

CSE 8A: Intro to Programming in Python

Spring 2021

Lecture 5 - conditionals, strings

UC San Diego

Announcement

- PAI due tomorrow
- Go to discussion this week (location on course schedule now)
- Lab 2 on Thursday
 - No make ups for labs unless it is due to emergency
 - Pair programming during lab activities.

Topics for Today

- Conditional statements
- strings processing

Comparisons

`==, !=, >, >=, <, <=`

`num = 12`

`num >= 12`

- A) True
- B) False
- C) None
- D) Error



Join at
slido.com
#29323

Operator Precedence - Boolean Operators

Order	Operator	Operator Name
1	not	Boolean NOT
2	and	Boolean AND
3	or	Boolean OR

Exercise: Booleans

What is the value of the expression below?

```
num = 12
```

```
(num != 12) and (num > 0) or (num % 2 == 1)
```

- A) True
- B) False
- C) None
- D) Error



Join at
slido.com
#29323

Exercise: Booleans

What is the value of the expression below?

```
num = 17
```

```
not num == 17 or num >= 17 and num // 2 == 8
```

- A) True
- B) False
- C) None
- D) Error



Join at
slido.com
#29323

Exercise: If/Else

What will be printed by the program below?

```
def mystery(n):  
    if n < 0:  
        return n * -1  
    else:  
        return n  
  
print(mystery(-42))
```

- A) -42
- B) 42
- C) The program will not print anything
- D) The program will result in an error



Join at
slido.com
#29323

Exercise: Nested Ifs

What will be printed?

```
num = 8
```

```
if num % 2 == 0:
```

```
    if num % 3 == 0:
```

```
        print('divisible by 2 and 3')
```

```
    else:
```

```
        print('divisible by 2 but not 3')
```

```
else:
```

```
    if num % 3 == 0:
```

```
        print('divisible by 3 but not 2')
```

```
    else:
```

```
        print('not divisible by 2 and 3')
```

A) divisible by 2 and 3

B) divisible by 2 but not 3

C) divisible by 3 but not 2

D) not divisible by 2 and 3



Join at
slido.com
#29323

Exercise: Nested Ifs

What will be printed?

```
num = 15
```

```
if num % 2 == 0:
```

```
    if num % 3 == 0:
```

```
        print('divisible by 2 and 3')
```

```
    else:
```

```
        print('divisible by 2 but not 3')
```

```
else:
```

```
    if num % 3 == 0:
```

```
        print('divisible by 3 but not 2')
```

```
    else:
```

```
        print('not divisible by 2 and 3')
```

A) divisible by 2 and 3

B) divisible by 2 but not 3

C) divisible by 3 but not 2

D) not divisible by 2 and 3



Join at
slido.com
#29323

Exercise: Else If (elif)

Write a function that takes a number (float) as an input parameter and print

0: if the number is a multiple of 2

1: elif the number is a multiple of 3

-1: elif the number is a multiple of 5

2: else none of the above is satisfied



Join at
slido.com
#29323

What is true for block 1 and 2?

#block 1

```
if condition A:  
    #statements A
```

```
elif condition B:  
    #statements B
```

```
elif condition C:  
    #statements C
```

```
else:  
    #statements D
```

#block 2

```
if condition A:  
    #statements A
```

```
if condition B:  
    #statements B
```

```
if condition C:  
    #statements C
```

```
else:  
    #statements D
```



Join at
slido.com
#29323

- A. they are basically the same code, no difference
- B. for block 1, it is impossible that statements A and D are both executed
- C. for block 2, it is impossible that statements A and D are both executed
- D. More than one of the answers are correct
- E. None of the answers is correct

Strings in Python

- String is a sequence of characters
- Python treats each character as a string too!
- Don't confuse string variable with string constants
 - 'name' or "name" is a string constant
 - name by itself can be a string variable

indexes for string characters

what is the result of the following operation?

```
name = "cse 8a"  
print(len(name))  
print(name[2])
```

A	B	C	D	E
5	6	6	5	4
s	s	e	e	e



Join at
slido.com
#29323

indexes for string characters

what is the result of the following operation?

```
name = "cse-8a"  
print(name[2:3])
```

- | | | | | |
|----|----|---|---|-------|
| A | B | C | D | E |
| se | e- | e | s | error |

How about negative indexing???



Join at
slido.com
#29323

indexes for string characters

```
name = "cse-8a"
```

+index	0	1	2	3	4	5
letter	c	s	e	-	8	a
-index	-6	-5	-4	-3	-2	-1

[a:b] all the characters from index a to index b (not including index b)

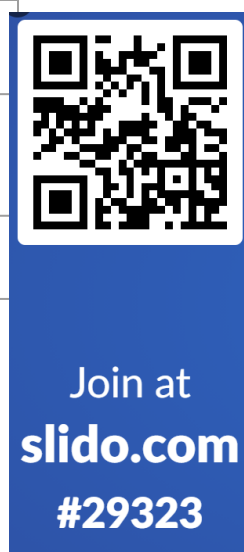
[a:] all the characters from index a to the end of the string (end is the right end of the string)

[:b] all the characters from index 0 to index b (not including index b)



Join at
slido.com
#29323

+index	0	1	2	3	4	5
letter	c	s	e	-	8	a
-index	-6	-5	-4	-3	-2	-1



```
name = "cse-8a"
print(name[2:])
print(name[-2:])
print(name[1:-1])
```

[a:b] all the characters from index a to index b
(not including index b)

[a:] all the characters from index a to the end of
the string (end is the right end of the string)

[:b] all the characters from index 0 to index b
(not including index b)

A

e-8a

8a

(empty str)

B

e-8

8

se-8

C

e-8a

8a

se-8

D

e-8

8a

(empty str)

E

error