#1 Control and Environments

TA: Jerry Chen (jerry.c@berkeley.edu)

Roses are red Violets are blue While x < 61 Increment it by 2

Hello

If you're new, welcome!

- TA: Jerry Chen (jerry.c@berkeley.edu)
- Where to find resources
- Reminder on policies

NO ELECTRONICS ALLOWED IN CLASS?



Administrivia

- Labs and Projs are graded on completeness
- Homework is graded on effort
- Attendance is tracked for discussion, but not lab

Spring 2018

Dogs or cats?







Dogs win! (again)

Fall 2017

Dogs or cats?





Spring 2017

Dogs or cats?





Fall 2016

Dogs or cats?





If you weren't here for disc00 v.gd/disc00

Shameless plug

Learn how to use Git

http://tinyurl.com/csmgit (Completely optional)

- Where: HP Auditorium in Soda
- When: Saturday 1pm-3pm (you can come late/leave early)
- Why: Industry standard for code backup and collab

Control

The missing link for all of computer science*

* Statement not endorsed by CS 61A staff pls don't get me in trouble

Computers aren't so complicated



Storage





•

×

+



Booleans

- Basic operators
- Not negates
- And is truthy if both truthy
- **Or** is truthy if either is truthy

Booleans

Short circuiting

False and 1 / 0

Booleans

Short circuiting

False and 1 / 0 True or 1 / 0

if statement

```
1 if n > 10:
 print("Big")
2
3 elif n > 5:
 print("Medium")
4
5
 • • •
6 elif n > 0:
 print("Small")
7
8 else:
     print("Nothing")
9
```

if statement



```
1 if n > 10:
     print("Big")
2
3 if n > 5:
      print("Medium")
4
5
  • • •
 elif n > 0:
6
      print("Small")
7
 else:
8
      print("Nothing")
9
```

2

FizzBuzz

Warmup problem

For every number from 1 to n:

- For multiples of three, print "Fizz"
- For multiples of five, print "Buzz"
- For multiples of both three and five print "FizzBuzz" only
- Otherwise, print the number

Fizzbuzz

```
1 def fizzbuzz(n):
     i = 1
2
       while i <= n:</pre>
3
            if i % 3 == 0 and i % 5 == 0:
 4
5
                print('FizzBuzz')
           if i % 3 == 0:
 6
                print('Fizz')
 7
           if i % 5 == 0:
8
                print('Buzz')
9
            else:
10
                print(i)
11
           i += 1
12
```

Fizzbuzz

```
1 def fizzbuzz(n):
    i = 1
 2
       while i <= n:</pre>
3
            if i % 3 == 0 and i % 5 == 0:
 4
                print('FizzBuzz')
 5
           elif i % 3 == 0:
 6
                print('Fizz')
 7
           elif i % 5 == 0:
 8
                print('Buzz')
 9
            else:
10
                print(i)
11
            <u>i += 1</u>
12
```



Site Reliability Engineer Interview

Anonymous Interview Candidate



Positive Experience

Difficult Interview

Application

I applied through a recruiter. The process took 2 weeks. I interviewed at LinkedIn.

Interview

They reached out to me on LinkedIn and I was selected for a phone screen. After passing the phone screen they set you up to interview with two people that currently have your job. If you pass the first interview, which is more code based, they move you to the 2nd interview which is more systems based. I didn't get the job but learned a lot about how a good...

Show More

Interview Questions

Asked about API calls and parsing data returned from them. Fizzbuzz.

→ Answer Question



They reached out to me on LinkedIn and I was selected for a p passing the phone screen they set you up to interview with tw have your job. If you pass the first interview, which is more co to the 2nd interview which is more systems based. I didn't get about how a good...

Show More

Interview Questions

Asked about API calls and parsing data returned from them. Fizzbuzz.

→ Answer Question

They reached out to me on LinkedIn and I was selected for a p passing the phone screen they set you up to interview with tw have your job. If you pass the first interview, which is more co to the 2nd interview which is more systems based. I didn't get about how a good...

Show More

Interview Questions Asked about API calls and parsing data returned from them. Fizzbuzz.

→ Answer Question

Higher Order Functions

Higher Order Functions

Pass around functions like any other variable

- Function argument
- Return value

More possibilities for abstraction

Higher Order Functions Example from Lab 01

- 1 def compose_twice(f, x):
- 2 return f(f(x))
- 3
- 4 def square(x):
- 5 return x * x
- 6
- 7 compose_twice(square, 2)