

CS 150: More Functions & Booleans

Cynthia Taylor

Oberlin College

February 21st 2014

A brief guide to string manip

- Building a string from letters:

```
s = ""
```

```
if x < 3:
```

```
    s = s + "A"
```

s = "hey"

s = "heyA"

A brief guide to string manip

- Accessing a specific letter in a string

#To set `y` to the 3rd letter in `s`

`y = s[3]`

s = "cs is awesome"
0 1 2 3
y = "i"

- To check if two letters are equal

`s[3] == q[4]`

True

s = "cats"
0 1 2 3
q = "dogss"
0 1 2 3 4

What will the output be?

```
def calculate(w, x, y):
```

```
    a=x
```

```
    b=w+1
```

```
    return a + b - y
```

$\left\{ \begin{array}{l} w = 1 \\ x = 2 \\ y = 3 \end{array} \right.$

main

$\left\{ \begin{array}{l} w = 1 \\ x = 2 \\ y = 3 \end{array} \right.$

calc

calc

$\left\{ \begin{array}{l} w = 2 \\ x = 3 \\ y = 1 \end{array} \right.$

$\left\{ \begin{array}{l} a = 3 \\ b = 3 \end{array} \right.$

```
print(calculate(x, y, w))
```

1

A

3

B

5

C

7

D

E. I don't know

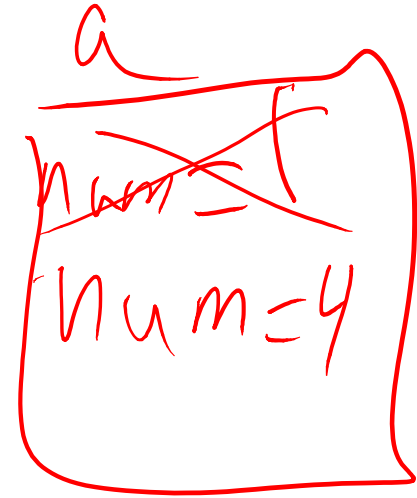
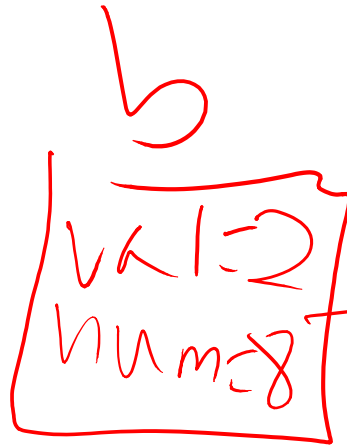
What will the output be?

```
def a(num):  
    num = 4  
    return 2
```

```
def b(val):  
    num = 8  
    print(a(1))
```

b(2)

b(43)



1

A

2

B

4

C

Error because of an undefined variable

D

E. I don't know

What are True and False?

- We know that True is true and False is false
- The integer 0, the float 0.0, the empty string, the None object, and other empty data structures are considered False; everything else is True

```
if 0:  
    print("hi")  
if 4:  
    print("bye")
```

//bye"

Short Circuit Evaluation

- When an expression contains and or or, Python evaluates from left to right
 - It stops evaluating once the truth value of the expression is known
 - For x and y, if x is False, there is no reason to evaluate y
 - For x or y, if x is True, there is no reason to evaluate y
 - Value of the expression is the last thing evaluated
-

Examples

0 and 3

3 and 0

3 and 5

1 or 0

0 or 1

True or 1 / 0

Which parts will be evaluated?

True
(7 > 2) or (*9 < 2*) and (*8 > 3*)

if (False) and False

- ☒ A. Only the **red** code
- B. Only the **green** code
- C. The **red** code and the **green** code
- D. All of the code is evaluated
- E. I don't know

Which parts will be evaluated?

$((7 > 2) \text{ and } (9 < 2)) \text{ or } (8 > 3)$

- A. Only the **red** code
- B. Only the **green** code
- C. The **red** code and the **green** code
- ☒ D. All of the code is evaluated
- E. I don't know

```
def is_odd(x):  
    if x%2 == 1:  
        → return True  
    else:  
        → return False
```

Which of the following does exactly the same thing?

$\text{is_odd}(2) = \text{False}$
 $\text{is_odd}(3) = \text{True}$

A. `def is_odd(x):`

$\begin{matrix} 2-0 \\ 3-1 \end{matrix}$ `return x%2` *numbe*

B. `def is_odd(x):`

$\begin{matrix} 2-F \\ 3-T \end{matrix}$ `return x%2 == 1` *==*

C. `def is_odd(x):` *even*

$\begin{matrix} 2-T \\ 3 \end{matrix}$ `return x%2 == 0`

D. None of the above

E. I don't know

Tips for Writing Code with Booleans

- If you're unsure of precedence, use parentheses
- Use the simplest expression possible
 - Avoid double negatives like not not a
 - Avoid == True and == False

$x < 3 == \text{True} \sim x < 3$
 $x < 3 == \text{False}$ $x > 3$
 $\text{not } x < 3$

```
age = int(input("Enter your age: "))  
if age < 18:  
    print("Youngin")  
elif age >= 18 and age < 35:  
    print("Grown up")  
elif age >= 35: else:  
    print("Older than Cynthia")  
else:  
    print("ageless")
```

What code can
be removed
without changing
the meaning?

- A. The red can be removed
- B. The blue can be removed
- ☒ C. Both red and blue can be removed
- D. Nothing can be removed
- E. I don't know

Next Class

- Types, Exceptions and Strings
- Reading
 - 5.1-5.2,5.5
- Lab 3 – Due Tuesday at 10pm
- Prelab 4 – Due Wednesday in class