UC Berkeley EECS
Lecturer
Michael Ball

Computational Structures in Data Science

Lecture 2:
Abstraction and Functions

Computing In The News

- How game-makers are catering to disabled players

Ars Technica, 8/29/2021

According to a recent study, more than 2 percent of the US population can't play video games due to poor accessibility options. This same study suggests more than 9 percent are unable to enjoy the traditional gaming experience because of visual, cognitive, or physical impairments. Additional research suggests 20 percent of the casual gaming audience is disabled in some fashion.

The Microsoft Adaptive Controller is easily the most prominent example of adaptive controls. With 19 different 3.5 mm jacks, it can be mounted for players who cannot hold or manipulate standard controllers.

Announcements

- We are working to expand the course. Details TBD.
  - Usually ~30-40 people get off the waitlist.
  - This year it keeps growing. ☹
- Join the EECS 101 and DATA 001 Ed Discussions!
  - https://eecs.link/join-ed
  - https://eecs.link/data-ed
- Hopefully not needed! *Please,* report any concerns about class / campus climate to the department. *You* are welcome here!
- https://eecs.link/climate

Links

- Q&A Thread: https://go.c88c.org/qa2
- Self-Check: https://go.c88c.org/2

UC Berkeley EECS
Lecturer
Michael Ball

Computational Structures in Data Science

Abstraction

Abstraction

- Detail removal

  "The act of leaving out of consideration one or more properties of a complex object so as to attend to others."

- Generalization

  "The process of formulating general concepts by abstracting common properties of instances"

- Technical terms: Compression, Quantization, Clustering, Unsupervized Learning

H MATISSE 52

Henri Matisse *"Naked Blue IV"*

Experiment

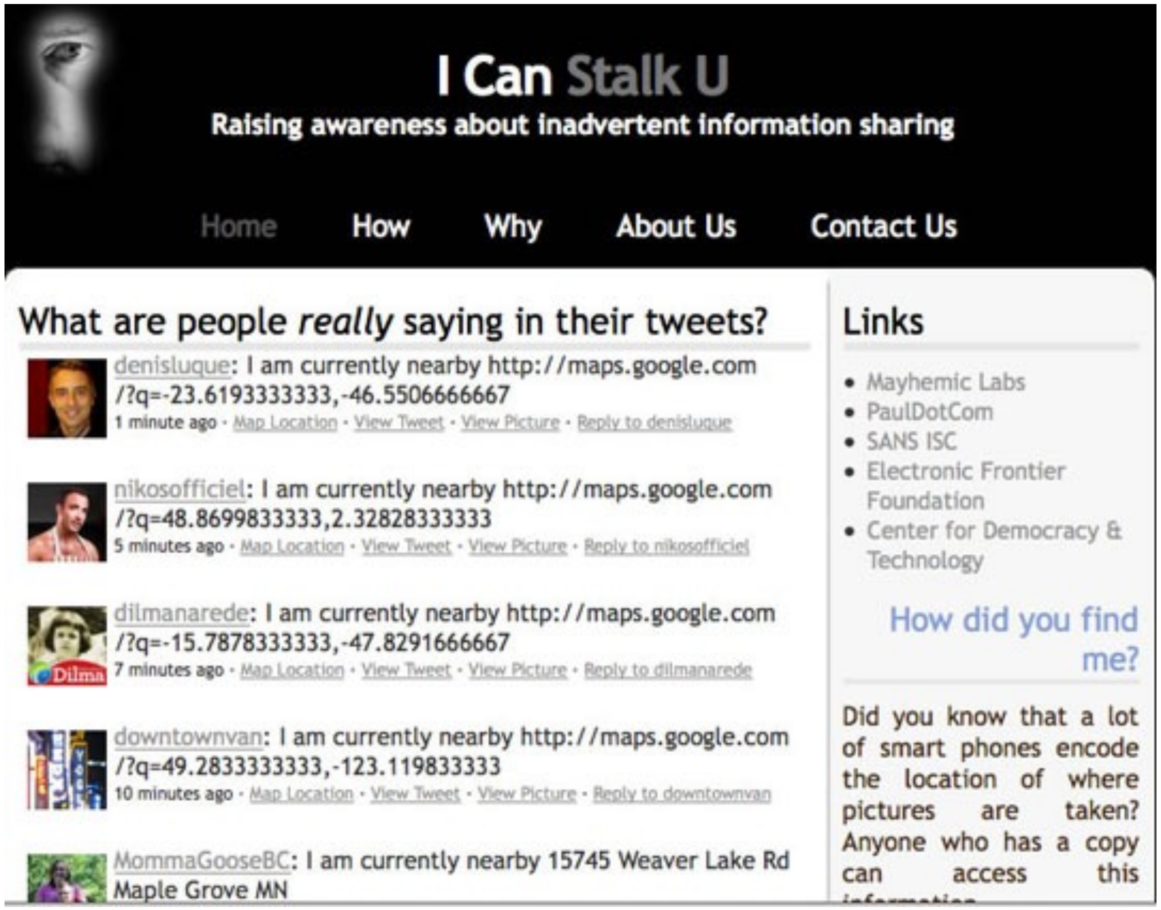
Where are you from?

Possible Answers:

- Planet Earth
- Europe
- California
- The Bay Area
- San Mateo
- 1947 Center Street, Berkeley, CA
- 37.8693° N, 122.2696° W

All correct but different levels of abstraction!

Abstraction gone wrong!



I Can Stalk U

Raising awareness about inadvertent information sharing

Home · How · Why · About Us · Contact Us

What are people *really* saying in their tweets?

denisluque: I am currently nearby http://maps.google.com/?q=-23.6193333333,-46.5506666667
1 minute ago · Map Location · View Tweet · View Picture · Reply to denisluque

nikosofficiel: I am currently nearby http://maps.google.com/?q=48.8699833333,2.32828333333
5 minutes ago · Map Location · View Tweet · View Picture · Reply to nikosofficiel

dilmanarede: I am currently nearby http://maps.google.com/?q=-15.7878333333,-47.8291666667
7 minutes ago · Map Location · View Tweet · View Picture · Reply to dilmanarede

downtownvan: I am currently nearby http://maps.google.com/?q=49.2833333333,-123.119833333
10 minutes ago · Map Location · View Tweet · View Picture · Reply to downtownvan

MommaGooseBC: I am currently nearby 15745 Weaver Lake Rd Maple Grove MN

Links

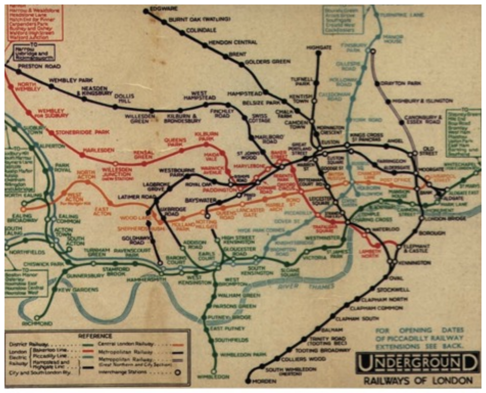
- Mayhemic Labs
- PaulDotCom
- SANS ISC
- Electronic Frontier Foundation
- Center for Democracy & Technology

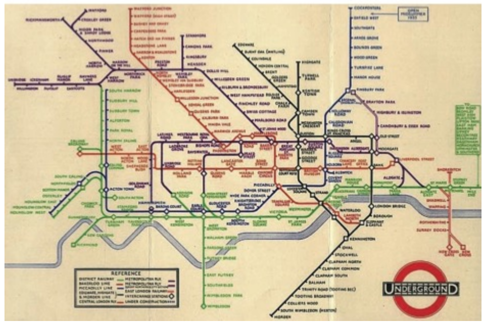
How did you find me?

Did you know that a lot of smart phones encode the location of where pictures are taken? Anyone who has a copy can access this

Detail Removal (in Data Science)

- You'll want to look at only the interesting data, leave out the details, zoom in/out…
- Abstraction is the idea that you focus on the essence, the cleanest way to map the messy real world to one you can build
- Experts are often brought in to know what to remove and what to keep!





The London Underground 1928 Map & the 1933 map by Harry Beck.

The Power of Abstraction, Everywhere!

- Examples:
  - Math Functions (e.g., sin x)
  - Hiring contractors
  - Application Programming Interfaces (APIs)
  - Technology (e.g., cars)

- Amazing things are built when these layer
  - And the abstraction layers are getting deeper by the day!

We only need to worry about the interface, or specification, or contract NOT how (or by whom) it's built

Above the abstraction line

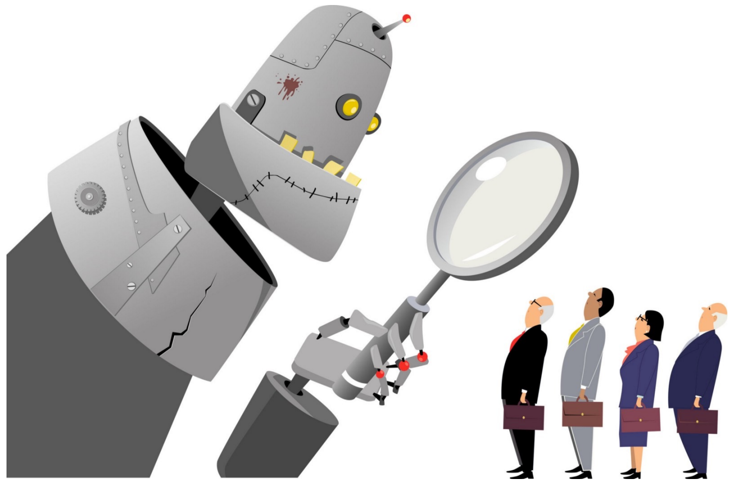
Abstraction Barrier (Interface)
(the interface, or specification, or contract)

Below the abstraction line

This is where / how / when / by whom it is actually built, which is done according to the interface, specification, or contract.

Abstraction: Pitfalls

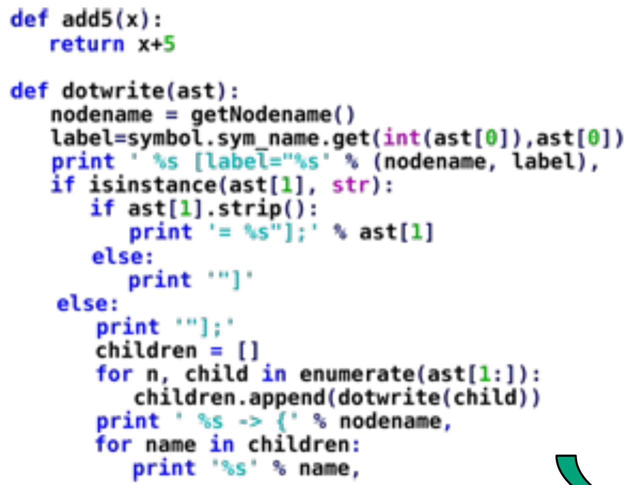
- Abstraction is not universal without loss of information (mathematically provable). This means, in the end, the complexity can only be "moved around"
- Abstraction makes us forget how things actually work and can therefore hide bias. Example: AI and hiring decisions.



- Abstractions can formalize a design or pattern. When something doesn't follow that pattern–perhaps a new use case emerges–it can be a burden to adapt.

Data or Code? Abstraction!

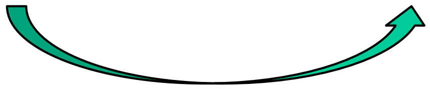
Human-readable code
(programming language)



```
def add5(x):
    return x+5

def dotwrite(ast):
    nodename = getNodename()
    label=symbol.sym_name.get(int(ast[0]),ast[0])
    print ' %s [label="%s' % (nodename, label),
    if isinstance(ast[1], str):
        if ast[1].strip():
            print '= %s"];' % ast[1]
        else:
            print '"]'
    else:
        print '"];'
        children = []
        for n, child in enumerate(ast[1:]):
            children.append(dotwrite(child))
        print ' %s -> {' % nodename,
        for name in children:
            print '%s' % name,
```

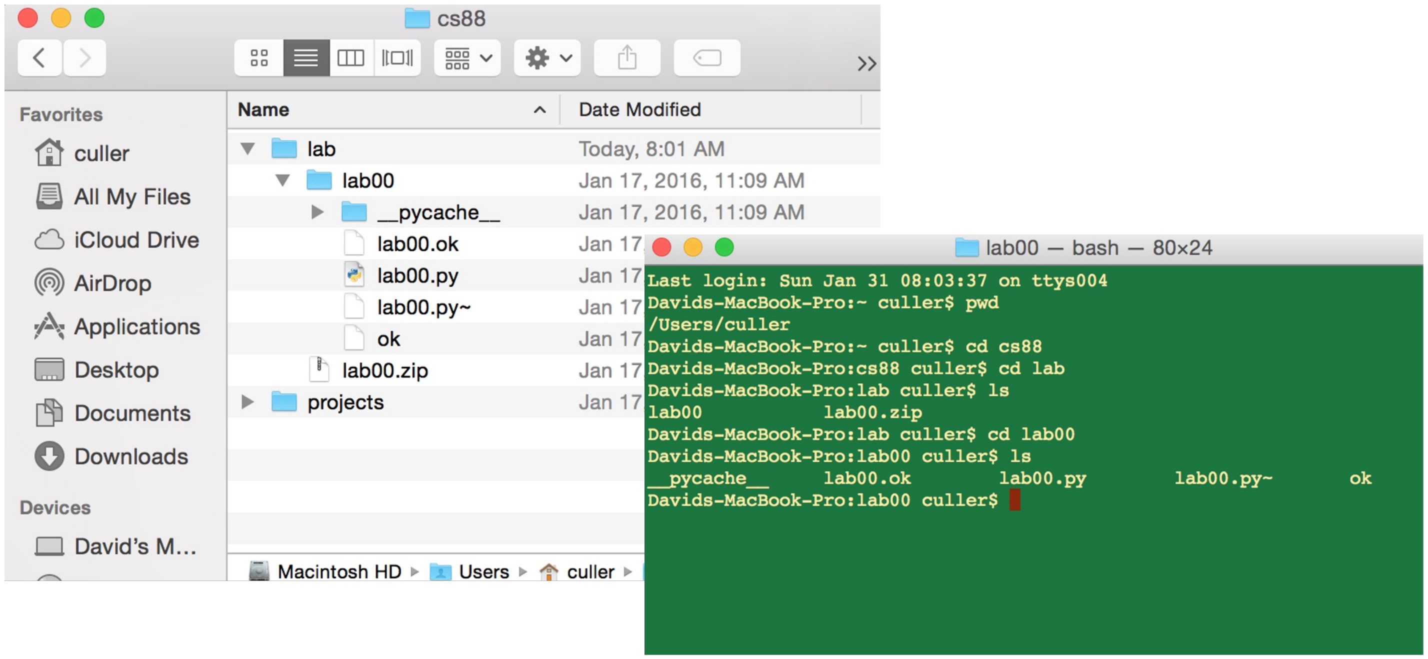
Machine-executable
instructions (byte code)



Compiler or Interpreter

Here: Python

Code or GUI: More Abstraction!



```
Last login: Sun Jan 31 08:03:37 on ttys004
Davids-MacBook-Pro:~ culler$ pwd
/Users/culler
Davids-MacBook-Pro:~ culler$ cd cs88
Davids-MacBook-Pro:cs88 culler$ cd lab
Davids-MacBook-Pro:lab culler$ ls
lab00           lab00.zip
Davids-MacBook-Pro:lab culler$ cd lab00
Davids-MacBook-Pro:lab00 culler$ ls
__pycache__     lab00.ok        lab00.py        lab00.py~       ok
Davids-MacBook-Pro:lab00 culler$ 
```

- Big Idea: Layers of Abstraction
  - The GUI look and feel is built out of files, directories, system code, etc.

Review:

- Abstraction:
  - Detail Removal or Generalizations
- Code:
  - Is an abstraction!

Computer Science is the study of abstraction

UC Berkeley EECS
Lecturer
Michael Ball

Computational Structures in Data Science

Python: Simple Statements

Learning Objectives

- Evaluate Python Expressions
- Call Functions in Python
- Assign data to Variables

Let’s talk Python

• Expression	`3.1 * 2.6`
• *Call* expression	`max(0, x)`
• Variables	`my_name`
• Assignment Statement	`my_name = <expression>`
• Define Statement:	`def function_name(<arguments>):`

• Control Statements:	`if …`
	`for …`
	`while …`

• Comments	`# Text after the # is ignored.`

Boolean Expressions

- **Booleans** are Yes/No values.
  - In Python: `True` and `False`
- `>, <, ==, !=, >=, <=, and, or`
  - Note the the *"double equals"*
- These expressions all return only True or False.
- `3 < 5 # returns True`
  - You can write `3 < 5 == True` – but this is redundant.
- We'll keep practicing over time

Live Coding Demo

- Open Terminal on the Mac
- Type `python3`
  - We are now in the "interpreter" and can type code.
- Python runs each line of code as we type it.
  - After each line, we see a result. This happens *only* in the interpreter.
- It's a very useful calculator.
- We can also run files!
- `python3 -i 02-Functions.py`
  - `-i :` This means open the interpreter after running the file. It's optional
- `python3 ok …`
  - This runs the file "ok" which is included with each lab / homework.

UC Berkeley EECS
Lecturer
Michael Ball

Computational Structures in Data Science

Python: Control Flow

Conditional Statement

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

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

- Example:

```
if (temperature>37.2):
       print("fever!")
else:
       print("no fever")
```

Live Coding Demo

UC Berkeley EECS
Lecturer
Michael Ball

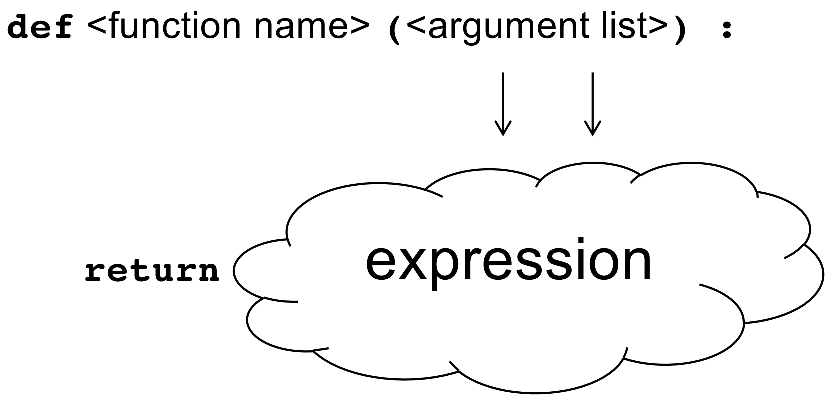
Computational Structures in Data Science

Python: Function Definitions

Learning Objectives

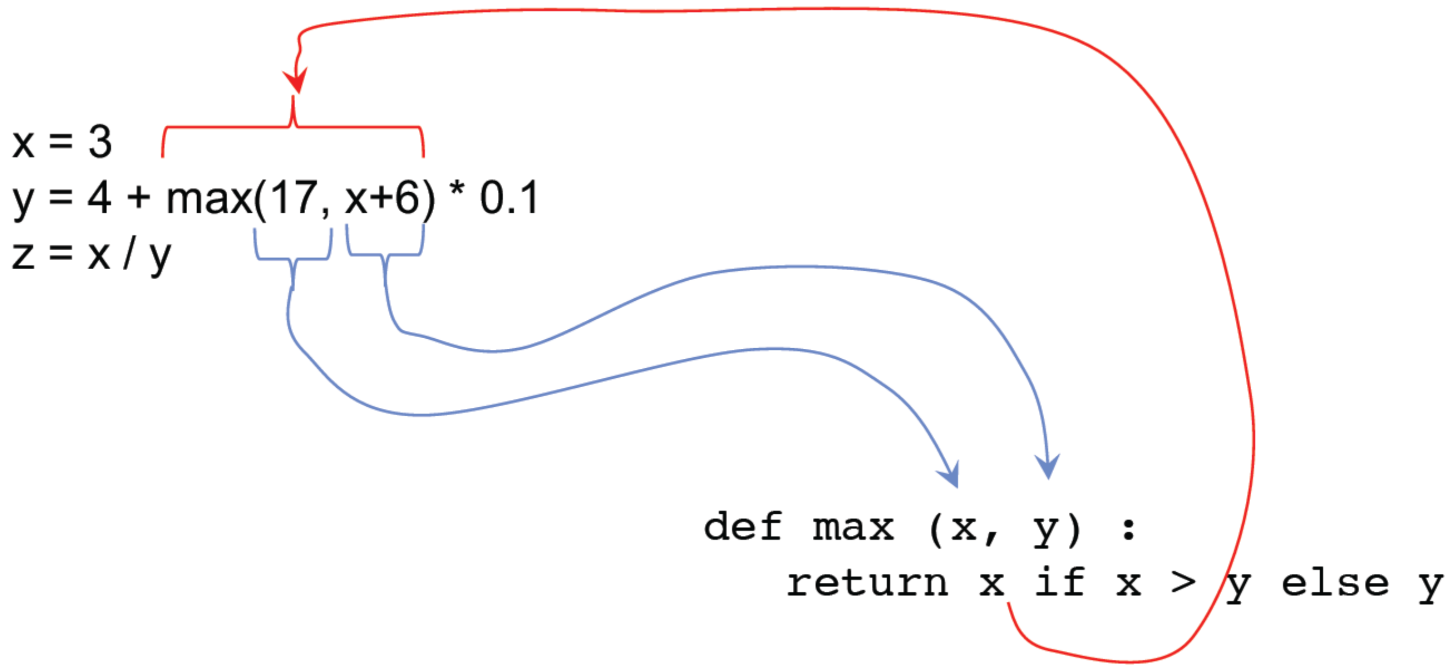
- Create your own functions.
- Use if and else to control the flow of code.

Defining Functions



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

Functions: Example



```
x = 3
y = 4 + max(17, x+6) * 0.1
z = x / y
```

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

Functions: Example

How to Write a Good Function

- Give a descriptive name
  - Function names should be lowercase. If necessary, separate words by underscores to improve readability. Names are extremely suggestive!
- Chose meaningful parameter names
  - Again, names are extremely suggestive.
- Write the docstring to explain *what* it does
  - What does the function return? What are corner cases for parameters?

Python Style Guide "PEP 8"

- Write *doctest* to show what it should do
  - Before you write the implementation.

Live Coding Demo

UC Berkeley EECS
Lecturer
Michael Ball

Computational Structures in Data Science

Functions and Environments

Functions: Calling and Returning Results

Python Tutor

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

x = 3
y = 4 + max(17, x + 6) * 0.1
z = x / y
```

UC Berkeley EECS
Lecturer
Michael Ball

Computational Structures in Data Science

Iteration With While Loops

Learning Objectives

- Write functions that call functions
- Learn How to use while loops.

`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>
```