



CSCI-UA-0002

Intro to Computer Programming (No Prior Experience)

Module 3: Boolean Data, Conditionals, Modules

Professor Emily Zhao

Section 008

T/R 12:30-1:45PM

Section 012

T/R 4:55-6:10PM



Agenda

- Review Ed Questions
- Module 3 Review
- Practice Problems

Module 3

- Boolean Expressions
- Comparison Operators
- Logical Operators
- One-way if statements
- Two-way if statements
- Multi-way if statements *
- Nested if statements
- Basic Python Modules
- Color in Turtle Graphics

Your Questions

How does `elif` work?

- When are `elif` and `else` skipped?
- What happens if the conditions for two `elif` conditions are valid?
- Do I have to use `elif` and `else`?

When do I need to import a module?

- How do I discover other modules?
- What is dot syntax?

Boolean Data

What are the two possible values a Boolean can have?

True

False

What's the difference between `=` and `==`?

`=` → assign values to variables

`==` → test to see if two values are equal

```
1 isSaturday = True # assign variable
2
3 if isSaturday == True: # test equality
4     print("You don't have class!")
```

You don't have class!

Comparison Operators

<code>x == y</code>	<code># equality</code>
<code>x != y</code>	<code># inequality</code>
<code>x > y</code>	<code># greater than</code>
<code>x >= y</code>	<code># greater than or equal to</code>
<code>x < y</code>	<code># less than</code>
<code>x <= y</code>	<code># less than or equal to</code>

Boolean Expression

an expression used in programming languages that produces a Boolean value (True or False) when evaluated

Logical Operators

and

or

not

What's the output?

`True and False` → `False`

`True or False` → `True`

`not True` → `False`

Conditional Statements

"if" keyword begins a selection statement

condition to be tested

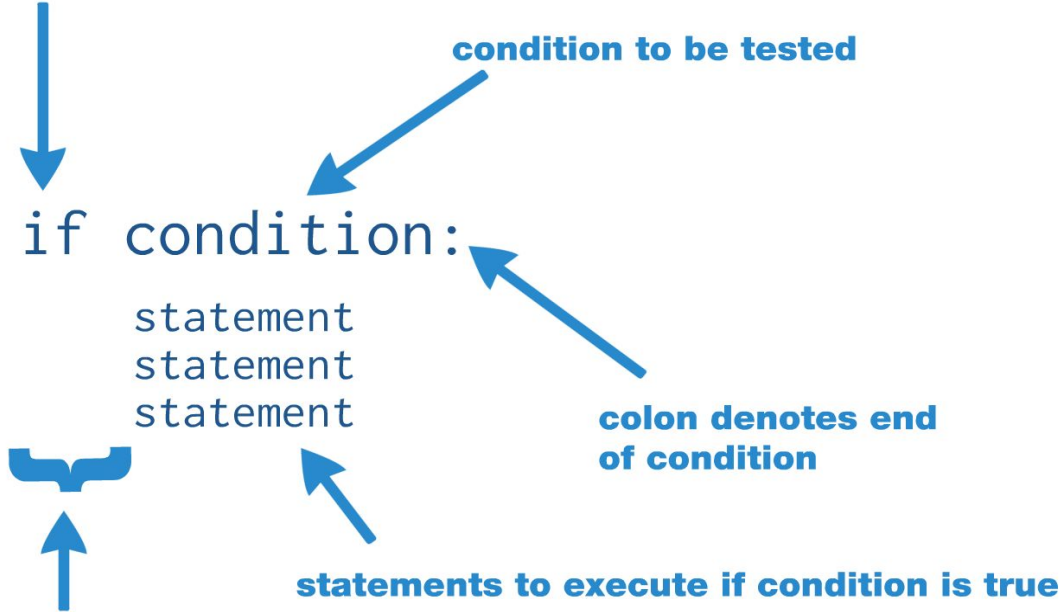
`if condition:`

`statement
statement
statement`

**colon denotes end
of condition**

statements to execute if condition is true

"block" of execution must be indented



Conditionals Practice

pollev.com/emilyzhao



Trace the Output [1]

→ One

→ Three

```
1      a = 5
2      b = 10
3
4      if a < b:
5          print( "one")
6      if a > b:
7          print( "two")
8      if a*2 == b:
9          print ("three")
10     if b < a:
11         print("four")
```


Trace the Output [2]

- One
- Three
- Five

```
1      a = 5
2      b = 10
3
4      if a < b:
5          print("one")
6      if a > b:
7          print ("two")
8      if a*2 == b:
9          print("three")
10     if b < a:
11         print("four")
12     else:
13         print("five")
```

Trace the Output [3]

→ One

```
1      a = 5
2      b = 10
3
4      if a < b:
5          print("one")
6      elif a > b:
7          print ("two")
8      elif a*2 == b:
9          print("three")
10     elif b < a:
11         print("four")
12     else:
13         print("five")
--
```

Programming Challenge: Grade Determination Program

- Input: ask the user for a numeric grade (i.e. 95)
- Process: convert the grade to its letter format (A through F)
- Output: print the letter grade

1. Try to use only `ifs`
2. Try to use only ONE `if`

A	90 – 100
B	80 – 89
C	70 – 79
D	60 – 69
F	59 and below

Programming Challenge: Grade Determination Program

```
grade = float(input("Enter a grade: "))

if 90 <= grade <= 100:
    print("A")
if 80 <= grade < 90:
    print("B")
if 70 <= grade < 80:
    print("C")
if 60 <= grade < 70:
    print("D")
if 0 <= grade < 60:
    print("F")
```

```
grade = float(input("Enter a grade: "))

if grade >= 90:
    print("A")
elif grade >= 80:
    print("B")
elif grade >= 70:
    print("C")
elif grade >= 60:
    print("D")
else:
    print("F")
```

Help me debug my programs!

Programming Challenge: Guessing Numbers

- Write a program where if a user guesses a number divisible by 7 or the secret number 13, they win!
- Otherwise, any other number they guess results in a loss.

Programming Challenge: Guessing Numbers [Broken]

```
1 # Guessing Numbers
2
3 secret = 13
4
5 guess = int(input("Guess a number: "))
6
7 # how do I check if a number is divisible by 7? modulo!
8 if guess % 7 == 0 or secret:
9     print("You win!")
10 else:
11     print("You lose!")
```

Programming Challenge: Guessing Numbers [Solution]

```
1 # Guessing Numbers
2
3 secret = 13
4
5 guess = int(input("Guess a number: "))
6
7 # how do I check if a number is divisible by 7? modulo!
8 if guess % 7 == 0 or guess == secret:
9     print("You win!")
10 else:
11     print("You lose!")
```


Programming Challenge: Am I Your Man?

- You are Lou Bega, artist behind hit pop song Mambo No 5.
- You only want a little bit of Angela, Pamela, Sandra, Rita, Monica, Erica, Tina, Sandra, Mary, or Jessica. No one else.
- Ask the user for a name and if does not match the name of the above women, print “I am not your man.”
- Otherwise, print “You make me your man.”
- *Feel free to just use a couple of the names*

[Chorus]

A little bit of Monica in my life
A little bit of Erica by my side
A little bit of Rita's all I need
A little bit of Tina's what I see
A little bit of Sandra in the sun
A little bit of Mary all night long
A little bit of Jessica, here I am
A little bit of you makes me your
man (Ha!)

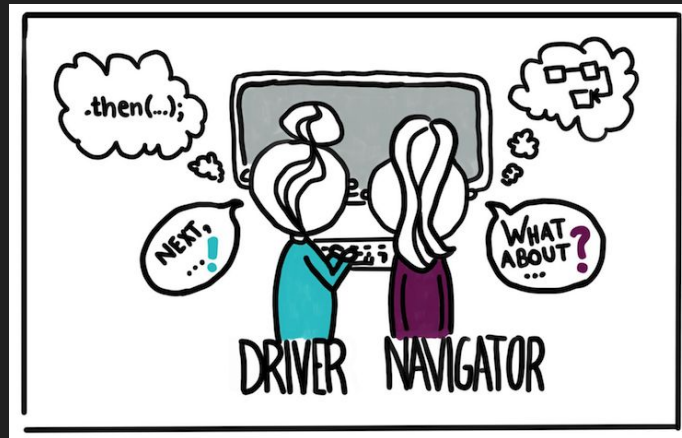
Programming Challenge: Mambo No 5 [Broken]

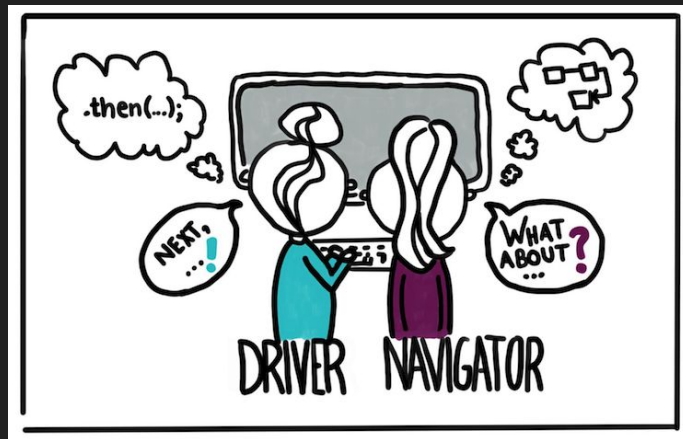
```
1 # Mambo no 5
2
3 name = input("Enter a woman's name: ")
4 if name != "Angela" or name != "Pamela":
5     print("I'm not your man!")
6 else:
7     print("I'm your man!")
```

Programming Challenge: Mambo No 5 [Solution]

```
name = input("Enter a woman's name:")
# user inputs Angela
# != Angela -> False
# != Pamela -> True
if name != "Angela" and name != "Pamela":
    print("I'm not your man")
else:
    print("You make me your man")
```

Pair Programming





The **Driver** is the person at the wheel, i.e. the keyboard.

- focused on completing the tiny goal at hand, ignoring larger issues for the moment.
- A driver should always talk through what she is doing while doing it.

The **Navigator** is in the observer position, while the driver is typing.

- reviews the code on-the-go, gives directions and shares thoughts.
- The navigator also has an eye on the larger issues, bugs, and makes notes of potential next steps or obstacles.

Pair Programming: Calculating a Bonus

- All sales people should receive 1% commission on their sales
- If a sales person made over 10,000 they should receive a \$500 bonus
- If a sales person made over 50,000, they should receive 5% commission on their sales (instead of 1%) – this is in addition to their \$500 bonus for making their quota
- Print out their total take-home amount (bonus + commission) at the end of the program

Programming Challenge: Calculating a Bonus

```
# user input
sales = float(input("Enter monthly sales: "))

# what are the variables?
# initializing my variables / setting "defaults"
percent = 0.01
bonus = 0

# create conditions
if sales >= 10000:
    bonus = 500 # updated the bonus

if sales >= 50000:
    percent = 0.05 # updated the percent

# calculate commission
commission = sales * percent

# output
print("Your take home value: $" + str(sales + commission + bonus))
```

String Comparison

- So far we have been writing Boolean expressions that evaluate based on numeric data
- We can also construct Boolean expressions that can test relationships between strings
- When we compare strings we are essentially reducing them to their zeros and ones and comparing them numerically

Standard ASCII Table

0	<u>NUL</u>	16	<u>DLE</u>	32	<u>SP</u>	48	0	64	@	80	P	96	`	112	p
1	<u>SOH</u>	17	<u>DC1</u>	33	!	49	1	65	A	81	Q	97	a	113	q
2	<u>STX</u>	18	<u>DC2</u>	34	"	50	2	66	B	82	R	98	b	114	r
3	<u>ETX</u>	19	<u>DC3</u>	35	#	51	3	67	C	83	S	99	c	115	s
4	<u>EOT</u>	20	<u>DC4</u>	36	\$	52	4	68	D	84	T	100	d	116	t
5	<u>ENQ</u>	21	<u>NAK</u>	37	%	53	5	69	E	85	U	101	e	117	u
6	<u>ACK</u>	22	<u>SYN</u>	38	&	54	6	70	F	86	V	102	f	118	v
7	<u>BEL</u>	23	<u>ETB</u>	39	'	55	7	71	G	87	W	103	g	119	w
8	<u>BS</u>	24	<u>CAN</u>	40	(56	8	72	H	88	X	104	h	120	x
9	<u>HT</u>	25	<u>EM</u>	41)	57	9	73	I	89	Y	105	i	121	y
10	<u>LF</u>	26	<u>SUB</u>	42	*	58	:	74	J	90	Z	106	j	122	z
11	<u>VT</u>	27	<u>ESC</u>	43	+	59	;	75	K	91	[107	k	123	{
12	<u>FF</u>	28	<u>FS</u>	44	,	60	<	76	L	92	\	108	l	124	
13	<u>CR</u>	29	<u>GS</u>	45	-	61	=	77	M	93]	109	m	125	}
14	<u>SO</u>	30	<u>RS</u>	46	.	62	>	78	N	94	^	110	n	126	~
15	<u>SI</u>	31	<u>US</u>	47	/	63	?	79	O	95	_	111	o	127	<u>DEL</u>

What's the output?

`"dog" > "cat"` → True

`"Camel" < "camel"` → True

`"dog" < "dogfight"` → True



Programming Challenge: Alphabetization Program

- Take in three names as inputs
- Return the three names in alphabetical order.

Programming Challenge: Alphabetization Program [Solution]

```
name1 = input("Enter a name: ")
name2 = input("Enter a name: ")
name3 = input("Enter a name: ")

# case where name1 is first
if name1 < name2 and name1 < name3:
    if name2 < name3:
        print(name1, name2, name3)
    else:
        print(name1, name3, name2)
#case where name2 is first
elif name2 < name1 and name2 < name3:
    if name1 < name3:
        print(name2, name1, name3)
    else:
        print(name2, name3, name1)
#case where name3 HAS TO BE first
else:
    if name2 < name1:
        print(name3, name2, name1)
    else:
        print(name3, name1, name2)
```

Programming Challenge: Password Protection

- Write a program that asks the user for a password
- Check to see if the password that was submitted is equal to the string "secret"
- If it is, print out a “welcome” message
- Otherwise, tell them to try again

Programming Challenge: Password Protection [Solution]

```
# ask user for password
password = input("Enter a password: ")

# check to see if password matches
if password == "secret":
    print("Welcome!")
else:
    print("Try again.")
```

String Manipulation

```
str.lower()
```

```
str.upper()
```

Dot Syntax

```
moduleName.functionName()
```

Used to access functions, attributes, and methods defined within modules and classes

String Manipulation

`str.lower("HELLO")` → `hello`

`"hello".upper()` → `HELLO`



Programming Challenge: Password Protection Extension

- Rewrite your password protection program to be case **insensitive** (i.e. the password "Secret" or "sEcReT" would work)

Programming Challenge: Password Protection Extension

```
# ask user for password
password = input("Enter a password: ")

# check to see if password matches, regardless of case
if password.lower() == "secret":
    print("Welcome!")
else:
    print("Try again.")
```

Modules

What are some modules that you know?

What are some modules that you know?

math

random

turtle

How do I generate a random integer between 1 and 10?

Random integer between 1 and 10

1. Import the module

```
import random
```

2. Call the function using “dot syntax”

```
num = random.randint(1, 10)
```


Programming Challenge: Drawing Shapes

- Ask the user how many sides they would like their shape to be (up to 5) or specify random if they don't care. (Assume they won't type an integer greater than 5)
- Then ask if they would like it drawn or named
 - If they want it drawn, draw it using turtle
 - If they want it named, tell them what it is (triangle, square, pentagon, etc...)

Programming Challenge: Half Birthday Month

```
1      # Half Birthday Month
2      month = int(input("Enter your birth month (i.e. 10): "))
3
4      if month > 6:
5          print("Your half birthday month is: ", month - 6)
6      else:
7          print("Your half birthday month is: ", month + 6)
8
9
10     # Solution without conditionals:
11     # print("Your half birthday month is: ", (month + 5) % 12 + 1)
```

Homework

- Assignment #2 (due Thurs @ 11:59PM)
- Self-Paced Learning Module #4 (due next Tues)
- Quiz #4 (due next Tues)