

# OUR AGENDA FOR TODAY

Arithmetic Order

Data Types

Expressions

Build Lines of code

Review



# PEMDAS

**Parentheses:** Operations enclosed within parentheses should be performed first. Parentheses are used to group parts of an expression and indicate that the operations within them should be evaluated before anything else.

**Exponents:** After parentheses, the next step is to evaluate any exponents or powers. An exponent refers to the number of times a base number is multiplied by itself.

**Multiplication and Division:** After evaluating parentheses and exponents, multiplication and division operations are performed from left to right. These operations have equal precedence, so they are performed in the order they appear in the expression.

**Addition and Subtraction:** Finally, after completing the multiplication and division operations, addition and subtraction are performed from left to right. Like multiplication and division, these operations have equal precedence and are evaluated in the order they appear in the expression.

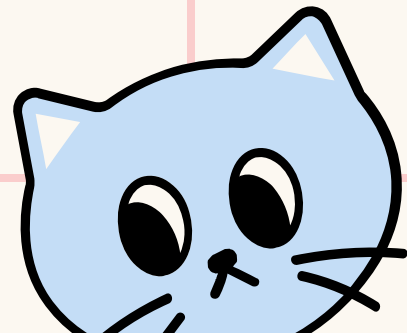
BIAS

## Exercise 1

- Setup a for loop which executes its body with the number variable going through the values -3, -2, -1


Q: How many asterisks ( `*` ) does the code output to the screen?


```
torque = 0
while torque != 0:
    torque //= 2
    print("*", end="")
else:
    print("*")
```

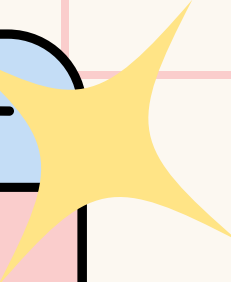



BIAS





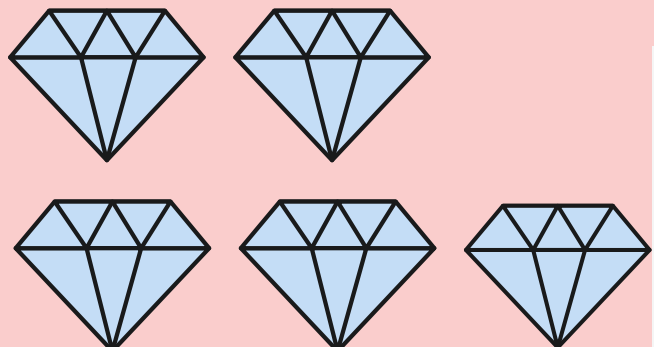






### Exercise 2

Arrange the code boxes in order to form a conditional instruction which guarantees that a certain statement is executed when the `tax_rate` variable is less than 23.5



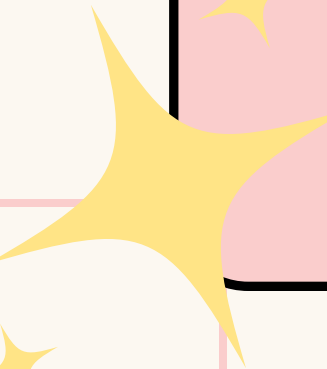
:

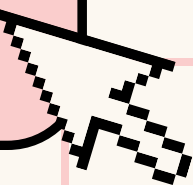
if

tax\_rate

23.5


<







A blue cartoon cat head with large eyes and whiskers.

A pink, fluffy cloud-like shape.



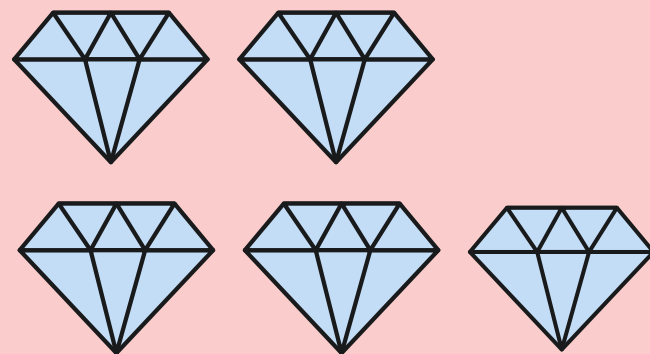




BIAS

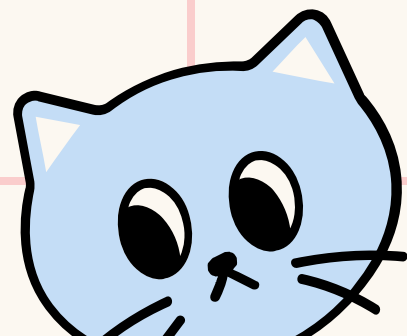


### Exercise 3

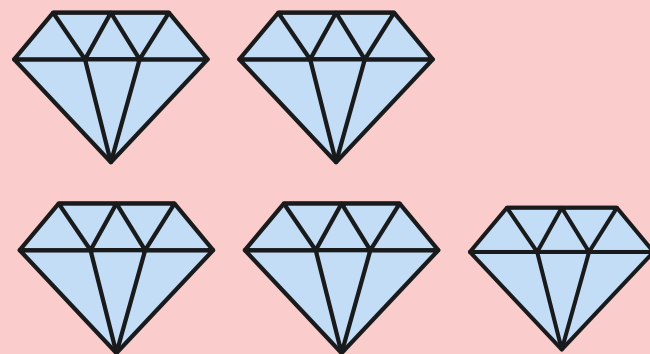


Q: What happens when the user runs the following code?

```
angle = -1
for i in range(-1, 1):
    if 2 * i < 4:
        angle += 1
    else:
        angle += 2
print(angle)
```



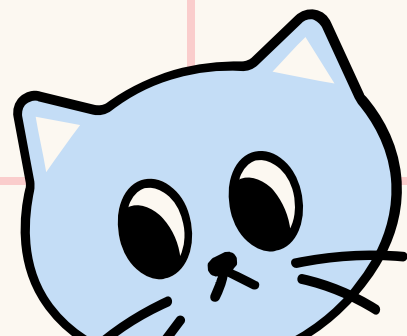
## Exercise 4



Q: What happens when the user runs the following code?

```
power = 2
while power < 5:
    power += 1
    if power == 3:
        continue
    print("@", end="")
else:
    print("@")
```

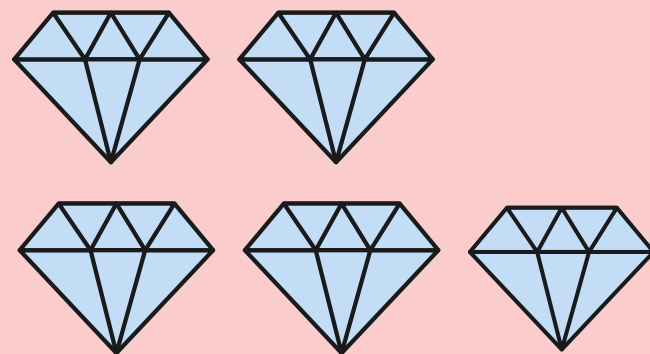
BIAS



BIAS

Drag and drop the conditional expressions to obtain a code which outputs `###` to the screen.  
(Note: some code boxes will not be used.)

### Exercise 5



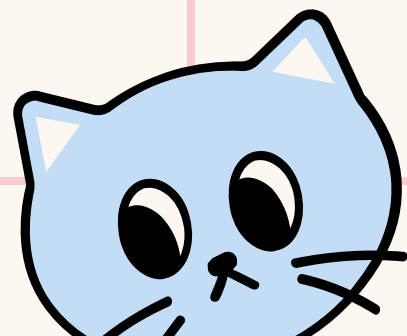
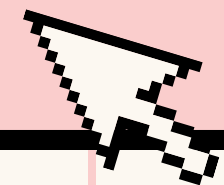
`shift = 0`

`shift != 0`

`shift < 0`

`shift == 0`

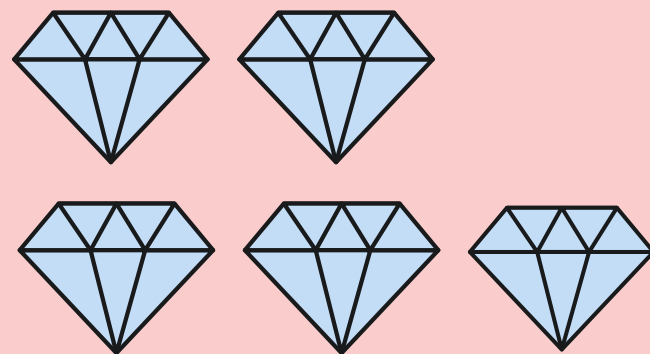
```
shift = 5 - 4 * 2
if shift > 0 :
    print("#")
elif  :
    print("##")
elif  :
    print("###")
```





BIAS

## Exercise 6



Q: What is the expected output of the following code?

```
list_one = [1,2]
list_two = list_one[:]
list_two.append(3)
print(list_one[-1] + list_two[-1])
```

