61A Lecture 1

Friday, August 30, 2013

Welcome to Berkeley Computer Science!

The Course Staff

John DeNero

The Course Staff

John DeNero

Google

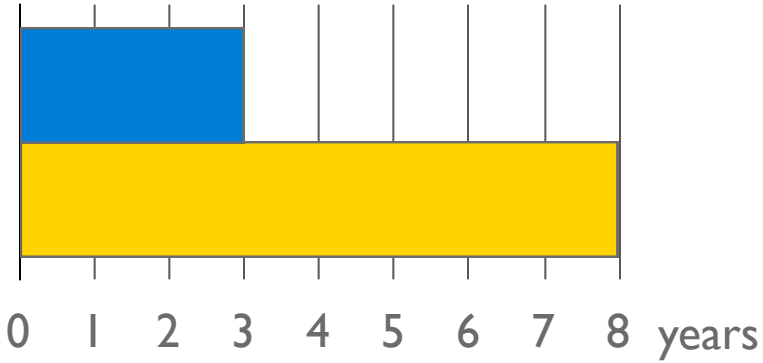
The Course Staff

John DeNero

The Course Staff

John DeNero

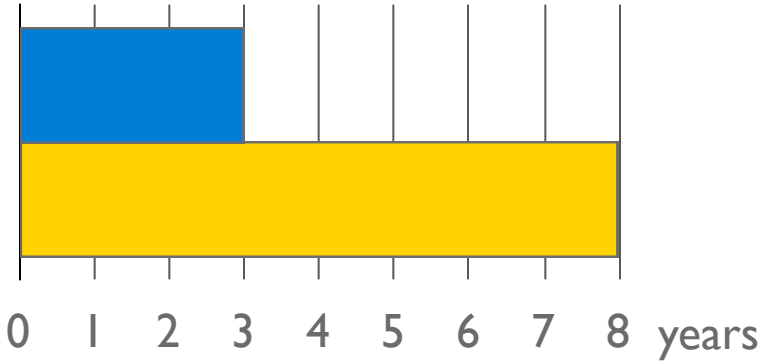
Google Cal



The Course Staff

John DeNero

Google Cal

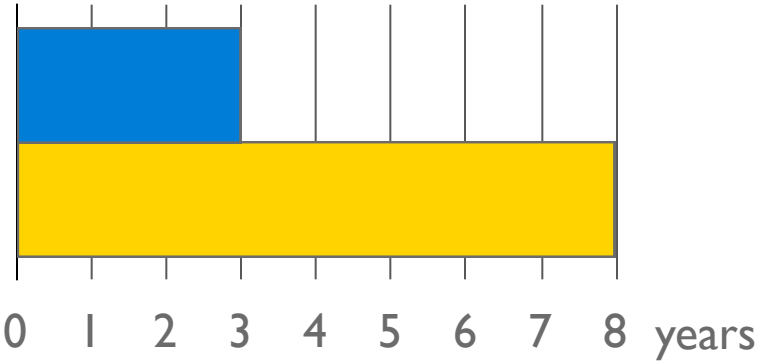


TAs hold discussion sections, labs, and *office hours*

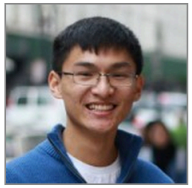
The Course Staff

John DeNero

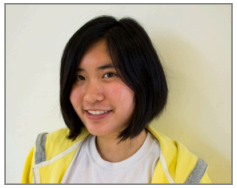
Google Cal



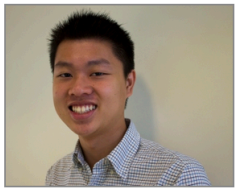
TAs hold discussion sections, labs, and *office hours*

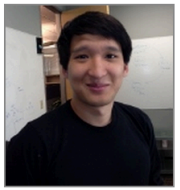



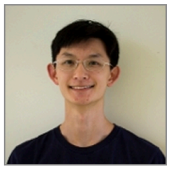
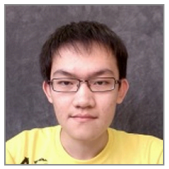






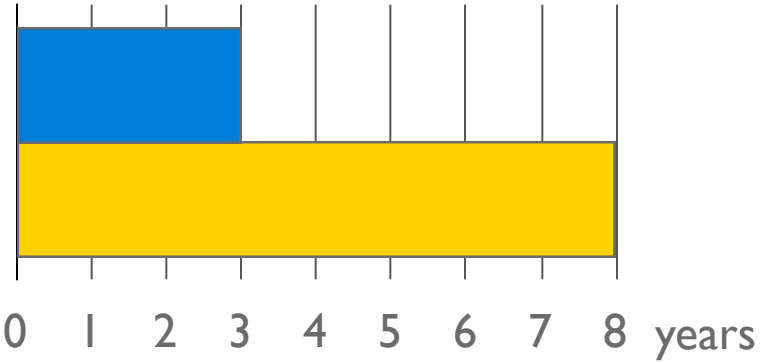



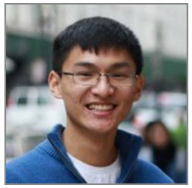

The Course Staff

John DeNero

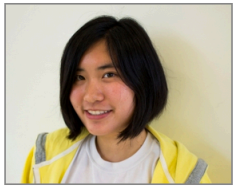
Google Cal



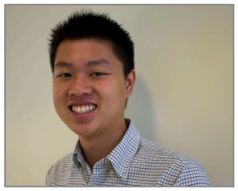
TAs hold discussion sections, labs, and *office hours*

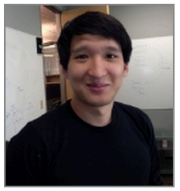



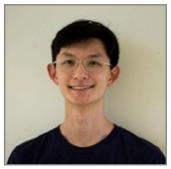
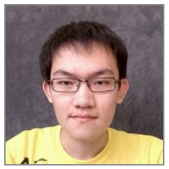






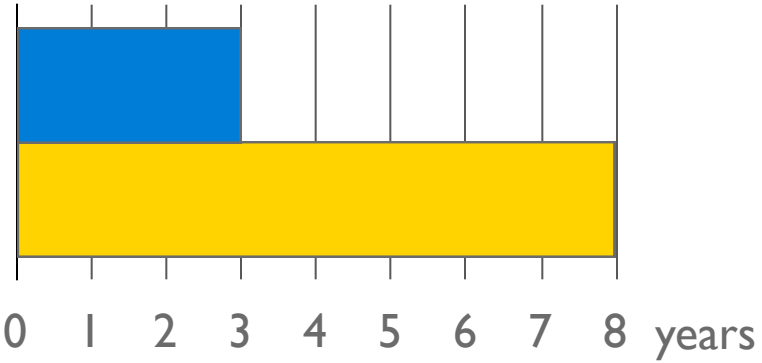




Readers are your personal programming mentors

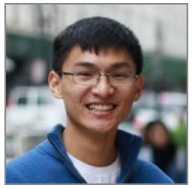
The Course Staff

John DeNero

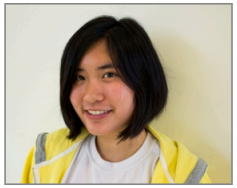
Google Cal



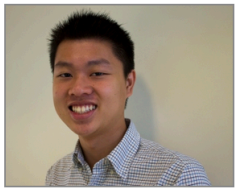
TAs hold discussion sections, labs, and *office hours*

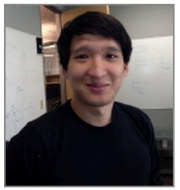



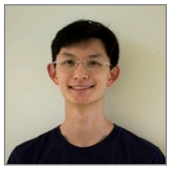
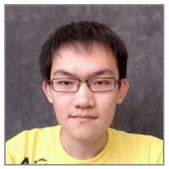







Readers are your personal programming mentors

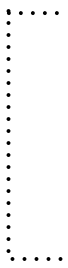
Lab Assistants ensure that you don't get stuck for too long

An Introduction to Computer Science

What is Computer Science?

What is Computer Science?

The study of



What is Computer Science?

The study of What problems can be solved using computation,

What is Computer Science?

The study of

- What problems can be solved using computation,
- How to solve those problems, and

What is Computer Science?

The study of

- What problems can be solved using computation,
- How to solve those problems, and
- What design choices lead to effective solutions.

What is Computer Science?

The study of

- What problems can be solved using computation,
- How to solve those problems, and
- What design choices lead to effective solutions.

Systems

What is Computer Science?

The study of

- What problems can be solved using computation,
- How to solve those problems, and
- What design choices lead to effective solutions.

Systems

Artificial Intelligence

What is Computer Science?

The study of

- What problems can be solved using computation,
- How to solve those problems, and
- What design choices lead to effective solutions.

Systems

Artificial Intelligence

Graphics

What is Computer Science?

The study of

- What problems can be solved using computation,
- How to solve those problems, and
- What design choices lead to effective solutions.

Systems

Artificial Intelligence

Graphics

Security

What is Computer Science?

The study of

- What problems can be solved using computation,
- How to solve those problems, and
- What design choices lead to effective solutions.

Systems

Artificial Intelligence

Graphics

Security

Networking

Programming Languages

Theory

Scientific Computing

...

What is Computer Science?

The study of

- What problems can be solved using computation,
- How to solve those problems, and
- What design choices lead to effective solutions.

Systems

Artificial Intelligence

Graphics

Security

Networking

Programming Languages

Theory

Scientific Computing

...

What is Computer Science?

The study of

- What problems can be solved using computation,
- How to solve those problems, and
- What design choices lead to effective solutions.

Systems

Artificial Intelligence

Graphics

Security

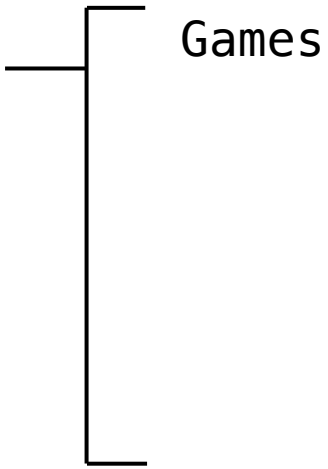
Networking

Programming Languages

Theory

Scientific Computing

...



What is Computer Science?

The study of

- What problems can be solved using computation,
- How to solve those problems, and
- What design choices lead to effective solutions.

- Systems
- Artificial Intelligence
  - Games
  - Robotics
- Graphics
- Security
- Networking
- Programming Languages
- Theory
- Scientific Computing
- ...

What is Computer Science?

The study of

- What problems can be solved using computation,
- How to solve those problems, and
- What design choices lead to effective solutions.

- Systems
- Artificial Intelligence
  - Games
  - Robotics
  - Natural Language Processing
- Graphics
- Security
- Networking
- Programming Languages
- Theory
- Scientific Computing
- ...

What is Computer Science?

The study of

- What problems can be solved using computation,
- How to solve those problems, and
- What design choices lead to effective solutions.

- Systems
- Artificial Intelligence
  - Games
  - Robotics
  - Natural Language Processing
  - ...
- Graphics
- Security
- Networking
- Programming Languages
- Theory
- Scientific Computing
- ...

What is This Course About?

What is This Course About?

- A course about managing complexity

What is This Course About?

- A course about managing complexity
  - Mastering abstraction

What is This Course About?

- A course about managing complexity
  - Mastering abstraction
  - Programming paradigms

What is This Course About?

- A course about managing complexity
  - Mastering abstraction
  - Programming paradigms
  - Not about 1’s and 0’s

What is This Course About?

- A course about managing complexity
  - Mastering abstraction
  - Programming paradigms
  - Not about 1's and 0's

- An introduction to Python

What is This Course About?

- A course about managing complexity
  - Mastering abstraction
  - Programming paradigms
  - Not about 1's and 0's

- An introduction to Python
  - All the features we really need: introduced today

What is This Course About?

- A course about managing complexity
  - Mastering abstraction
  - Programming paradigms
  - Not about 1's and 0's

- An introduction to Python
  - All the features we really need: introduced today
  - Understanding through implementation

What is This Course About?

- A course about managing complexity
  - Mastering abstraction
  - Programming paradigms
  - Not about 1's and 0's

- An introduction to Python
  - All the features we really need: introduced today
  - Understanding through implementation
  - How computers interpret programming languages

What is This Course About?

- A course about managing complexity
  - Mastering abstraction
  - Programming paradigms
  - Not about 1's and 0's

- An introduction to Python
  - All the features we really need: introduced today
  - Understanding through implementation
  - How computers interpret programming languages
- A challenging course that will demand **a lot** of you

What is This Course About?

Plone Conference. Photo courtesy of Kriszta Szita

Course Logistics and Policies

Alternatives to This Course

Alternatives to This Course

CS 61AS: Self-paced 61A

Alternatives to This Course

CS 61AS: Self-paced 61A

CS 10: The Beauty and Joy of Computing

Course Policies

Course Policies

```
The purpose of this course is to help you learn
```

Course Policies

The purpose of this course is to help you learn

The staff is here to make you successful

Course Policies

The purpose of this course is to help you learn

The staff is here to make you successful

All the details are online:

http://inst.eecs.berkeley.edu/~cs61A/fa13/about.html

Collaboration

Collaboration

- Discuss everything with each other

Collaboration

- Discuss everything with each other
- **EPA:** Effort, participation, and altruism

Collaboration

- Discuss everything with each other
- **EPA:** Effort, participation, and altruism
- Homework can be completed with a partner

Collaboration

- Discuss everything with each other
- **EPA:** Effort, participation, and altruism
- Homework can be completed with a partner
- Projects *should* be completed with a partner

Collaboration

- Discuss everything with each other
- **EPA:** Effort, participation, and altruism
- Homework can be completed with a partner
- Projects *should* be completed with a partner
- Find a project partner in your section (if you can)

Collaboration

- Discuss everything with each other
- **EPA:** Effort, participation, and altruism
- Homework can be completed with a partner
- Projects *should* be completed with a partner
- Find a project partner in your section (if you can)

The limits of collaboration

Collaboration

- Discuss everything with each other
- **EPA:** Effort, participation, and altruism
- Homework can be completed with a partner
- Projects *should* be completed with a partner
- Find a project partner in your section (if you can)

The limits of collaboration

- One simple rule: Don’t share your code, except with partners

Collaboration

- Discuss everything with each other
- **EPA:** Effort, participation, and altruism
- Homework can be completed with a partner
- Projects *should* be completed with a partner
- Find a project partner in your section (if you can)

The limits of collaboration

- One simple rule: Don't share your code, except with partners
- Copying project solutions is a serious offense!

Collaboration

- Discuss everything with each other
- **EPA:** Effort, participation, and altruism
- Homework can be completed with a partner
- Projects *should* be completed with a partner
- Find a project partner in your section (if you can)

The limits of collaboration

- One simple rule: Don’t share your code, except with partners
- Copying project solutions is a serious offense!
- We really do catch people who violate the rules

Collaboration

- Discuss everything with each other
- **EPA:** Effort, participation, and altruism
- Homework can be completed with a partner
- Projects *should* be completed with a partner
- Find a project partner in your section (if you can)

The limits of collaboration

- One simple rule: Don't share your code, except with partners
- Copying project solutions is a serious offense!
- We really do catch people who violate the rules
  - We also know how to search the web for solutions

Collaboration

- Discuss everything with each other
- **EPA:** Effort, participation, and altruism
- Homework can be completed with a partner
- Projects *should* be completed with a partner
- Find a project partner in your section (if you can)

The limits of collaboration

- One simple rule: Don't share your code, except with partners
- Copying project solutions is a serious offense!
- We really do catch people who violate the rules
  - We also know how to search the web for solutions
  - We let computers detect copying for us

Expressions

Types of expressions

An expression

describes a computation

and evaluates to a value

Types of expressions

An expression

describes a computation

and evaluates to a value

$$18 + 69$$

Types of expressions

An expression

describes a computation

and evaluates to a value

$$18 + 69$$

$$\frac{6}{23}$$

Types of expressions

An expression

describes a computation

and evaluates to a value

$18 + 69$

$\frac{6}{23}$

$\sqrt{3493161}$

Types of expressions

An expression

describes a computation

and evaluates to a value

$18 + 69$

$\frac{6}{23}$

$\sin \pi$

$\sqrt{3493161}$

Types of expressions

An expression

describes a computation

and evaluates to a value

$18 + 69$

$\frac{6}{23}$

$\sin \pi$

$\sqrt{3493161}$

$|-1869|$

Types of expressions

An expression

describes a computation

and evaluates to a value

$18 + 69$

$\frac{6}{23}$

$\sin \pi$

$\sqrt{3493161}$

$\sum_{i=1}^{100} i$

$|-1869|$

Types of expressions

An expression

describes a computation

and evaluates to a value

$18 + 69$

$\frac{6}{23}$

$\sin \pi$

$\sqrt{3493161}$

$\sum_{i=1}^{100} i$

$|-1869|$

$\binom{69}{18}$

Types of expressions

An expression

describes a computation

and evaluates to a value

$18 + 69$

$\sin \pi$

$\frac{6}{23}$

$\sqrt{3493161}$

$f(x)$

$\sum_{i=1}^{100} i$

$\binom{69}{18}$

$|-1869|$

Types of expressions

An expression

describes a computation

and evaluates to a value

$18 + 69$

$\frac{6}{23}$

$\sin \pi$

$f(x)$

$\sqrt{3493161}$

$\sum_{i=1}^{100} i$

$|-1869|$

$\binom{69}{18}$

Call Expressions in Python

All expressions can use function call notation

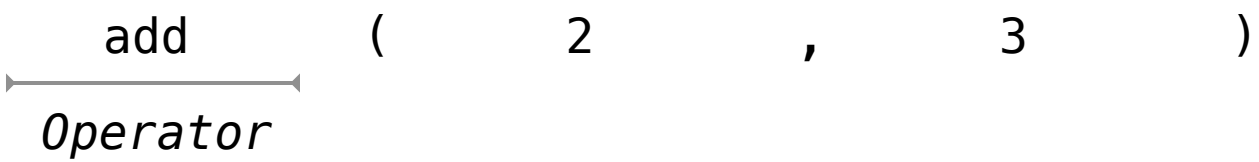
(Demo)

Anatomy of a Call Expression

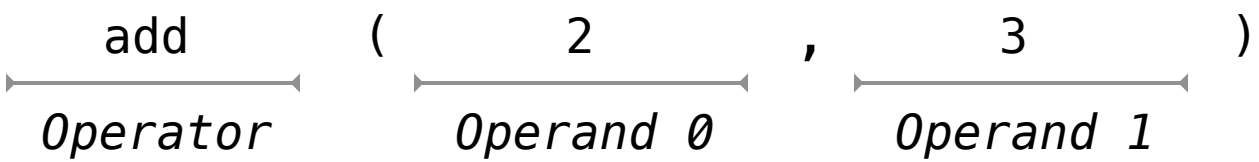
Anatomy of a Call Expression

```
add     (     2     ,     3     )
```

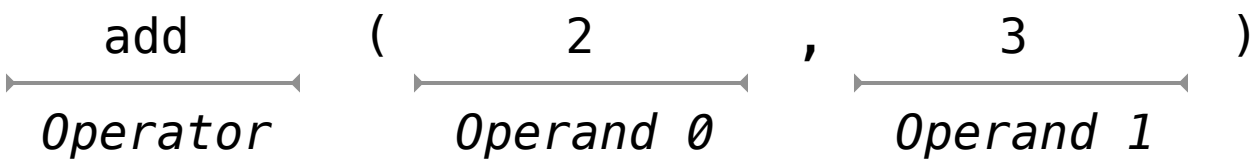
Anatomy of a Call Expression



Anatomy of a Call Expression

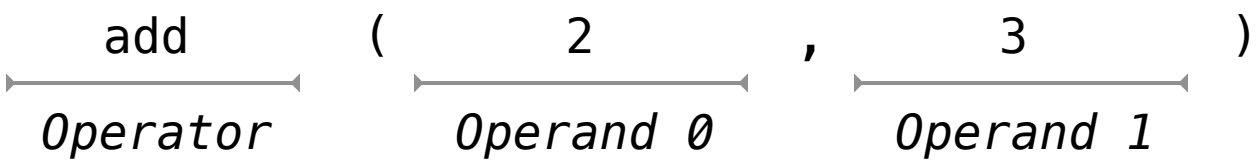


Anatomy of a Call Expression



Operators and operands are expressions

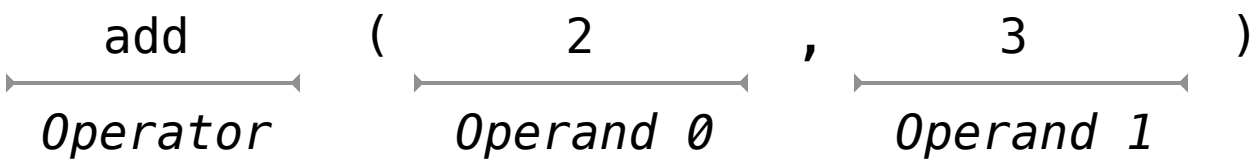
Anatomy of a Call Expression



Operators and operands are expressions

So they evaluate to values

Anatomy of a Call Expression

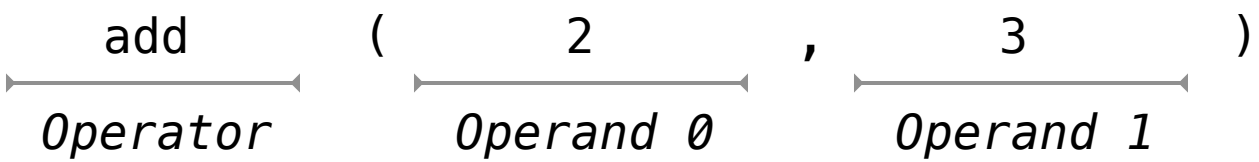


Operators and operands are expressions

So they evaluate to values

Evaluation procedure for call expressions:

Anatomy of a Call Expression



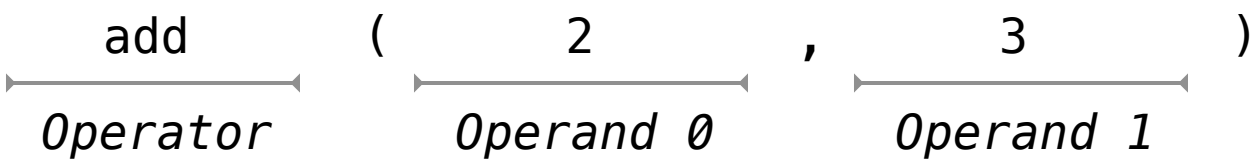
Operators and operands are expressions

So they evaluate to values

Evaluation procedure for call expressions:

1. Evaluate the operator and operand subexpressions

Anatomy of a Call Expression



Operators and operands are expressions

So they evaluate to values

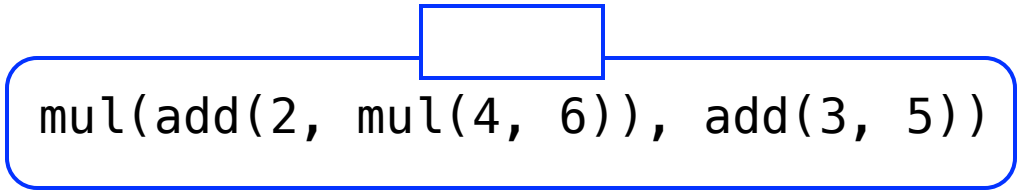
Evaluation procedure for call expressions:

1. Evaluate the operator and operand subexpressions
2. Apply the function that is the value of the operator subexpression to the arguments that are the values of the operand subexpression

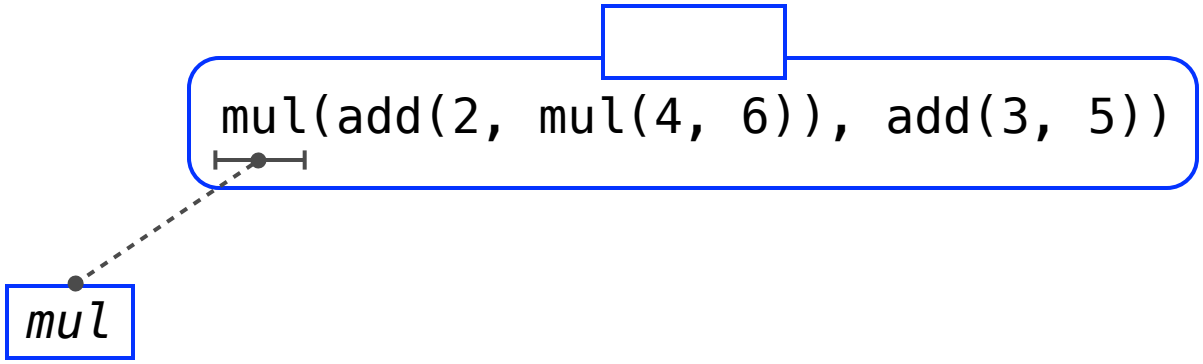
Evaluating Nested Expressions

```
mul(add(2, mul(4, 6)), add(3, 5))
```

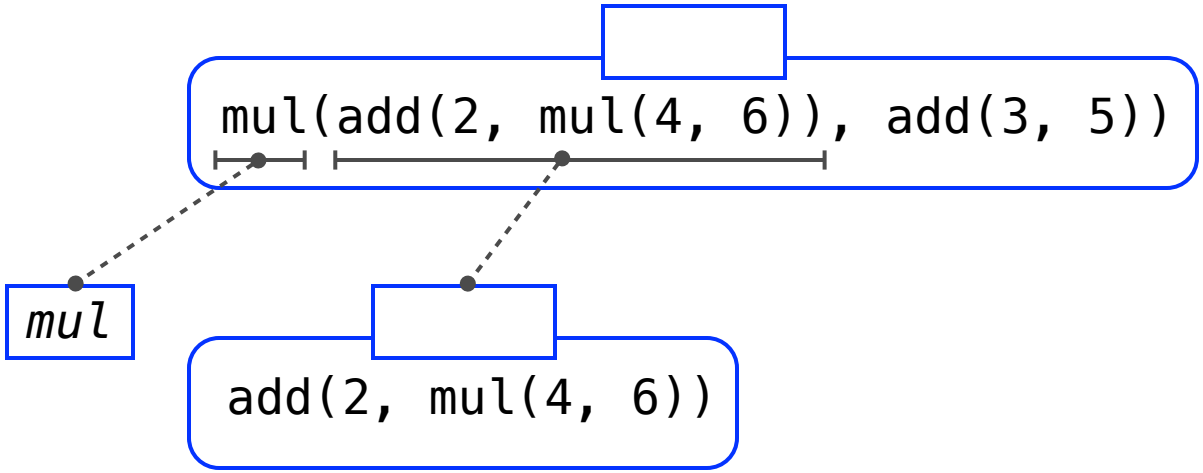
Evaluating Nested Expressions



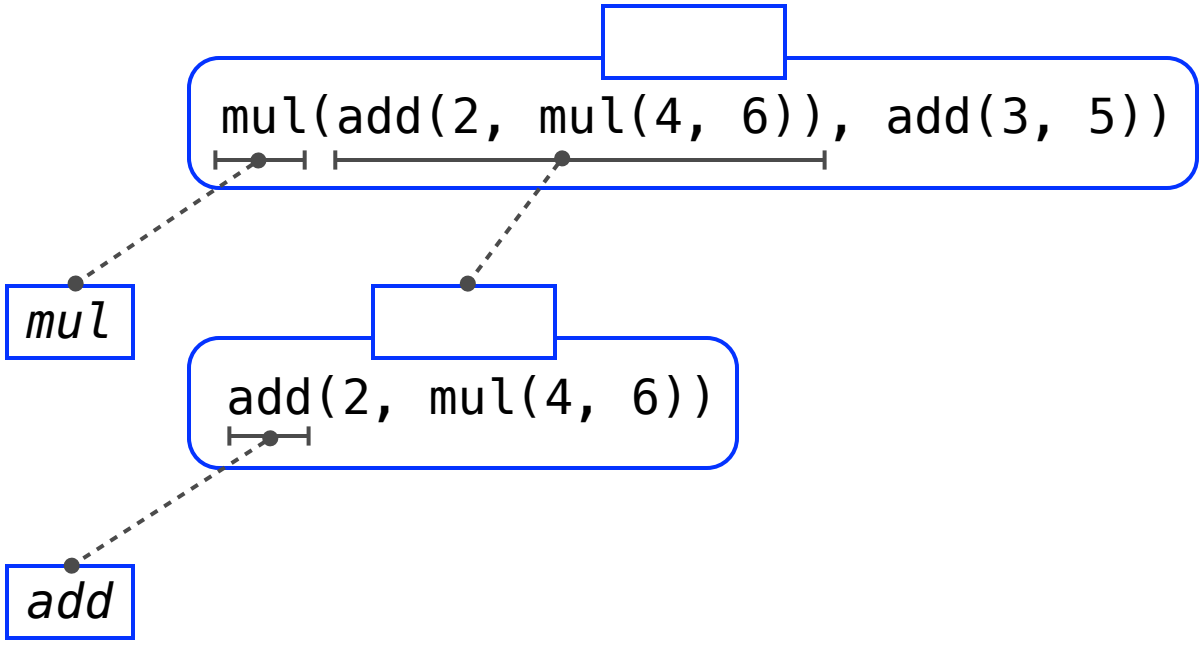
Evaluating Nested Expressions



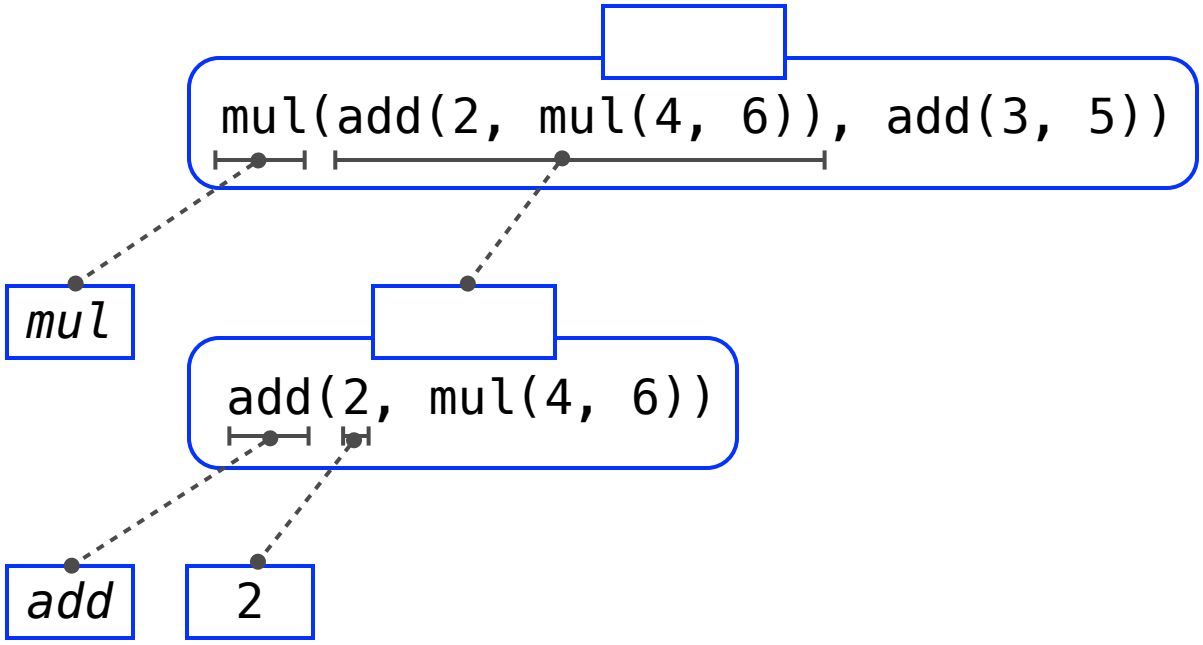
Evaluating Nested Expressions



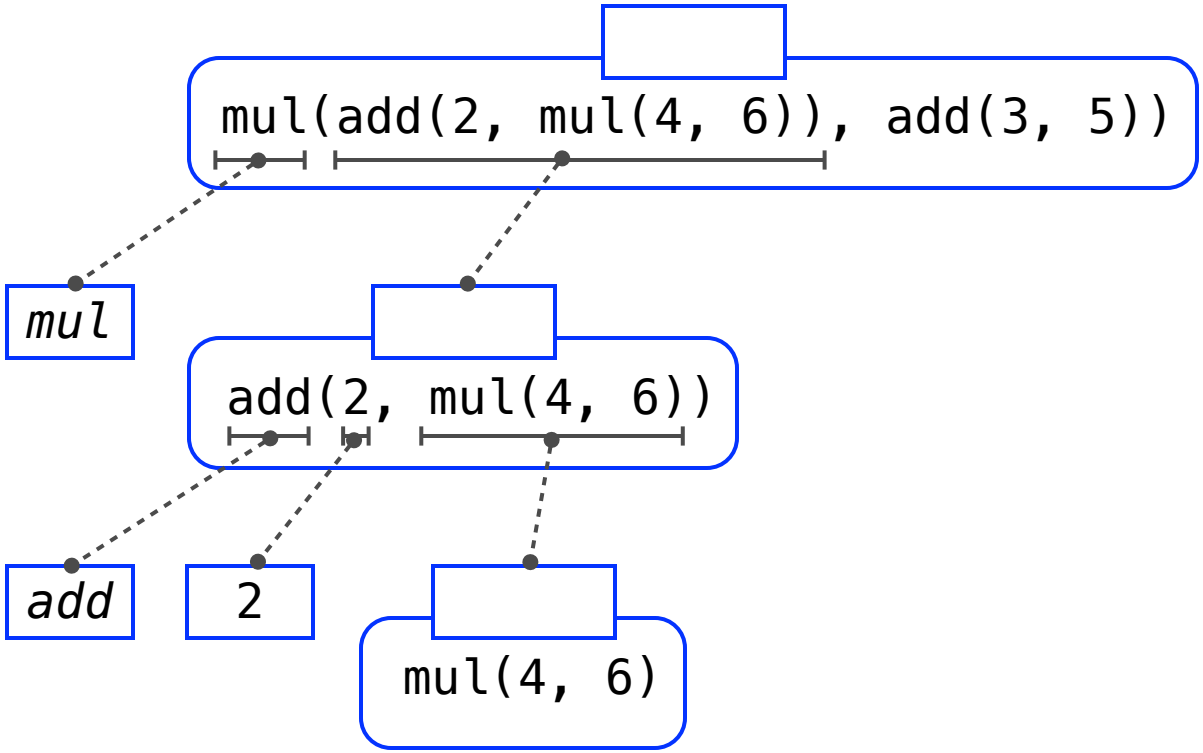
Evaluating Nested Expressions



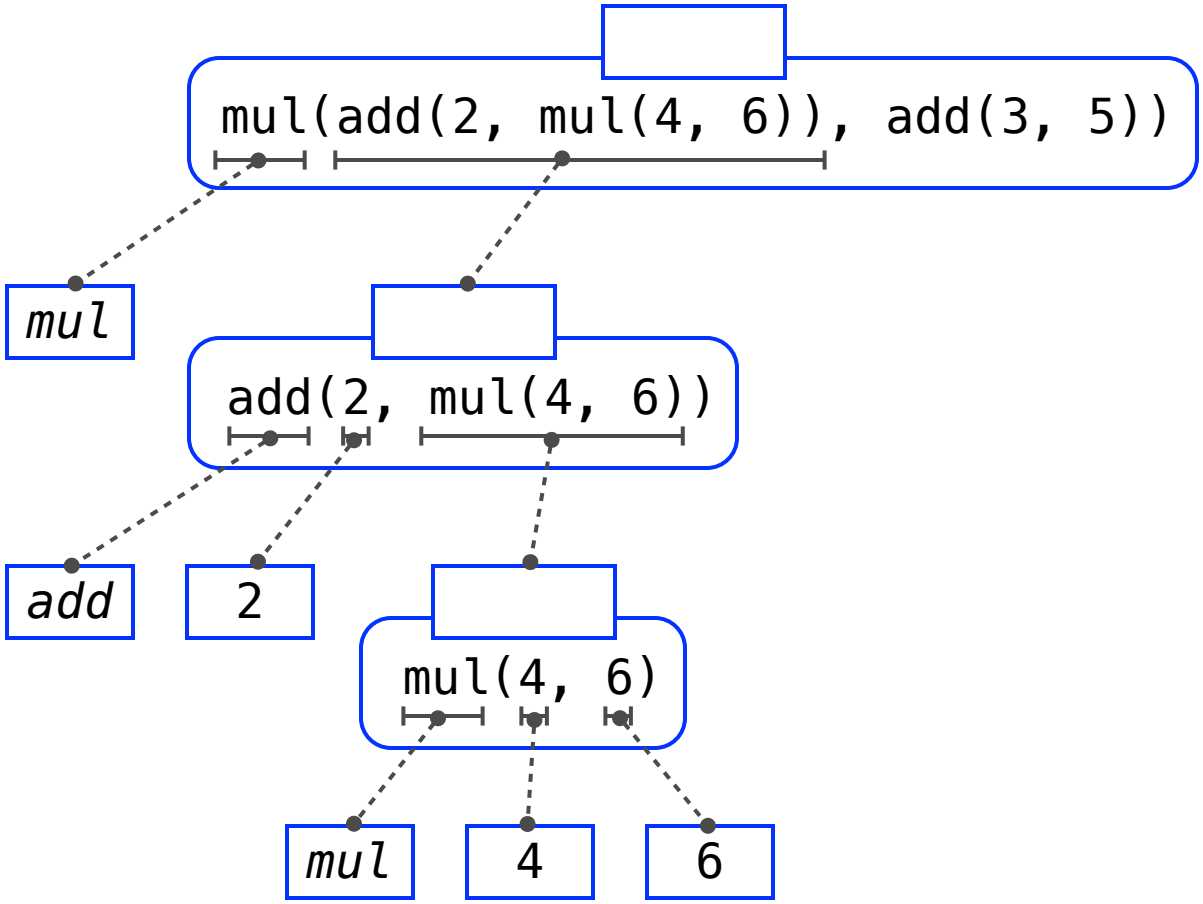
Evaluating Nested Expressions



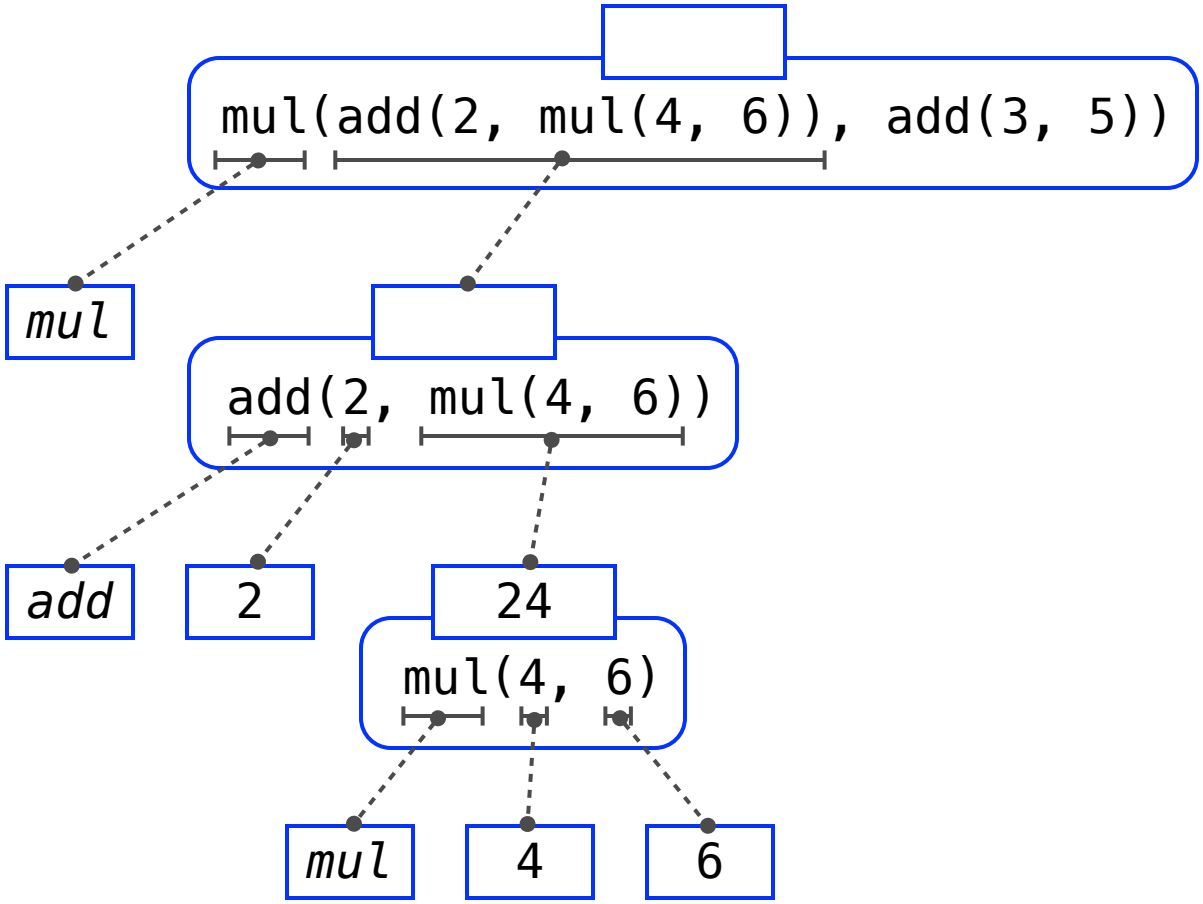
Evaluating Nested Expressions



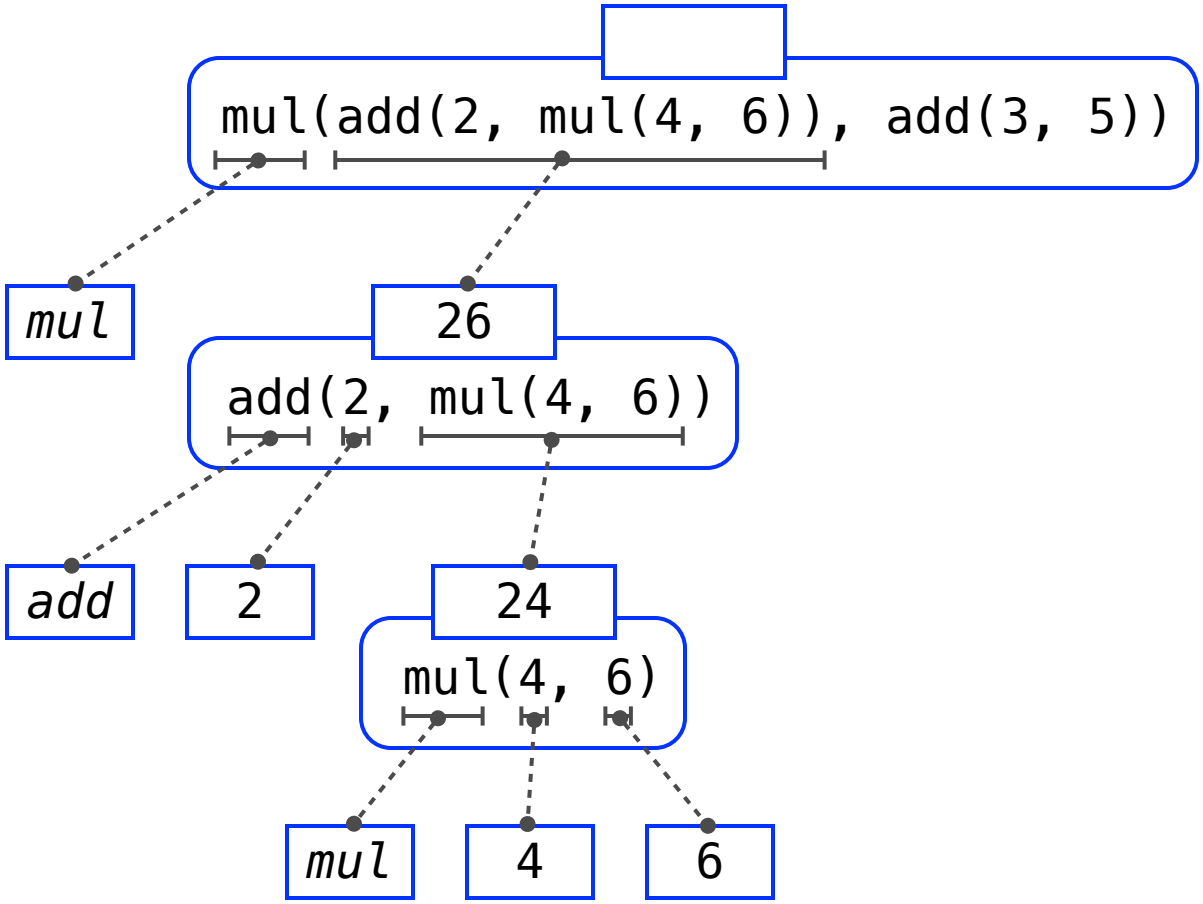
Evaluating Nested Expressions



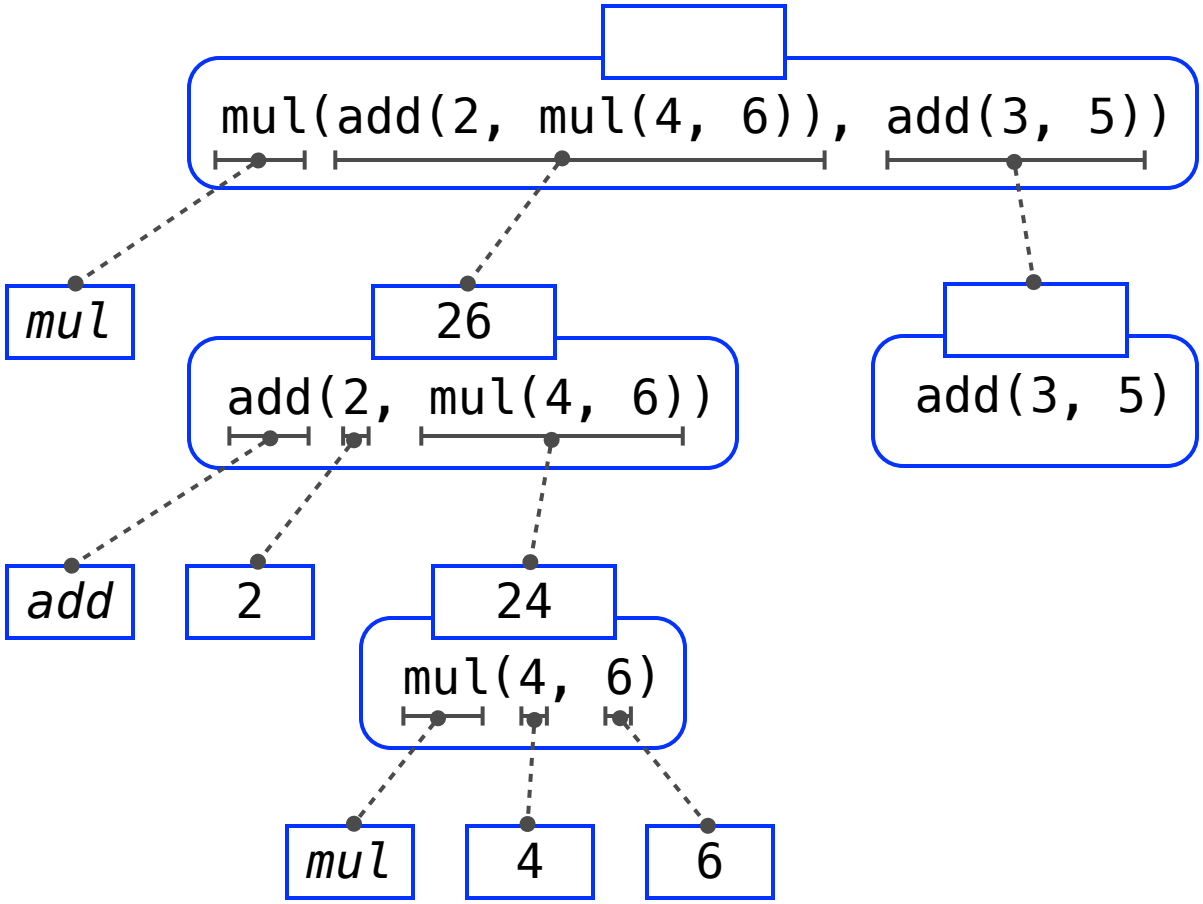
Evaluating Nested Expressions



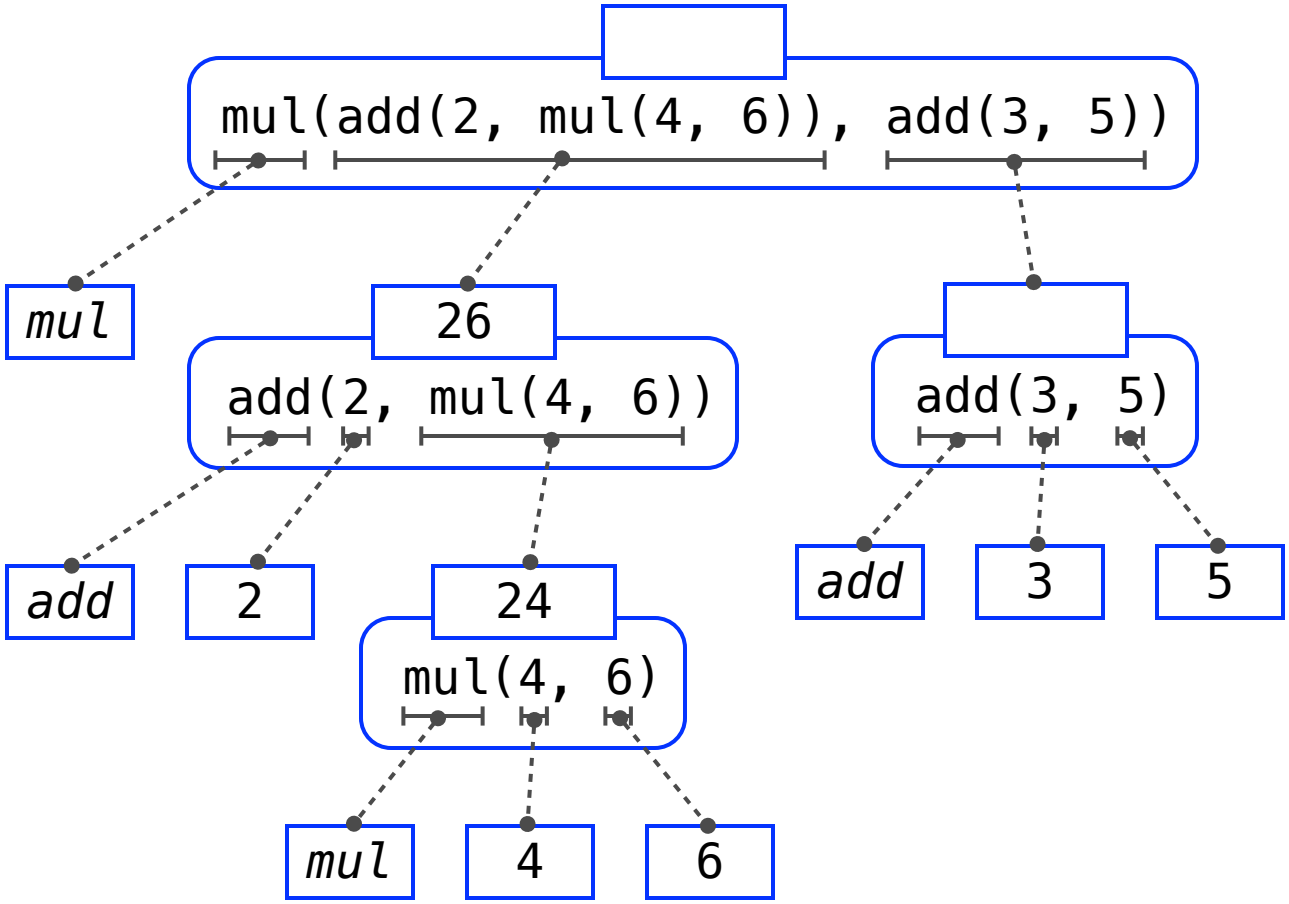
Evaluating Nested Expressions



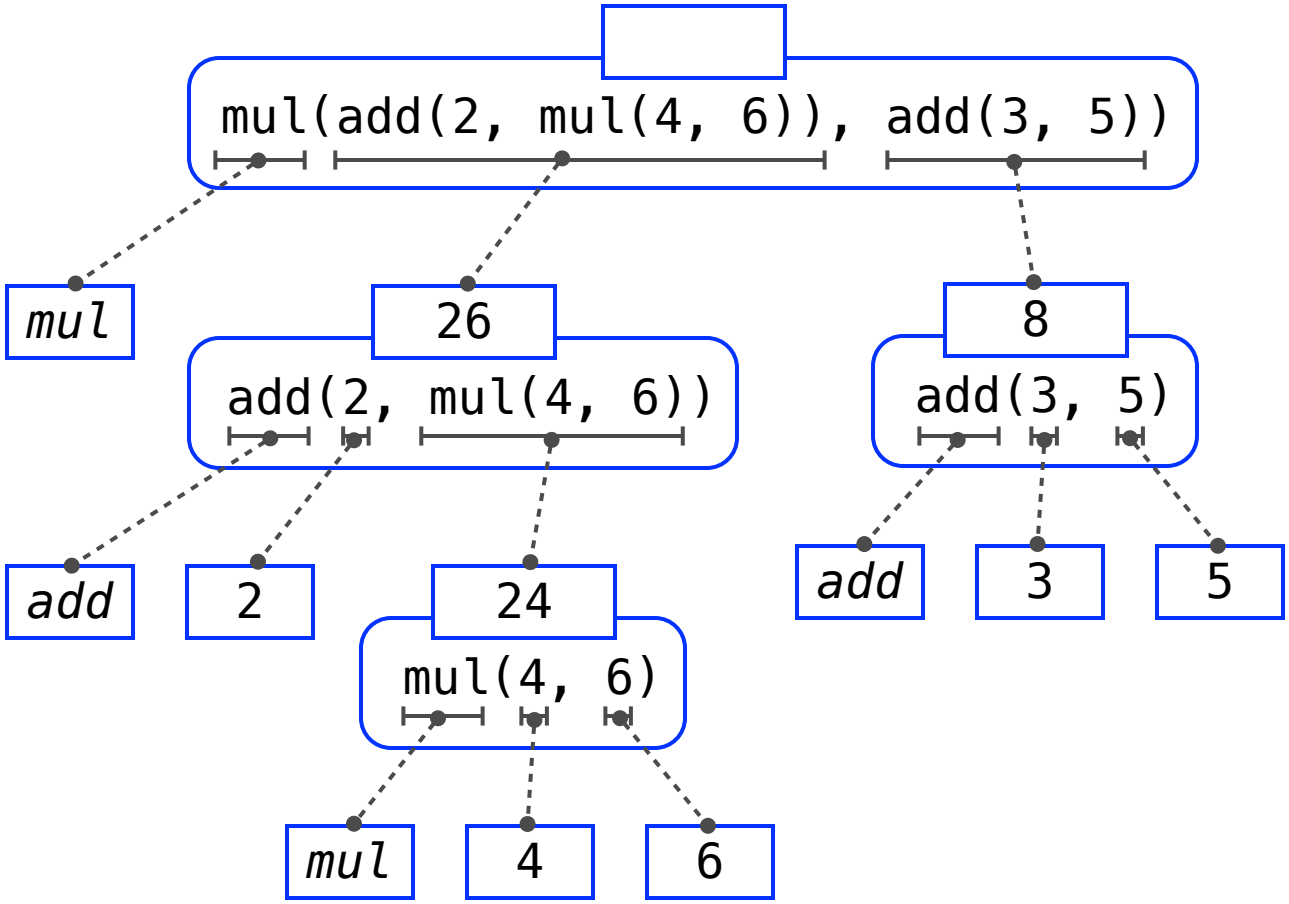
Evaluating Nested Expressions



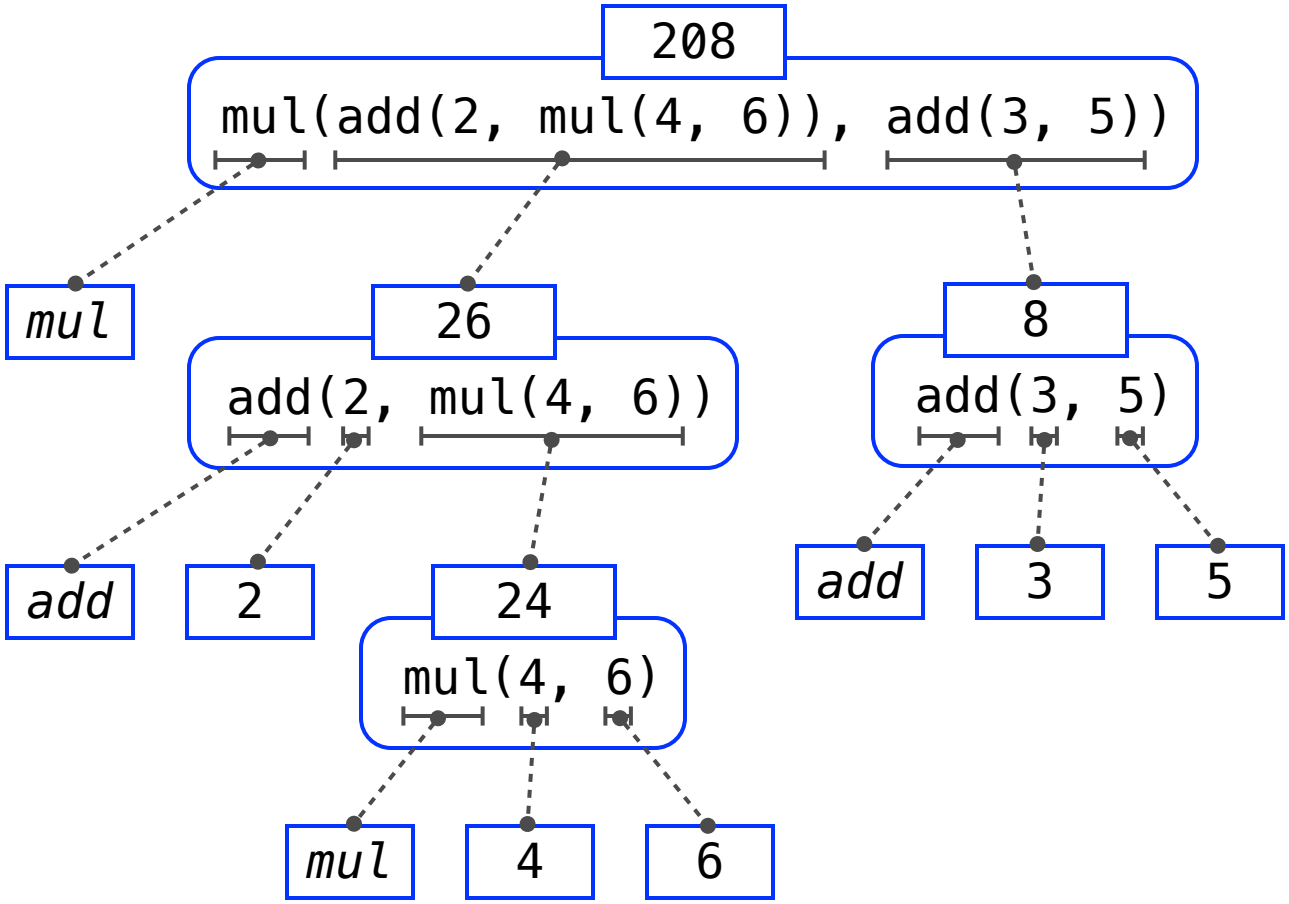
Evaluating Nested Expressions



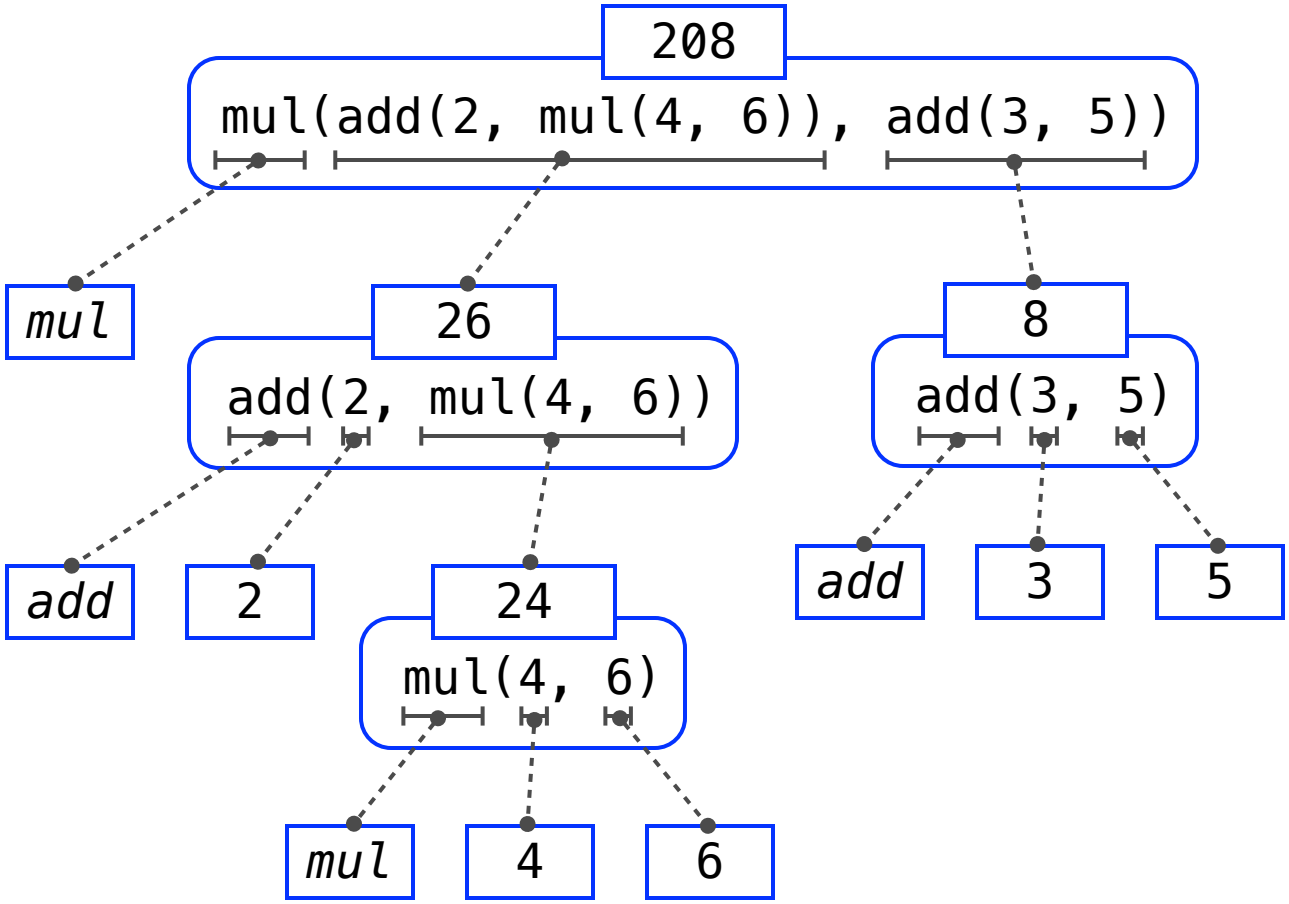
Evaluating Nested Expressions



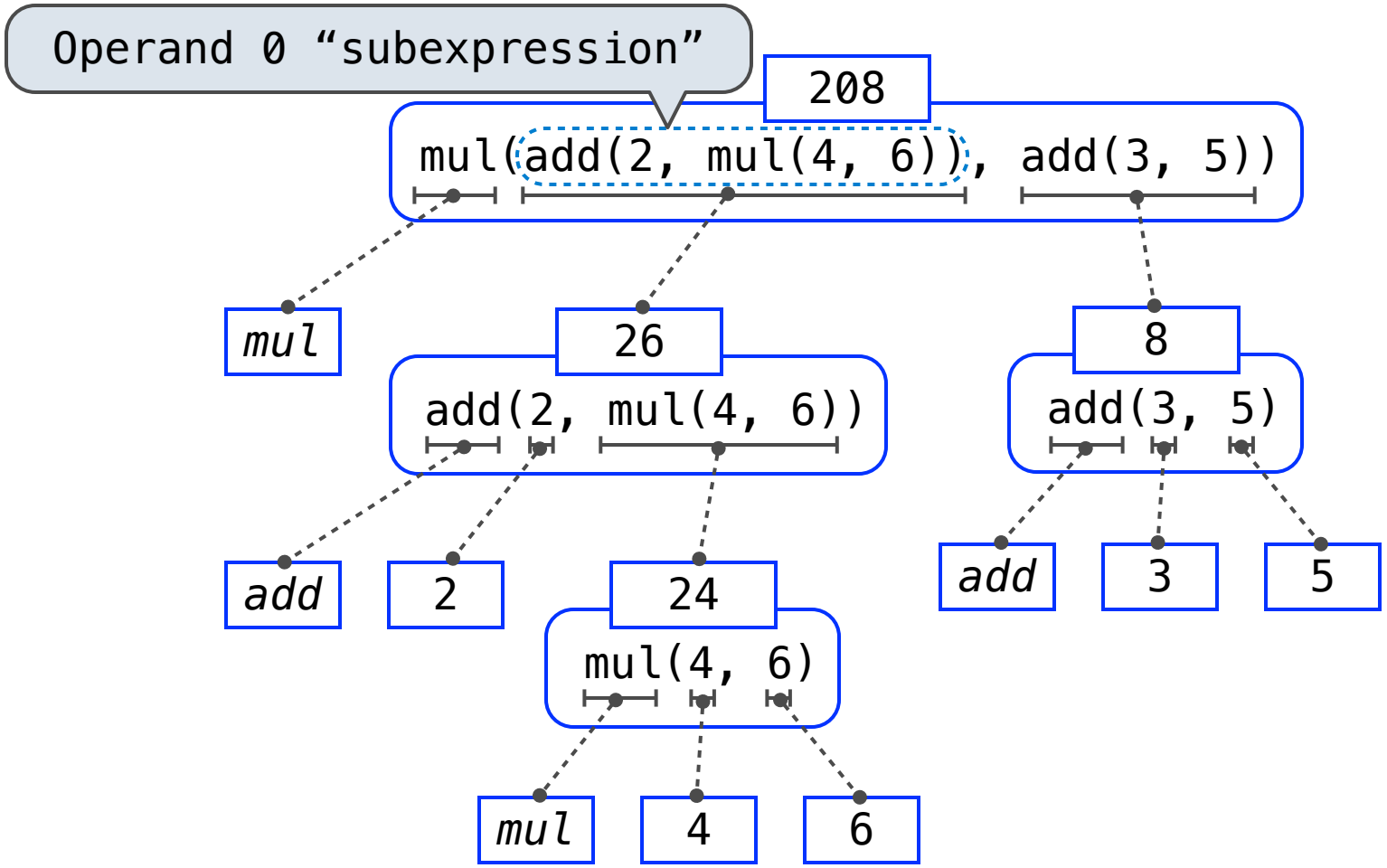
Evaluating Nested Expressions



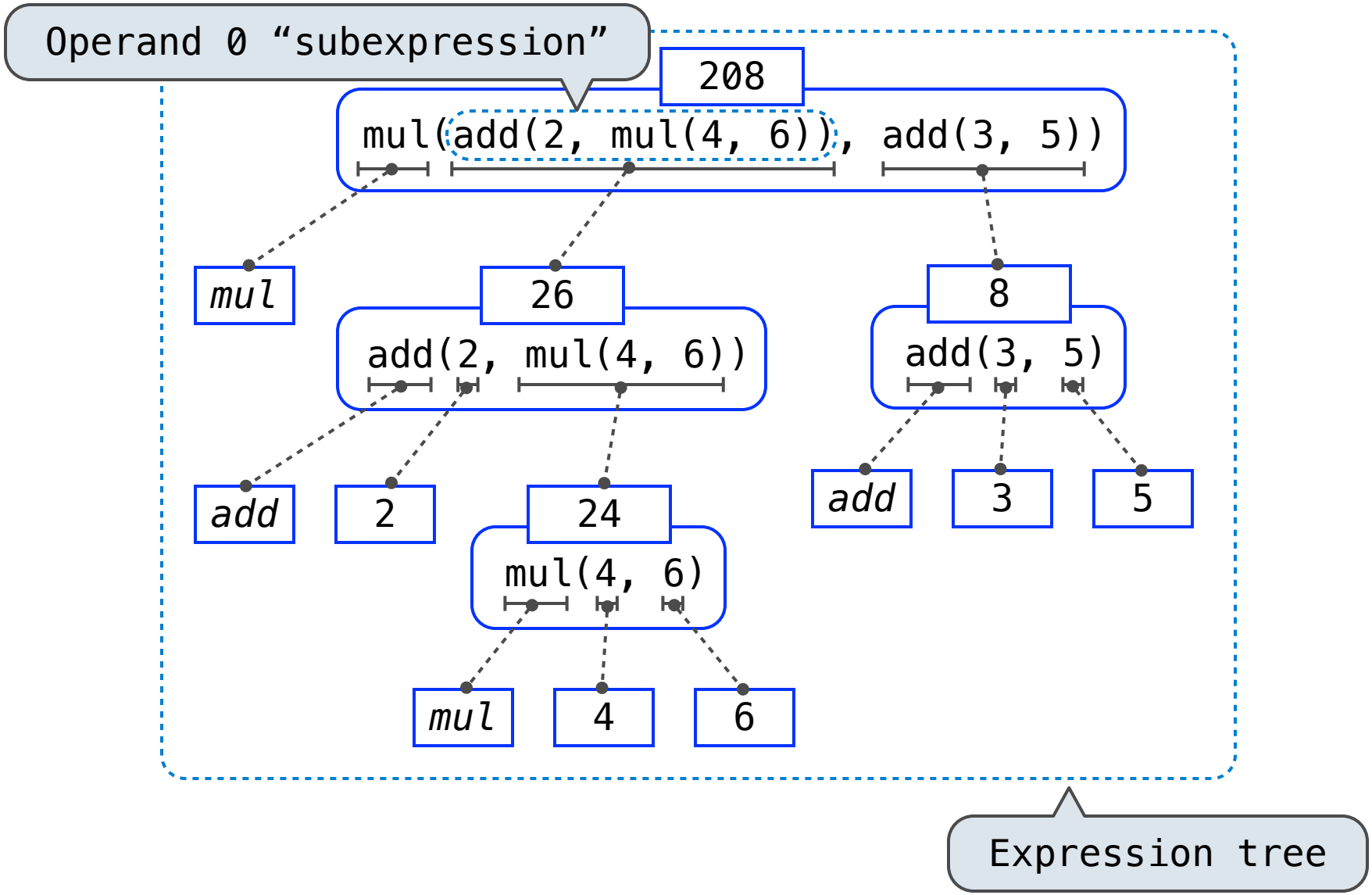
Evaluating Nested Expressions



Evaluating Nested Expressions



Evaluating Nested Expressions



Data, Functions, and Interpreters

Data, Functions, and Interpreters

Data: The things that programs fiddle with

Data, Functions, and Interpreters

Data: The things that programs fiddle with

```
2
```

Data, Functions, and Interpreters

Data: The things that programs fiddle with

“The Art of Computer Programming”

2

Data, Functions, and Interpreters

Data: The things that programs fiddle with

“The Art of Computer Programming”

2

Donald Knuth

Data, Functions, and Interpreters

Data: The things that programs fiddle with

“The Art of Computer Programming”

2

Shakespeare’s 37 plays

Donald Knuth

Data, Functions, and Interpreters

Data: The things that programs fiddle with

"The Art of Computer Programming"

2

Shakespeare's 37 plays

Donald Knuth

Functions: Rules for manipulating data

Data, Functions, and Interpreters

Data: The things that programs fiddle with

“The Art of Computer Programming”

2

Shakespeare’s 37 plays

Donald Knuth

Functions: Rules for manipulating data

Add up numbers

Data, Functions, and Interpreters

Data: The things that programs fiddle with

"The Art of Computer Programming"

2

Shakespeare's 37 plays

Donald Knuth

Functions: Rules for manipulating data

Count the words in a line of text

Add up numbers

Data, Functions, and Interpreters

Data: The things that programs fiddle with

"The Art of Computer Programming"

2

Shakespeare's 37 plays

Donald Knuth

Functions: Rules for manipulating data

Count the words in a line of text

Add up numbers

Pronounce someone's name

Data, Functions, and Interpreters

Data: The things that programs fiddle with

"The Art of Computer Programming"

2

Shakespeare's 37 plays

Donald Knuth
(Ka-NOOTH)

Functions: Rules for manipulating data

Count the words in a line of text

Add up numbers

Pronounce someone's name

Data, Functions, and Interpreters

Data: The things that programs fiddle with

"The Art of Computer Programming"

2

Shakespeare's 37 plays

Donald Knuth
(Ka-NOOTH)

Functions: Rules for manipulating data

Count the words in a line of text

Add up numbers

Pronounce someone's name

Interpreter: An implementation of the procedure for evaluation