CS 61A SU26

Lecture 02: Control

June 23, 2026

Rebecca Dang

Join pollev.com/rebeccadang831 (or scan QR code) on your phone or laptop

We'll begin at Berkeley time (10 minutes after), as per tradition!

CC BY NC SA

Potpourri

5 min

- Announcements
- Askademia
- Computing in the news
- WWPD: Functions

CC BY NC SA

Announcements

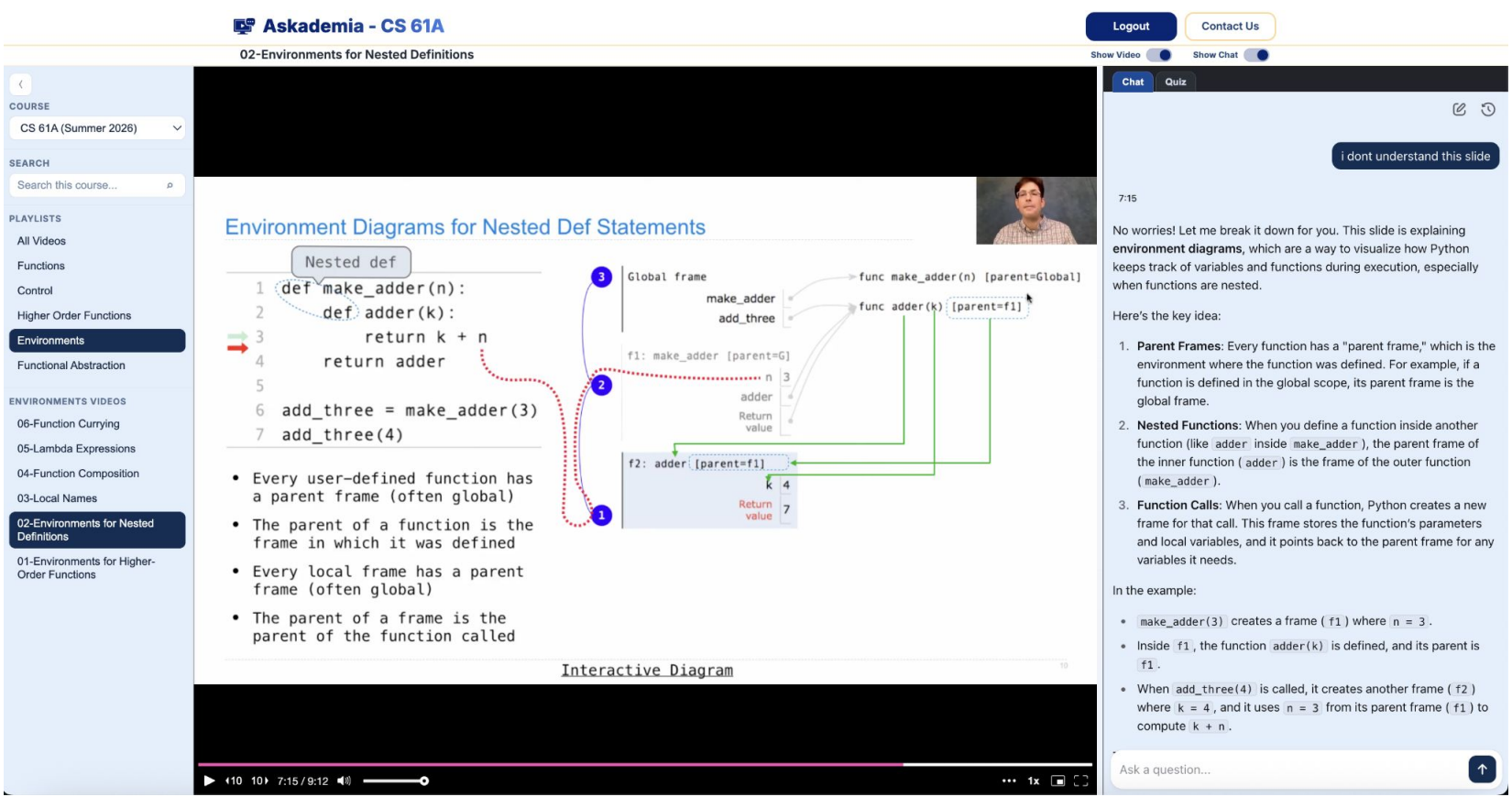
- Lec 1 recording is up on bCourses > Media Gallery (last bit cut off, just read the slides)
- **Please use your @berkeley.edu email for accessing all course emails**
  - All share requests to external emails will be denied
- If you need to be assigned a lab/disc section or reschedule due to an **immovable time conflict**, email cs61a@berkeley.edu ASAP
- If you were in Esha's lab yesterday and lost your **keys** and/or **Berkeley Bowl tote bag**, retrieve it from the **Soda Hall lost and found** (3rd floor front desk, across from Soda 384) [Ed post]

CC BY NC SA

Askademia

- **[new resource]** Askademia: A Berkeley research project (paper) that allows students to ask questions during live and recorded lectures and it responds in-context
- askademia.org/cs61a/su26
- Video playlists linked on course website under each lecture (if applicable), roughly correspond to the YouTube playlists
- **As always, be careful about hallucinations or out of scope content**
- This semester's assignments/lectures/labs/discussions are the "canonical" scope

CC BY NC SA



Askademia interface

Computing in the news

Yesterday: Google Invests $75 Million in A24 to Develop AI-Powered Filmmaking Tools (Variety)

- A24 is a studio that has developed many critically acclaimed films, including *Moonlight*, *Everything Everywhere All At Once*, and most recently, *Backrooms*
- DeepMind is a AI subsidiary of Google behind AlphaGo (AI Go player that beat one of the best human players) and AlphaFold (protein folding prediction, 2024 Nobel Prize in Chemistry)
- **Big contention in creative communities about the role of AI**
  - Should artists use AI to make or help them make art?
  - How will this affect creative jobs? How ethical is it to use AI models trained on art without the artist's consent or compensating artists?
  - Is AI capable of making art? What even is *art*?

CC BY NC SA

When poll is active respond at **PollEv.com/rebeccadang831**

Activate this Poll with the Poll Everywhere Live app.

If video conferencing, you must be sharing your full screen to share the activated poll.

Functions: What would Python display?

Boolean Expressions

15 min

- Booleans
- Truthy or Falsy
- Boolean expressions
  - Inequality operators
  - Logical operators
- Short circuiting

CC BY NC SA

Booleans

- Primitive data type that has only 2 possible values: `True` or `False`
- We use it to represent truth values, aka "answers to yes or no questions"
  - Is your age 18 or older?
  - Have you done your homework and your chores?
  - Is the light on?

CC BY NC SA

Truthy and Falsy Values

- In Python, *all* data types can behave like booleans, even if they aren't actually booleans!
- **Falsy** values: `False`, `0`, `0.0`, `None`, or empty containers, such as an empty string (`''`)
  - How to remember: "Falsy if it's zero or represents nothing"
- **Truthy** values: Everything else

CC BY NC SA

Boolean expressions

- A **boolean expression** is an expression that evaluates to some truthy or falsy value
- You can construct boolean expressions by combining values or names with **operators**
  - Inequality operators: `>`, `<`, `>=`, `<=`, `==`, `!=`
  - Logical operators: `and`, `or`, `not`

```
>>> age = 21
>>> age > 18
True
>>> age > 21
False
```

```
>>> hw_done = True
>>> chores_done = False
>>> can_scroll = hw_done and chores_done
>>> can_scroll
False
```


CC BY NC SA

and operator truth table: both must be T → T

	T	**F**
T	T	F
F	F	F

CC BY NC SA

or operator truth table: at least one must be T → T

	T	**F**
T	T	T
F	T	F

CC BY NC SA

`not` operator: flips the current value

- `not True` → `False`
- `not False` → `True`

CC BY NC SA

When poll is active respond at **PollEv.com/rebeccadang831**

Activate this Poll with the Poll Everywhere Live app.

If video conferencing, you must be sharing your full screen to share the activated poll.

Booleans: What would Python display?

What just happened? Short circuiting!

- Motivation: Python doesn't want to do more computation than is necessary to evaluate a boolean expression
  - Exit early if needed
    - `and`: Can stop after the first falsy value
    - `or`: Can stop after the first truthy value
  - The full expression evaluates to the "last looked at" value (not necessarily a boolean data type)
- Another thing that was going on: Operator precedence
  - If no parentheses provided, Python "groups together" `not`, then `and`, then `or`

`5 or (6 / 0) and not None` is equivalent to

```
5 or ((6 / 0) and (not None))
```


CC BY NC SA

Conditional Statements

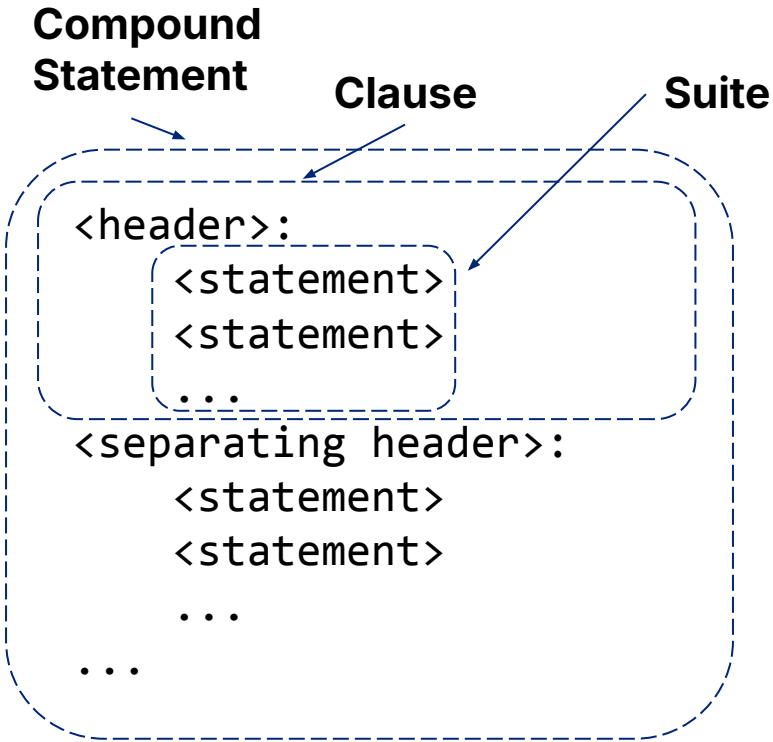
15 min

- Statements
- Conditional Statements (If-Elif-Else)

CC BY NC SA

Statements

A **statement** is executed by the interpreter to perform an action



The **header** determines the statement type

The header of a clause "controls" the suite that follows

So far, we've only learned one compound statement: `def` statements

CC BY NC SA

Conditional Statements

- Motivation: How can we only execute a set of statements (suite) under certain conditions?
  - Ex: If the user is a TA, display the student roster. Else, hide the student roster page.
- Solution: **Conditional statements** (aka if statements)

```
if <expression>:
    <suite>
elif <expression>:
    <suite>
else:
    <suite>
```


cc BY NC SA

Example: absolute value

```
def absolute_value(x):
    if x < 0:
        return -x
    elif x == 0:
        return 0
    else:
        return x
```

(really, you should just use the built-in `abs` function)

CC BY NC SA

Example: absolute value (simplified, logically equivalent)

```
def absolute_value(x):
    if x < 0:
        return -x
    else:
        return x
```

(`elif` and `else` are optional, and you can have multiple `elif` clauses!)

CC BY NC SA

When poll is active respond at **PollEv.com/rebeccadang831**

Activate this Poll with the Poll Everywhere Live app.

If video conferencing, you must be sharing your full screen to share the activated poll.

Conditionals: What would Python display? (one per line)

When poll is active respond at **PollEv.com/rebeccadang831**

Activate this Poll with the Poll Everywhere Live app.

If video conferencing, you must be sharing your full screen to share the activated poll.

Conditionals II: What would Python display? (one per line)

When poll is active respond at **PollEv.com/rebeccadang831**

Activate this Poll with the Poll Everywhere Live app.

If video conferencing, you must be sharing your full screen to share the activated poll.

True or false: my_function1 is equivalent to my_function2.

5 min break

CC BY NC SA

Iteration

30 min

- While loop
- Infinite loop
- Practice

CC BY NC SA

While loop

- Motivation: What if I want to repeat a set of statements multiple times?
  - (Naive) solution: Copy and paste the code blocks
  - (Better) solution: Use iteration or loops
- While loops are a certain kind of iterative statement:
  - While some boolean condition is true, execute everything in the body of the loop

```
while <condition>:
    <suite>
```


CC BY NC SA

Demo: Add all integers from 1 to 100

```
>>> i = 1
>>> total = 0
>>> while i <= 100:
...     total += i
...
<runs forever, infinite loop>
```

```
>>> i = 1
>>> total = 0
>>> while i <= 100:
...     total += i
...     i += 1
...
>>> total
5050
```


cc BY NC SA

Demo: Alternate solutions

```
>>> i = 1
>>> total = 0
>>> while i < 101:
...     total += i
...     i += 1
...
>>> total
5050
```

```
>>> i = 100
>>> total = 0
>>> while i > 0:
...     total += i
...     i -= 1
...
>>> total
5050
```


CC BY NC SA

Practice: Add All Even Div Three

- Implement a function `add_all_even_div_three` which takes a positive integer `n` and returns the sum of all even numbers from 1 to `n` that are also divisible by 3
- Download starter code `02.py` from the course website under Lecture 2
- Run doctests with `python3 -m doctest 02.py`
- 5 min: Try implementing it yourself
- 5 min: Discuss with someone next to you and compare your implementations
  - What worked?
  - What didn't?
  - Why?

CC BY NC SA

Practice: Fizz Buzz

- Implement a function `fizzbuzz` which takes a positive integer `n` and prints the numbers from 1 to `n`, replacing:
  - Multiples of 3 with `"fizz"`
  - Multiples of 5 with `"buzz"`, and
  - Multiples of both with `"fizzbuzz"`
- Download starter code `02.py` from the course website under Lecture 2
- Run doctests with `python3 -m doctest 02.py`
- 5 min: Try implementing it yourself
- 5 min: Discuss with someone next to you and compare your implementations
  - What worked?
  - What didn't?
  - Why?

CC BY NC SA

Practice: Sum Digits (Lab 01 Q10)

- Implement a function `sum_digits` which takes a positive integer `n` and returns the sum of the digits of `n`
- Download starter code `02.py` from the course website under Lecture 2
- Run doctests with `python3 -m doctest 02.py`
- 5 min: Try implementing it yourself
- 5 min: Discuss with someone next to you and compare your implementations
  - What worked?
  - What didn't?
  - Why?

CC BY NC SA