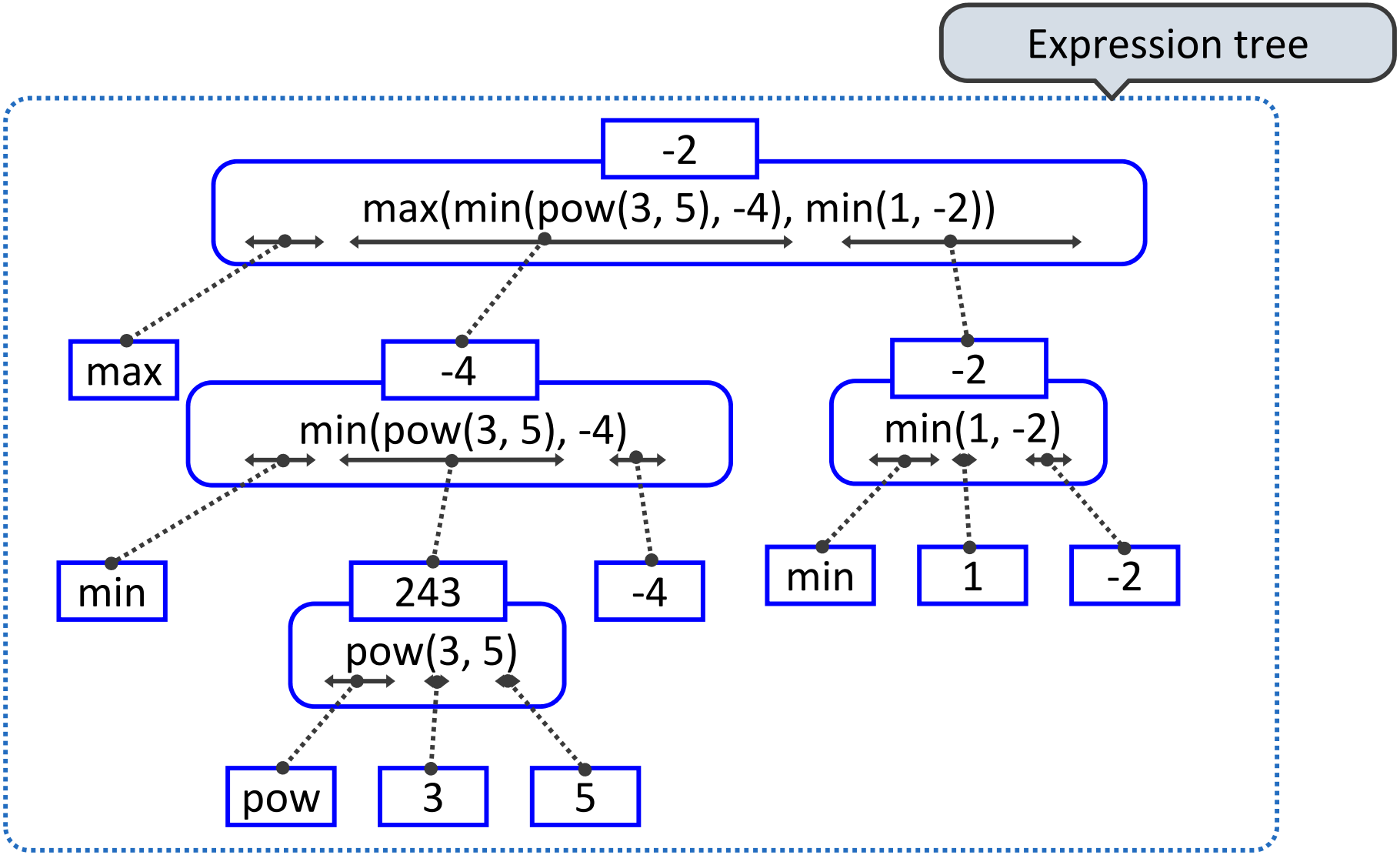
Review: Evaluation Procedure

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

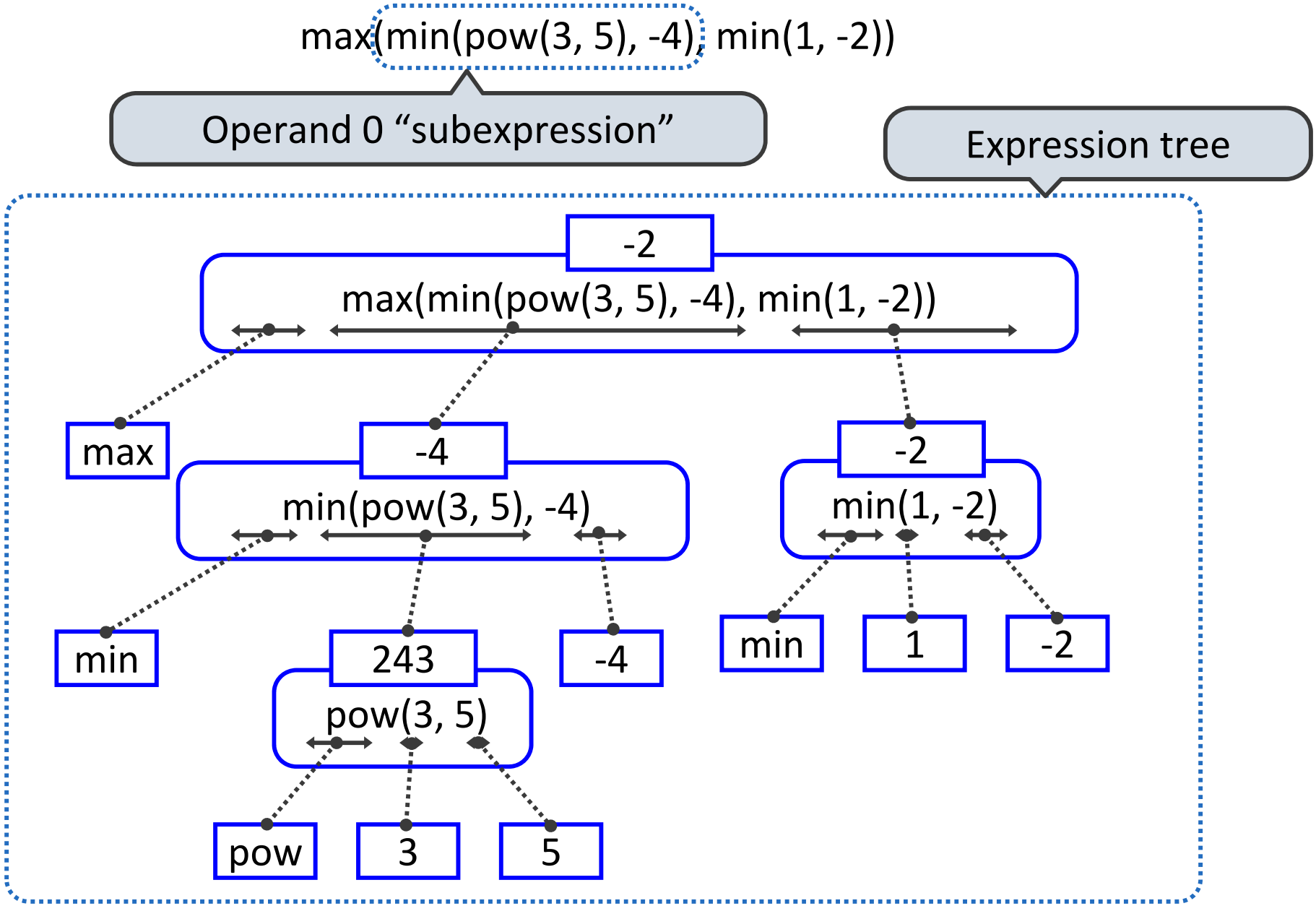


Review: Evaluation Procedure

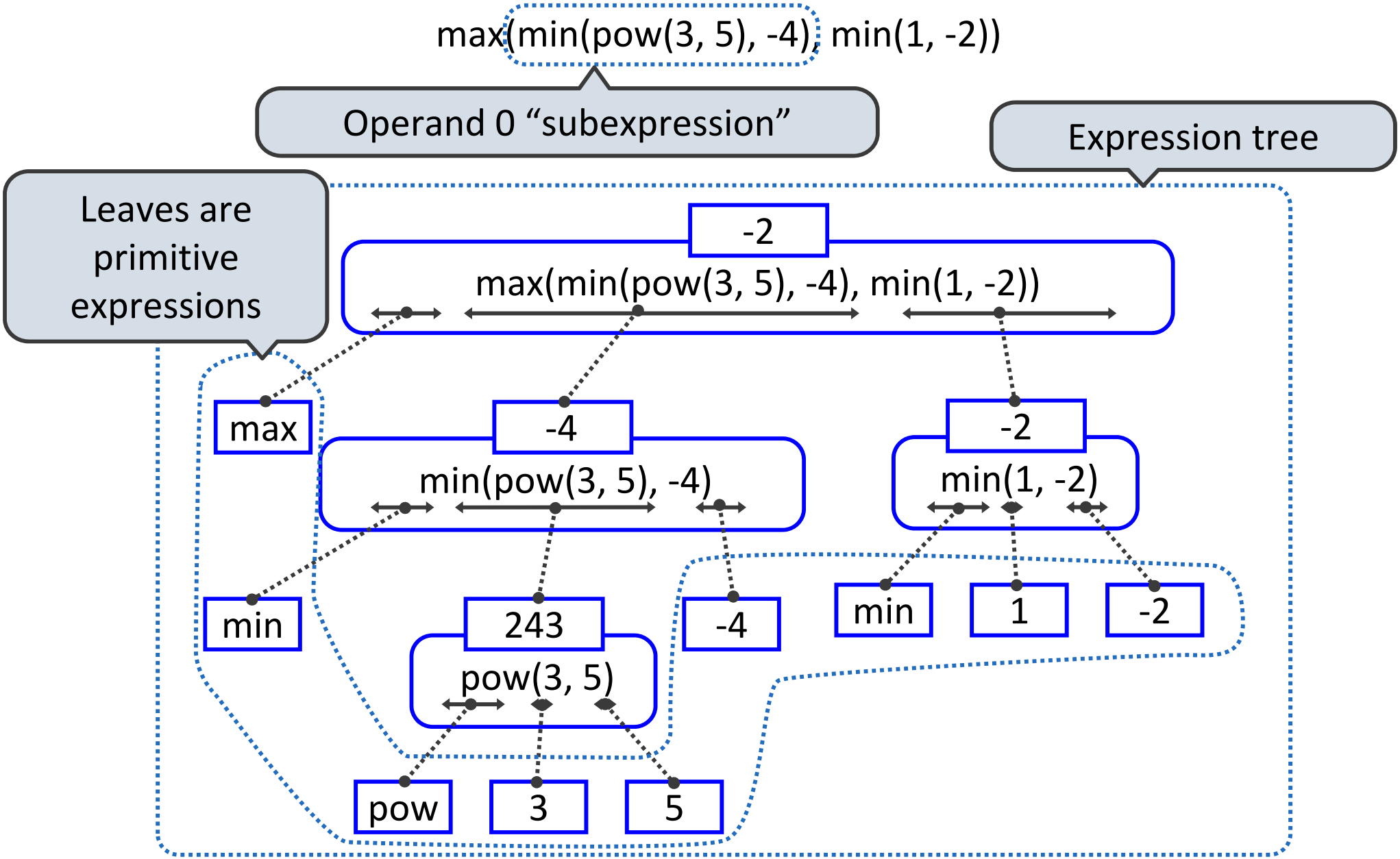
max(min(pow(3, 5), -4), min(1, -2))



Review: Evaluation Procedure



Review: Evaluation Procedure



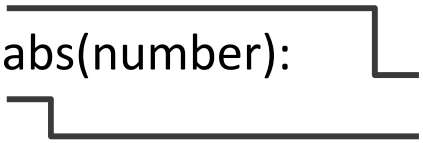
Types of Functions

Pure Functions

Non-Pure Functions

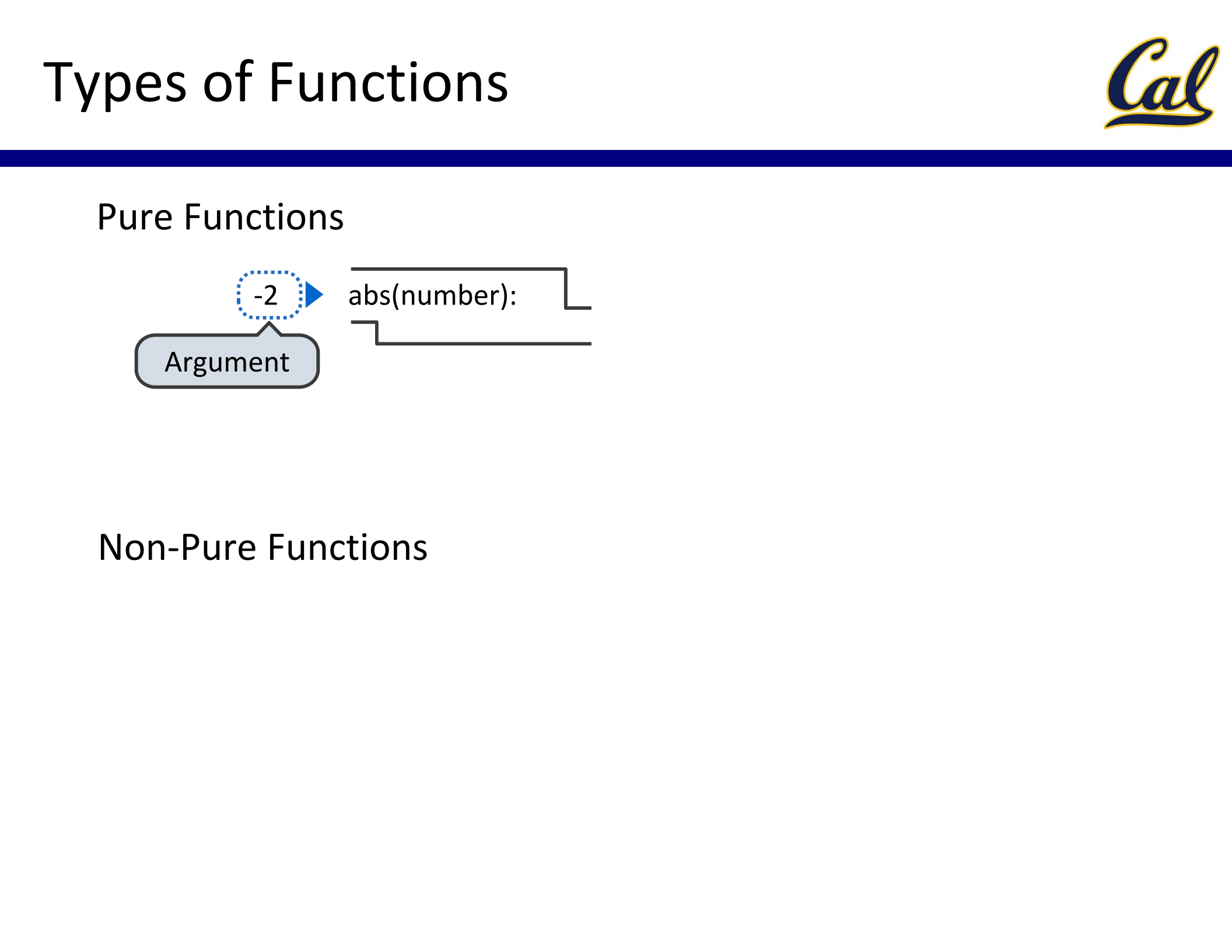
Types of Functions

Pure Functions



Non-Pure Functions

Types of Functions

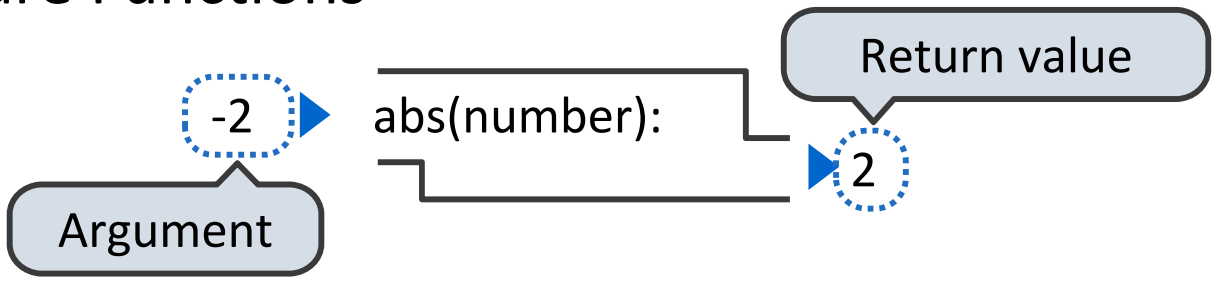


Pure Functions

Non-Pure Functions

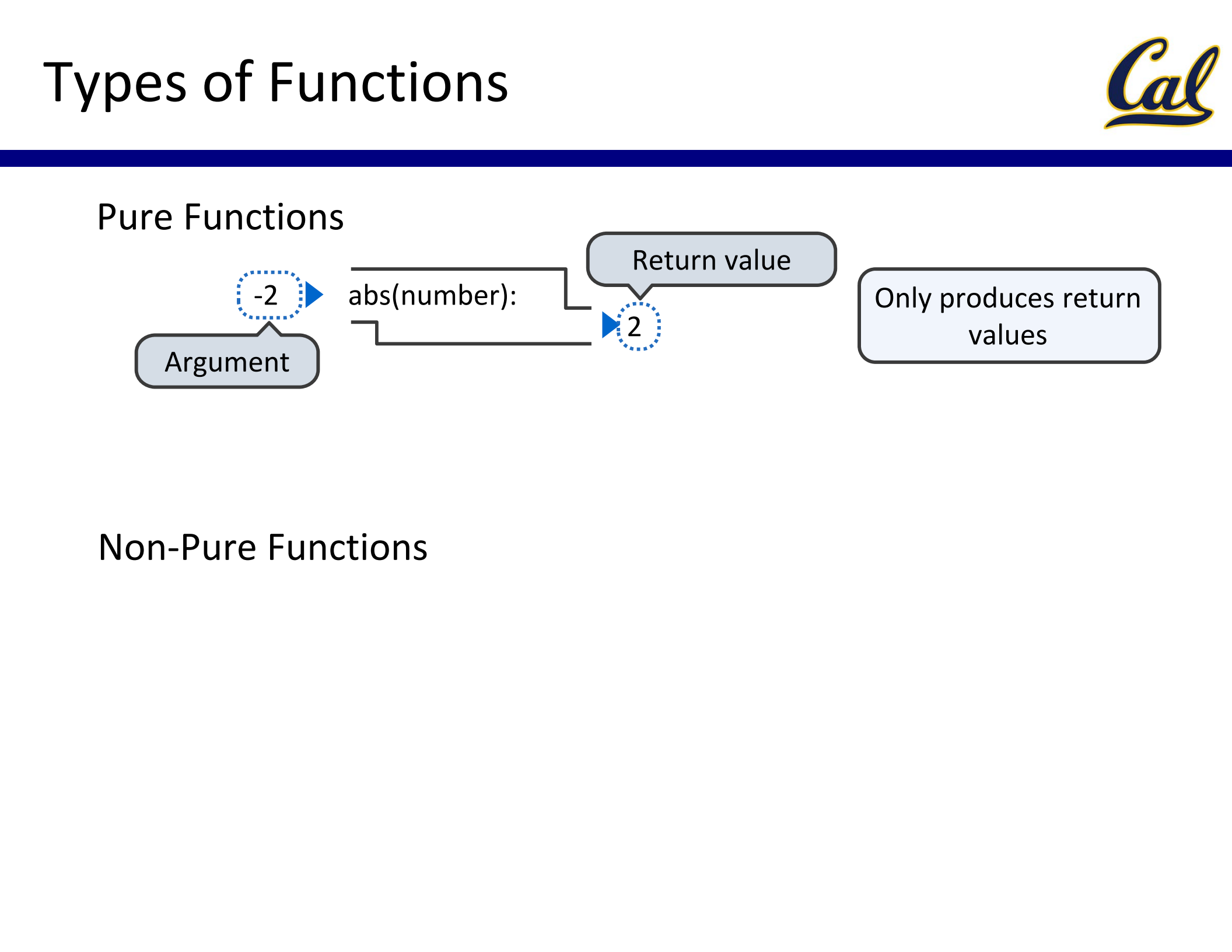
Types of Functions

Pure Functions



Non-Pure Functions

Types of Functions

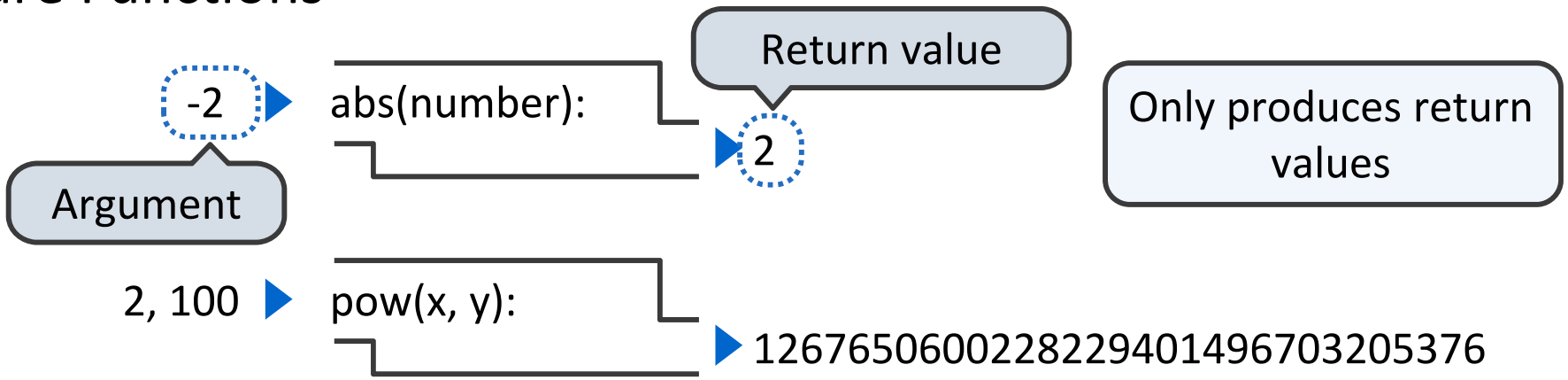


Pure Functions

Non-Pure Functions

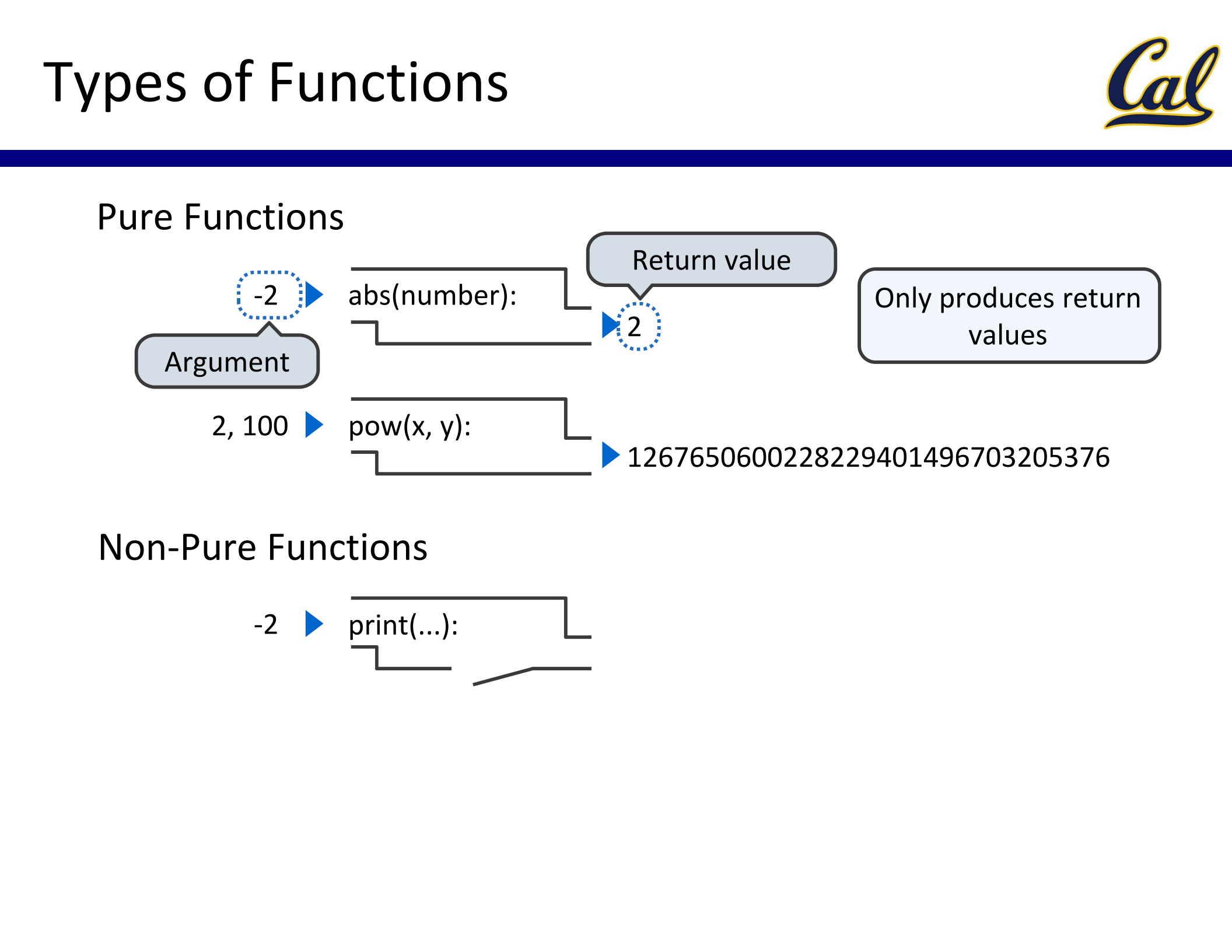
Types of Functions

Pure Functions



Non-Pure Functions

Types of Functions



Pure Functions

-2 ▶ abs(number): ▶ 2

Argument

Return value

Only produces return values

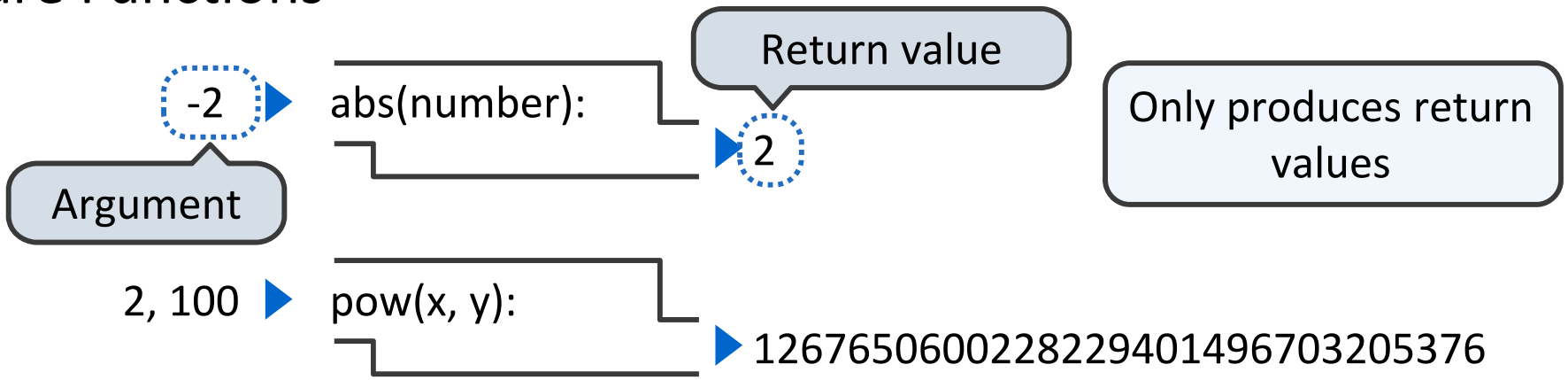
2, 100 ▶ pow(x, y): ▶ 1267650600228229401496703205376

Non-Pure Functions

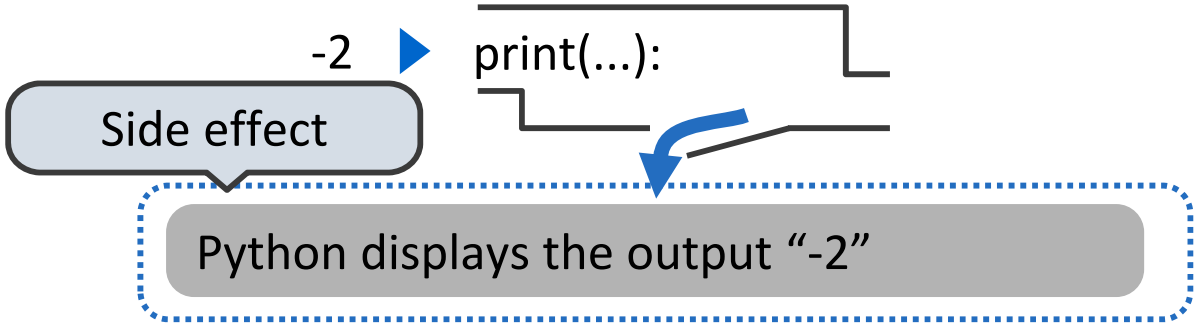
-2 ▶ print(...):

Types of Functions

Pure Functions

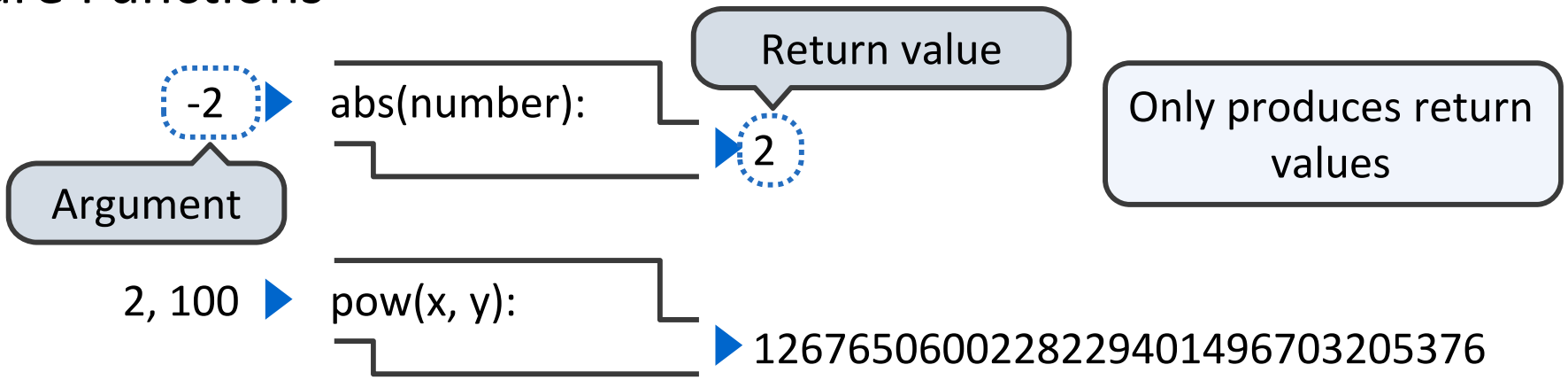


Non-Pure Functions

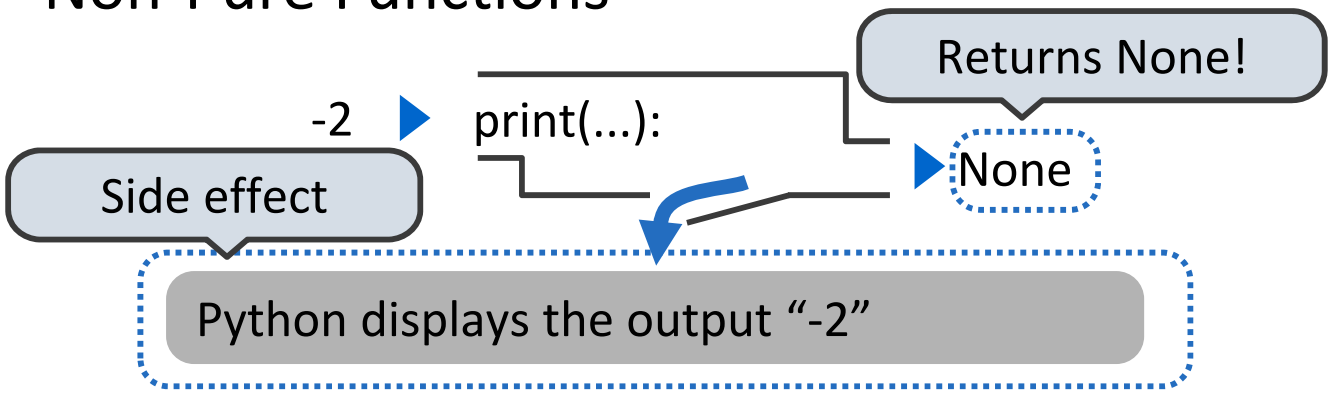


Types of Functions

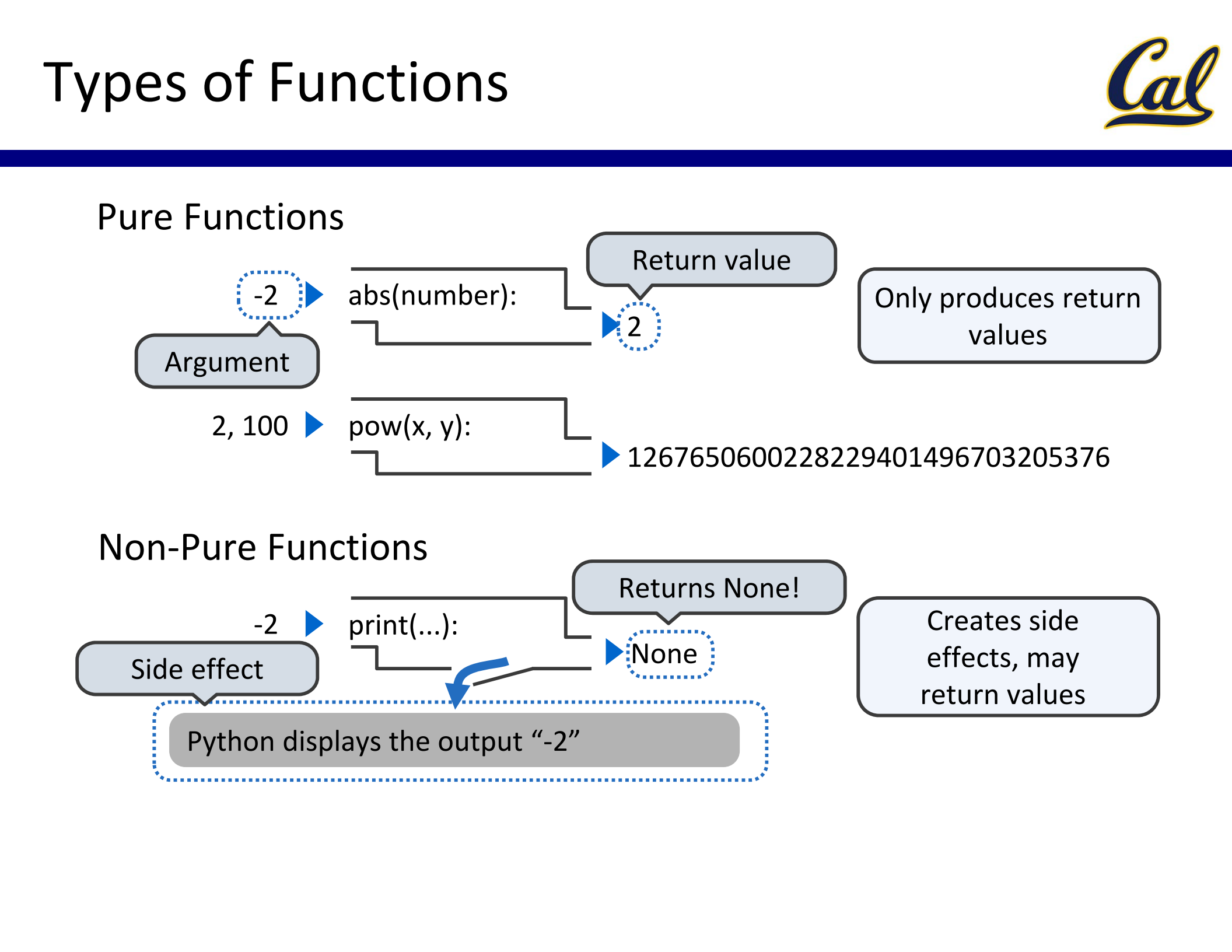
Pure Functions



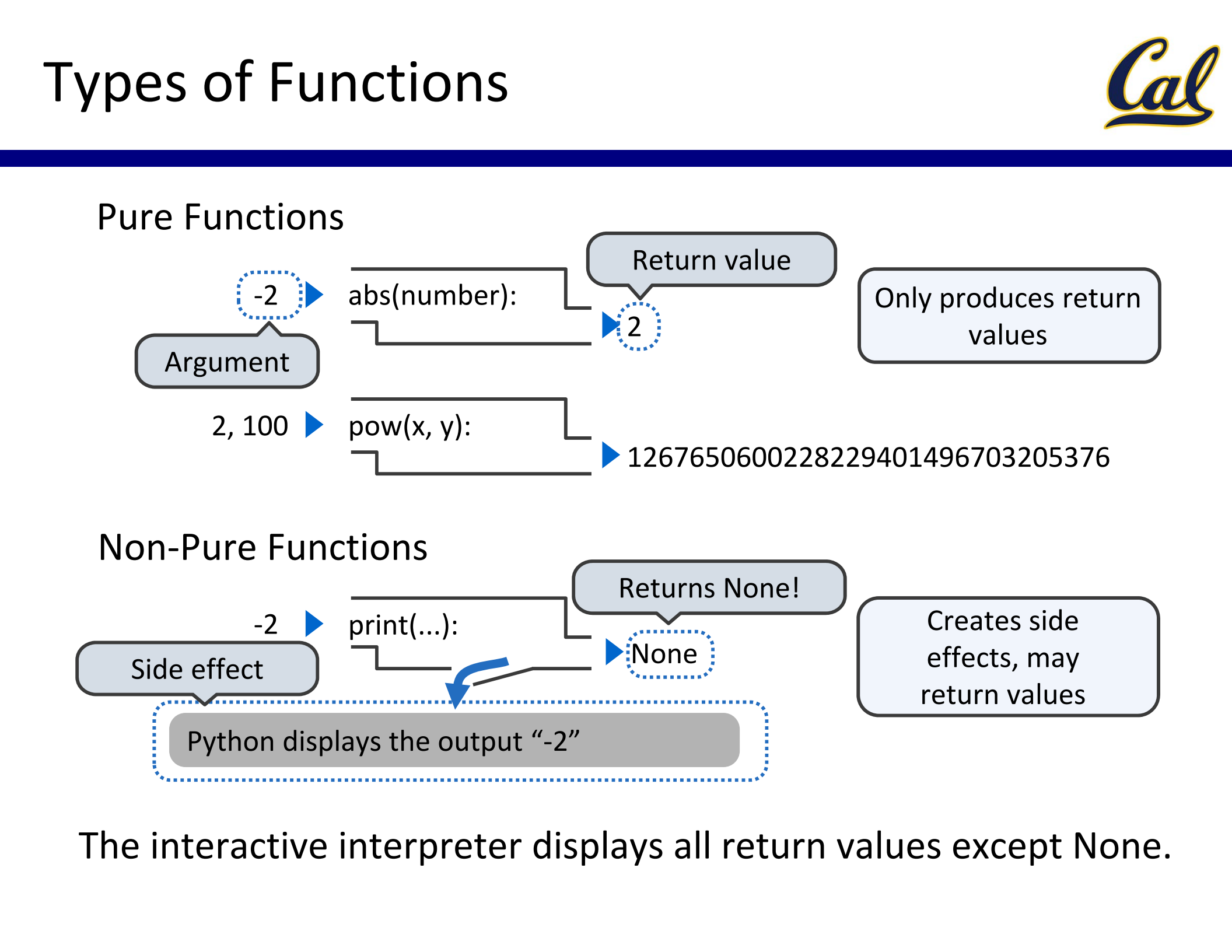
Non-Pure Functions



Types of Functions

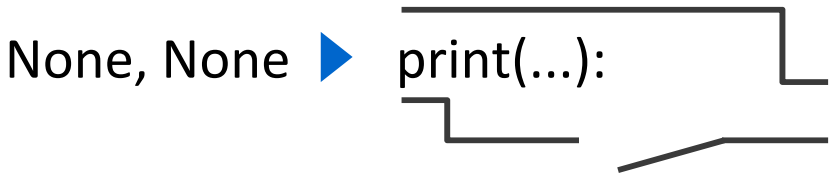


Types of Functions



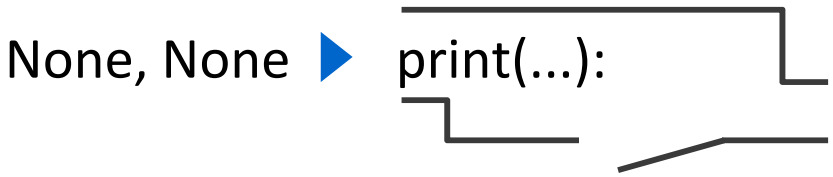
The interactive interpreter displays all return values except None.

Nested Print Expressions

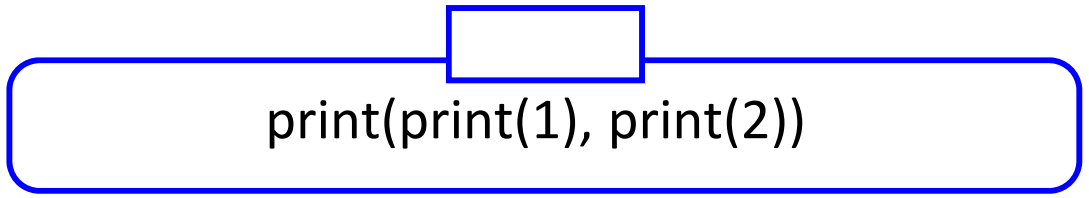



```
>>> print(print(1), print(2))
```

Nested Print Expressions



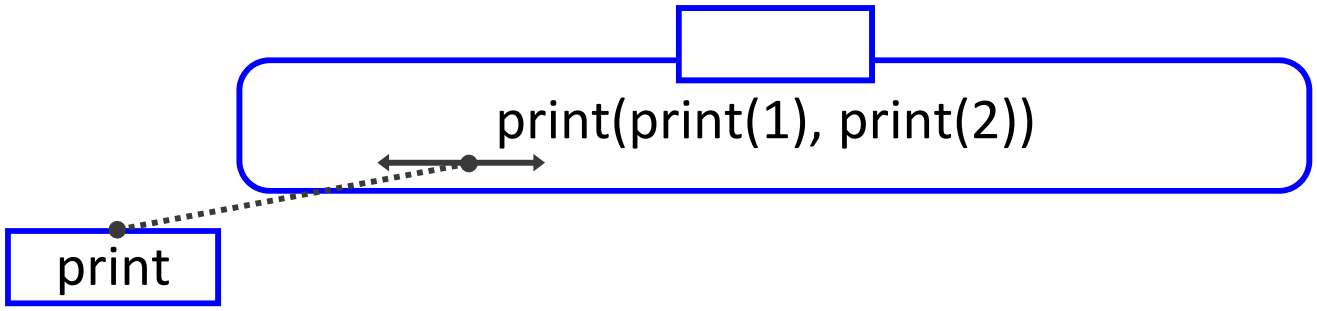

```
>>> print(print(1), print(2))
```



Nested Print Expressions

None, None ▶ print(...):

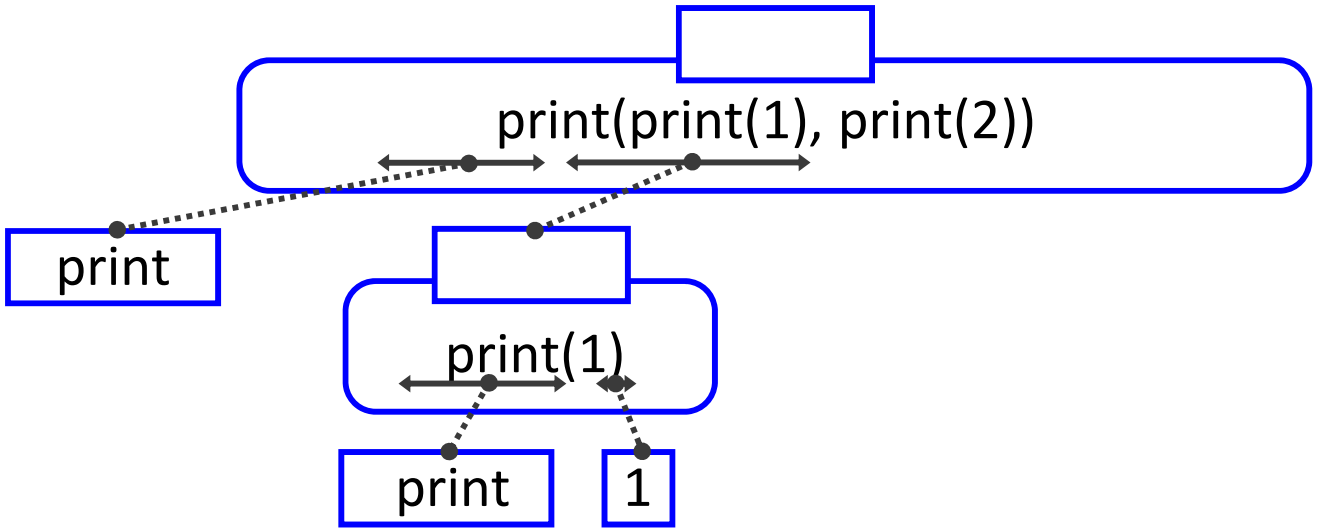
```
>>> print(print(1), print(2))
```



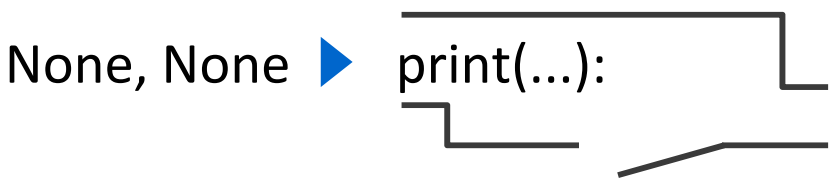
Nested Print Expressions

None, None ▶ print(...):

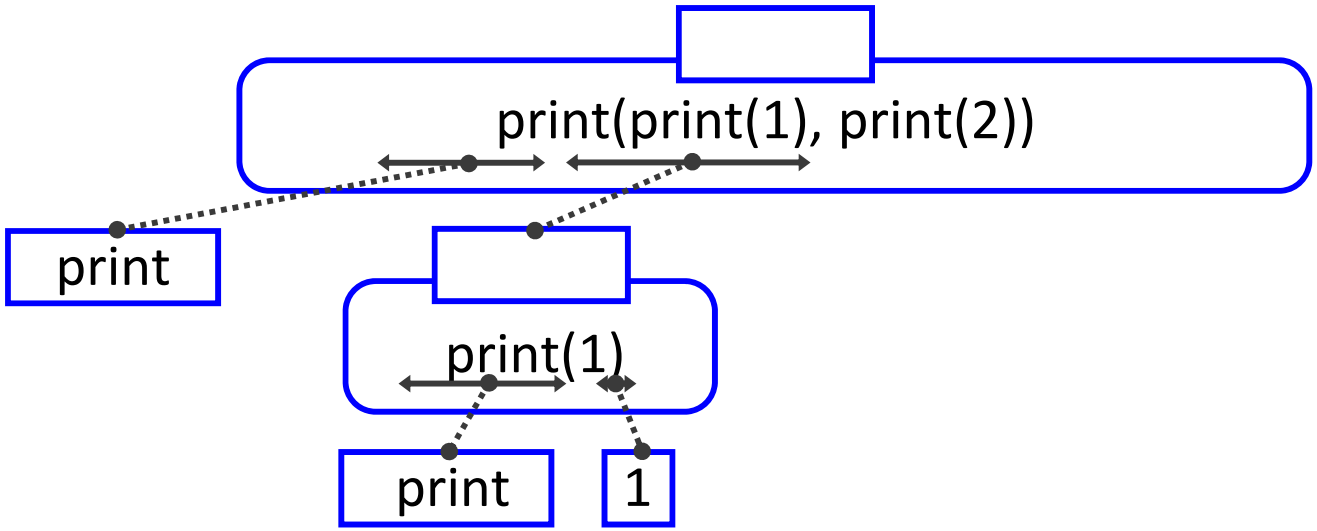
```
>>> print(print(1), print(2))
```

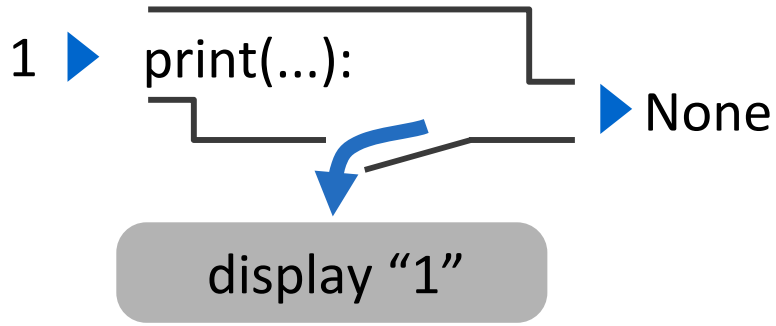


Nested Print Expressions

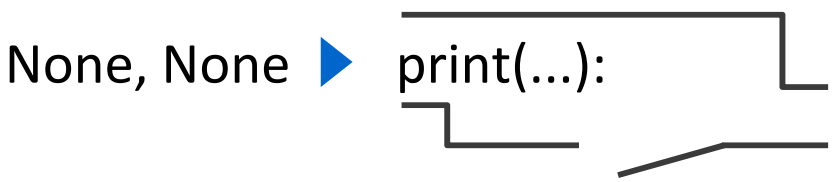



```
>>> print(print(1), print(2))
1
```

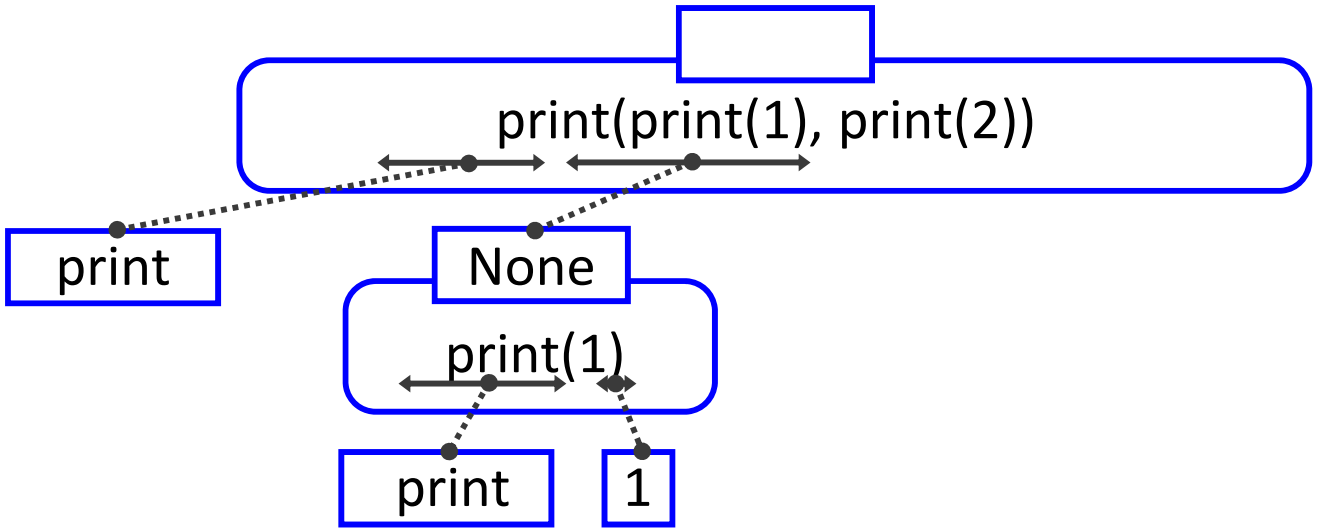


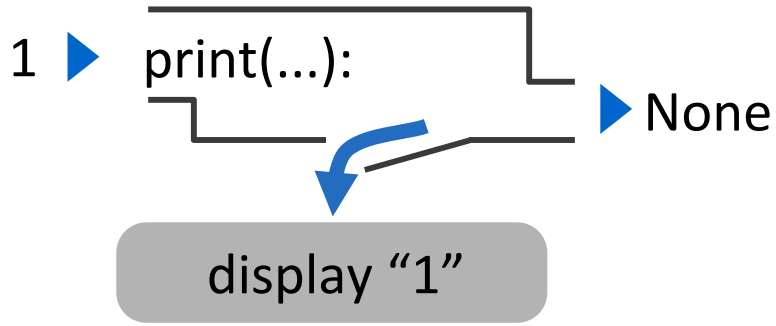


Nested Print Expressions

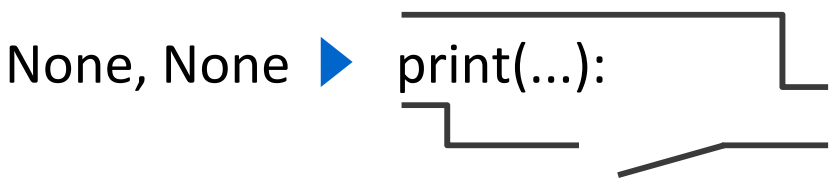



```
>>> print(print(1), print(2))
1
```

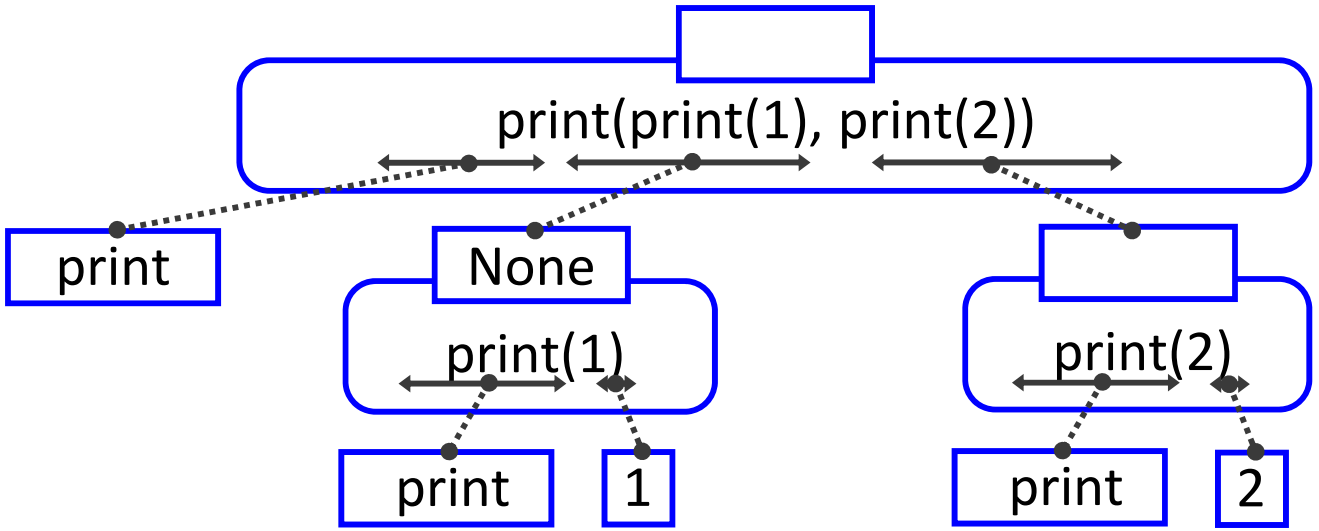


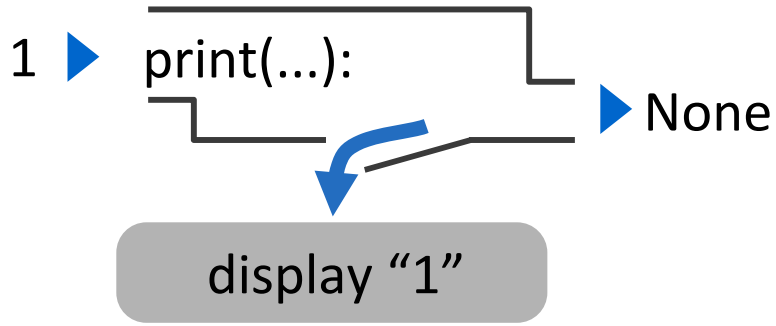


Nested Print Expressions

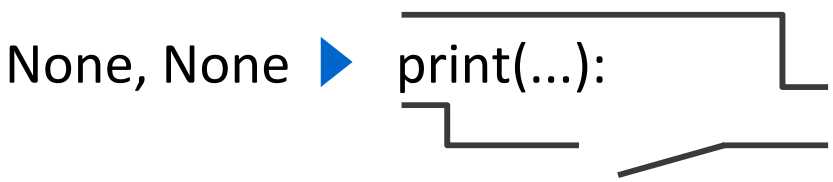



```
>>> print(print(1), print(2))
1
```

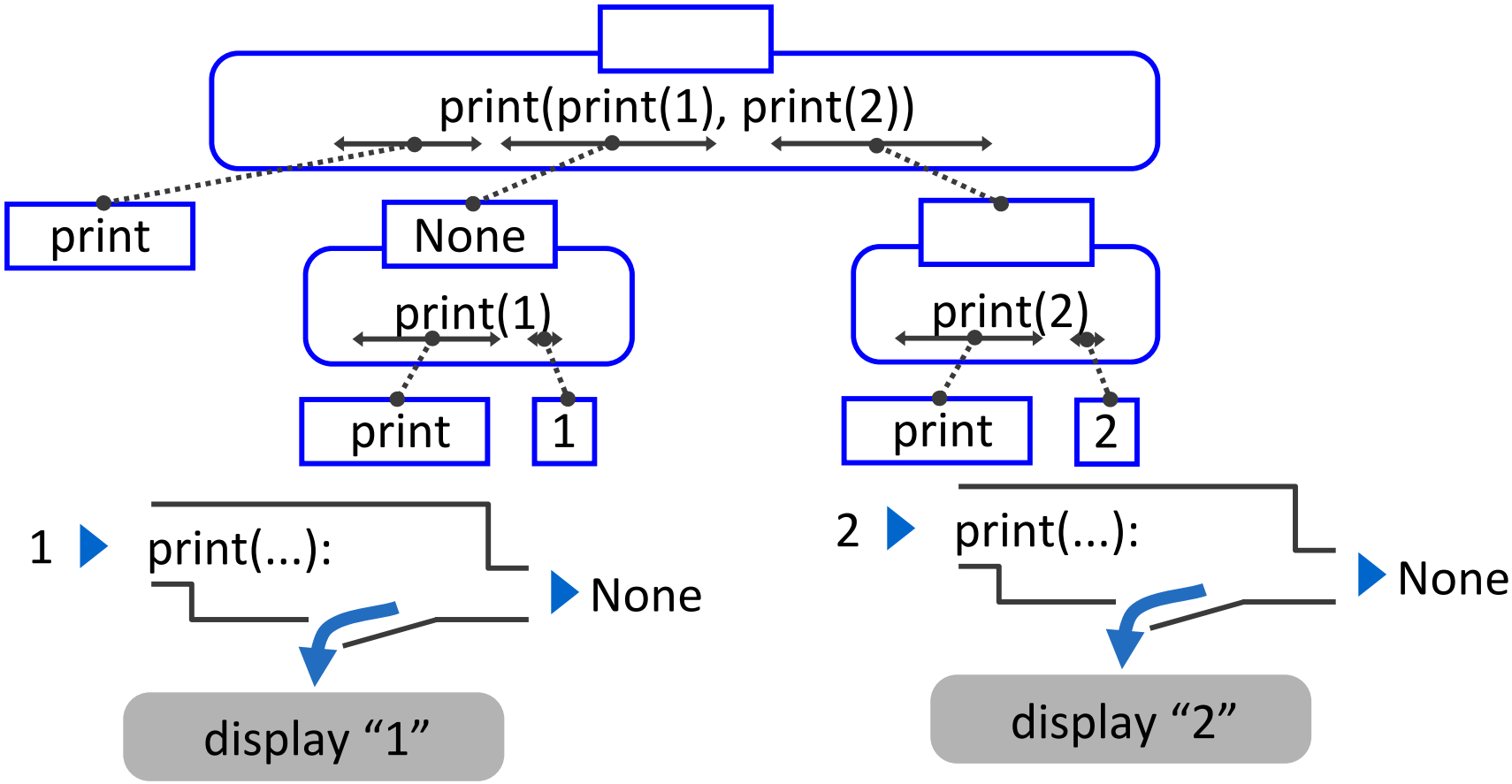




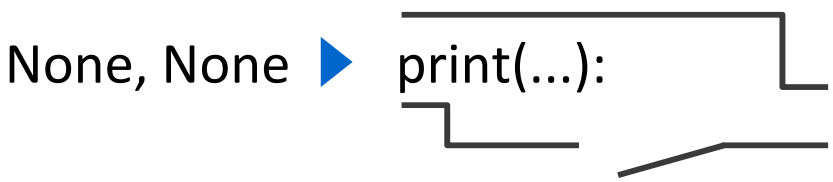
Nested Print Expressions



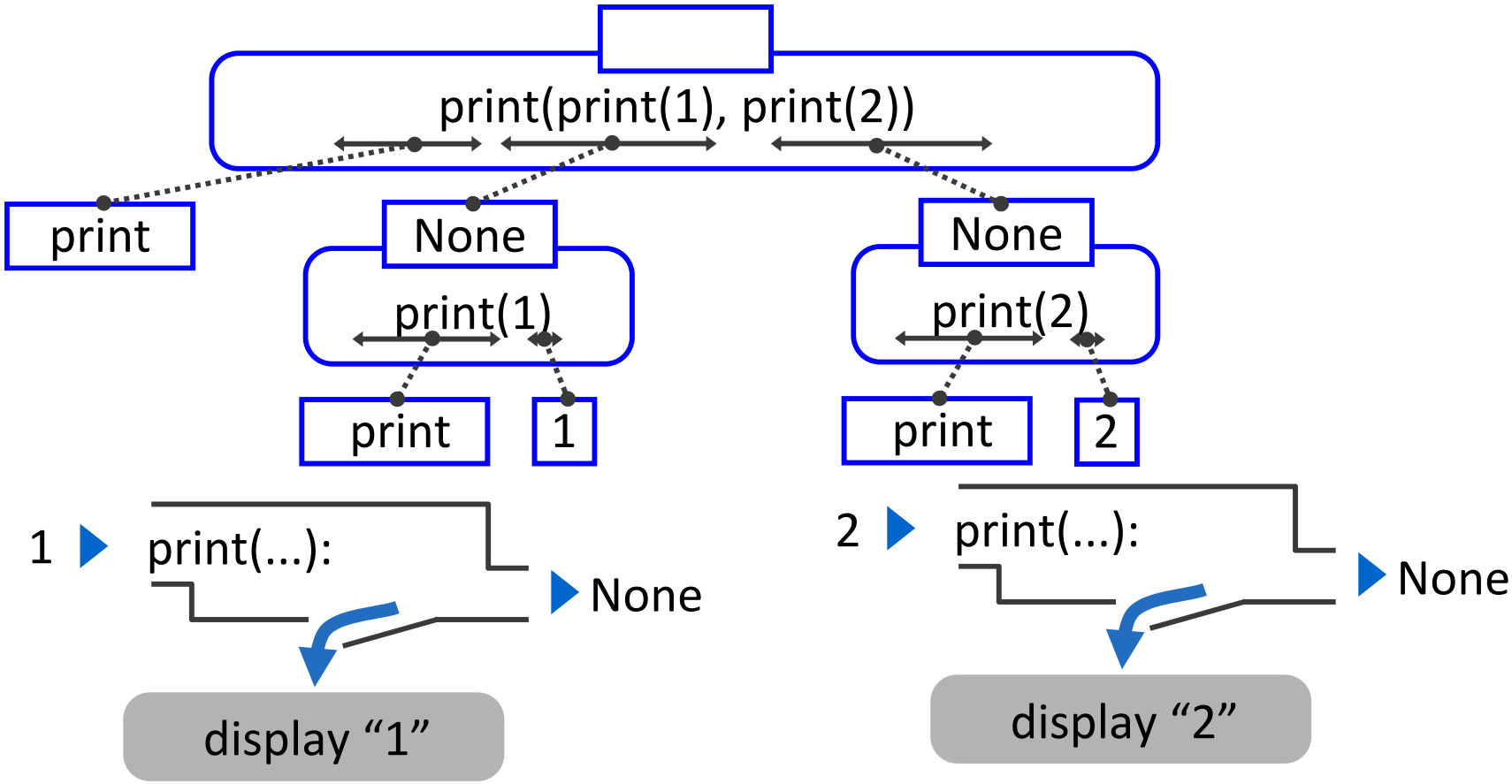

```
>>> print(print(1), print(2))
1
2
```



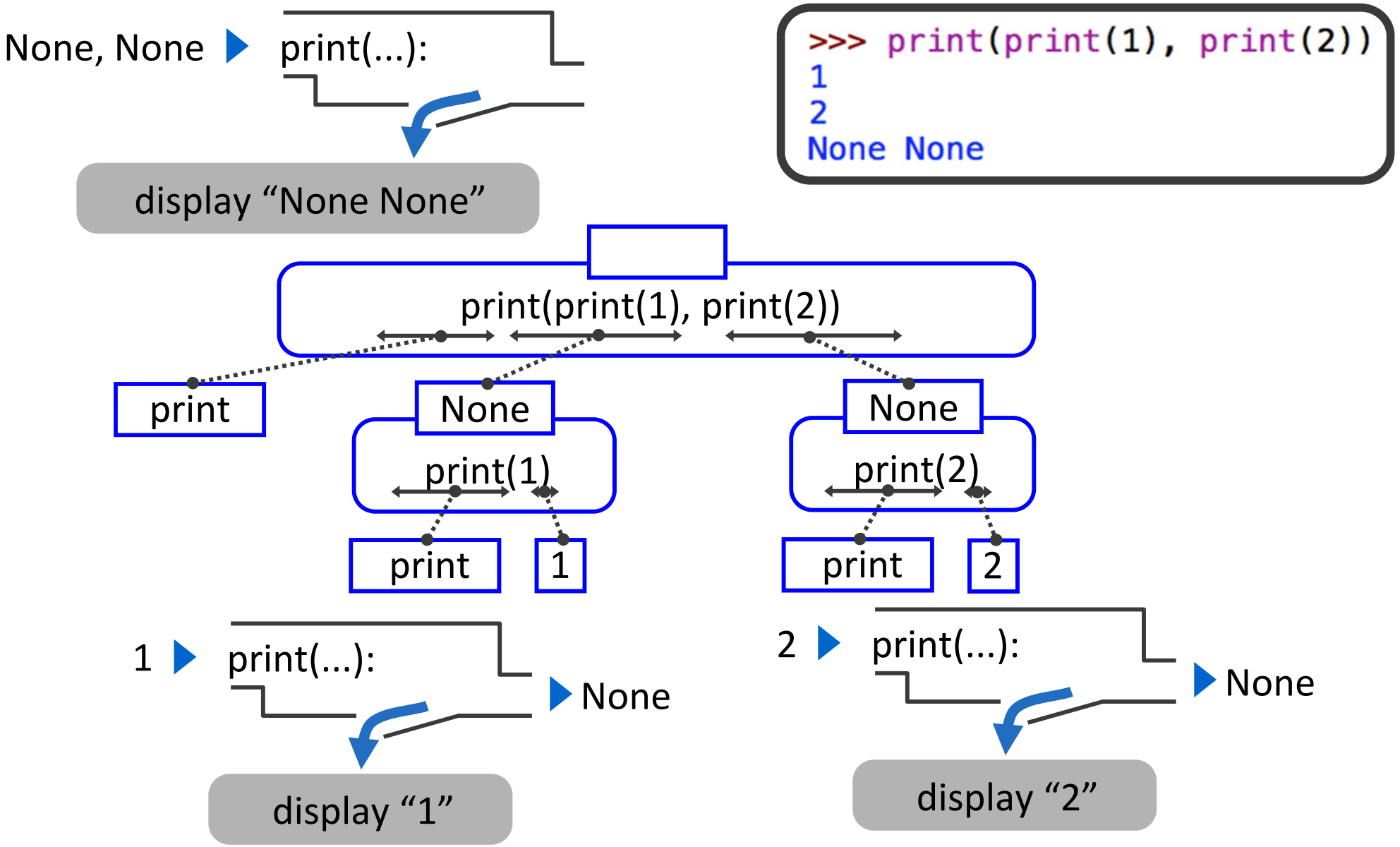
Nested Print Expressions



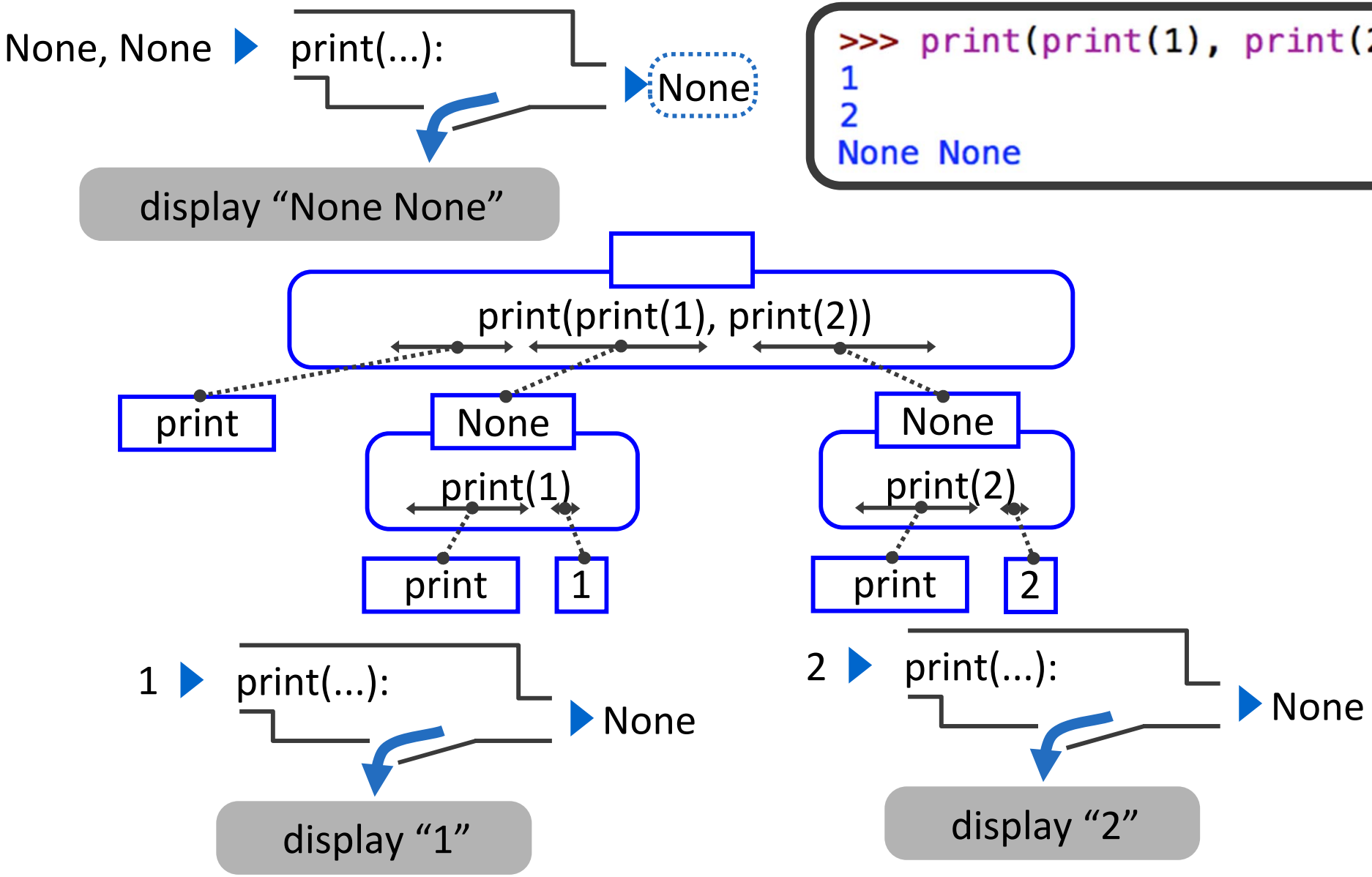

```
>>> print(print(1), print(2))
1
2
```



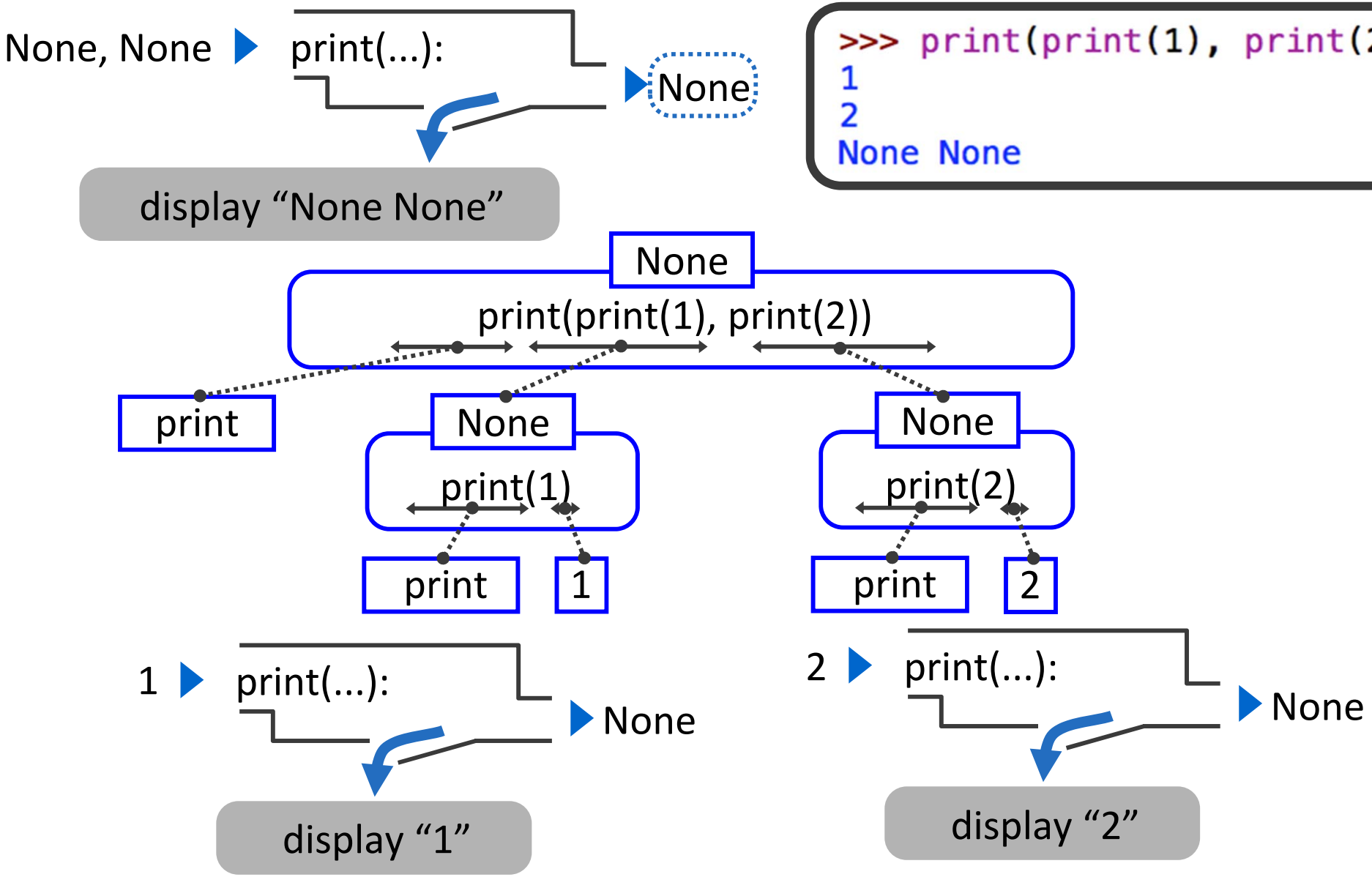
Nested Print Expressions



Nested Print Expressions



Nested Print Expressions



The Elements of Programming

The Elements of Programming

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

The Elements of Programming

- Primitive Expressions and Statements
  - The simplest building blocks of a language
- Means of Combination
  - Compound elements built from simpler ones

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