



Control Flow Statements

- Conditionals
- Loops





Conditional Statements





Table of Contents

- ▶ Structure of the `if` Statements
- ▶ Comparison Operators
- ▶ `if-else` Statements
- ▶ `if-elif-else` Statements
- ▶ Nested `if-elif-else` Statements



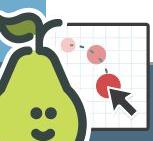
1

Structure of the if Statements

```
if a > b :
```

```
    print('a is greater than b')
```

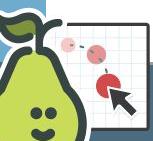
How was the Python course by now? Did you satisfied?



Students, drag the icon!



How was the pre-class and in-class contents? Did you satisfied?

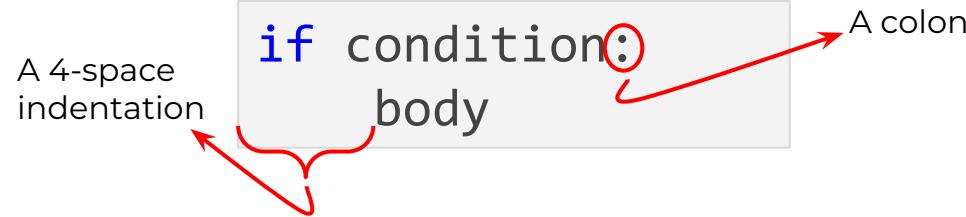


Students, drag the icon!



Structure of the `if` Statements

- The basic structure  of an `if` statement is :



Structure of the **if** Statements

- Here's the simple **pre-class** examples of the **if** Statements :

```
1 if True:  
2     print('it is true')  
3
```

Structure of the **if** Statements

- Here's the simple **pre-class** examples of the **if** Statements :

```
1 if True:  
2     print('it is true')  
3
```

```
1 it is true  
2
```



if Statements

► Task : Cooking a hamburger.

- ▷ We need some ingredients that are not in our kitchen.
- ▷ There is only one **grocery store** in our village and its availability is crucial.
- ▷ Ingredients (stated below) required for cooking hamburgers with **greens** (it does not matter which one. **lettuce / pepper**)
- ▷ Set a logical boolean algorithm onto **hamburger** to be able to eat.
- ▷ Set a condition **hamburger** variable with if statement that gives us a message "Bon Appetit." if we can cook hamburger, do nothing if we can not.

```
#ingredients and requirements:  
minced meat (must)  
hamburger bread (must)  
lettuce } (must)  
pepper  
grocery store (must)
```





if Statements

- ▶ The code can be like:

```
1 minced = True  
2 bread = True  
3 lettuce = False  
4 pepper = True  
5 grocer = True  
6  
7 hamburger = (minced and grocer and bread) and (lettuce or pepper)  
8  
9 if hamburger :  
10     print("Bon Appetit")  
11  
12
```



The values (**True/False**) are up to you

Output

Bon Appetit



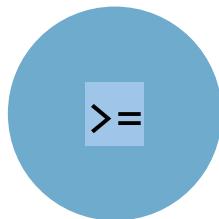
2

Comparison Operators

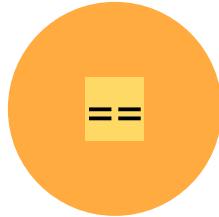
Draw lines to match the image to the answer:



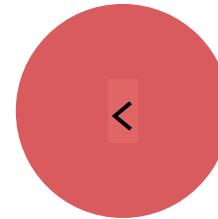
equal



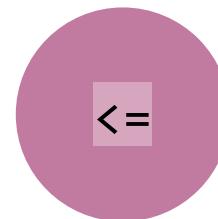
greater than



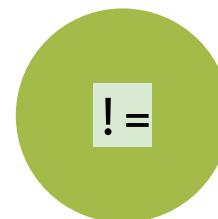
greater than or equal



less than



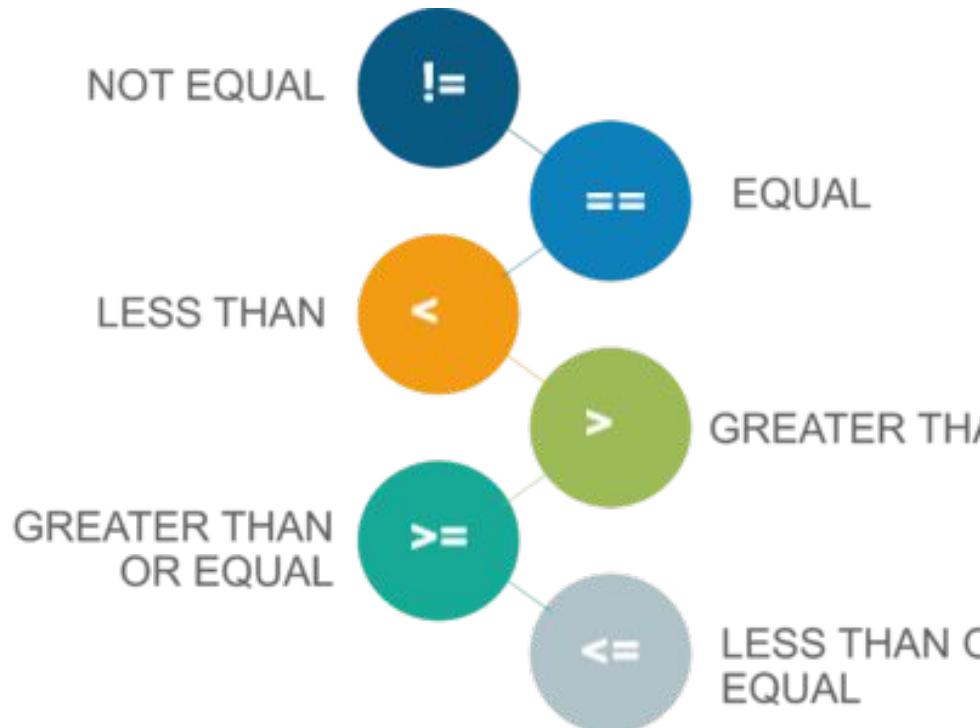
less than or equal



Students, draw anywhere on this slide!



Comparison Operators



These Operators
return True or False



Comparison Operators

- Here's the simple **pre-class** examples of the **if** Statements :

```
1 empty_seat = 14
2
3 if empty_seat > 3: # in this case, 14>3=True, so the body will execute
4     print('there is still seat to sit')
5
```

Comparison Operators

- Here's the simple pre-class examples of the **if** Statements :

```
1 empty_seat = 14
2
3 if empty_seat > 3: # in this case, 14>3=True, so the body will execute
4     print('there is still seat to sit')
5
```

comparison operator



```
1 there is still seat to sit
2
```



Comparison Operators

- ▶ Take a look at the following examples :

```
1 print(1 == 1)
2 print("henry" == "Henry")
3 print(12 < 12.1)
4 print("hard" != "easy")
5
6
```

Output

```
True
False
True
True
```

Comparison Operators (review)



Opr.	How it works ?	Sample
<code>==</code>	Returns <code>True</code> if two values are equal or <code>False</code> if different	<code>2 == 2 (True), 2 == 3 (False)</code>
<code>!=</code>	Returns <code>True</code> if two values are not equal or <code>False</code> if equal	<code>2 != 2 (False), 2 != 3 (True)</code>
<code>></code>	Returns <code>True</code> if the value on the left is greater than the value on the right otherwise returns <code>False</code>	<code>3 > 2 (True), 2 > 3 (False)</code>
<code><</code>	Returns <code>True</code> if the value on the left is less than the value on the right otherwise returns <code>False</code>	<code>2 < 3 (True), 3 < 2 (False)</code>
<code>>=</code>	Returns <code>True</code> if the value on the left is greater than or equal to the value on the right otherwise returns <code>False</code>	<code>3 >= 2 (True), 3 >= 3 (True), 2 >= 3 (False)</code>
<code><=</code>	Returns <code>True</code> if the value on the left is less than or equal to the value on the right otherwise returns <code>False</code>	<code>3 <= 2 (False), 3 <= 3 (True), 2 <= 3 (True)</code>

Comparison Operators

- ▶ Let's examine the following **pre-class** example carefully :

```
1 x = 6
2 y = 9
3 print ("is x equal to y?      :" , x == y)
4 print ("is x not equal to y?  :" , x != y)
5 print ("is x less than y?     :" , x < y)
6 print ("is x greater than y?   :" , x > y)
7 print ("is x less than or equal to y? :" , x <= y)
8 print ("is x greater than or equal to y? :" , x >= y)
9
```



Comparison Operators

- ▶ Let's examine the following example carefully :

```
1 x = 6
2 y = 9
3 print ("is x equal to y?      :" , x == y)
4 print ("is x not equal to y?   :" , x != y)
5 print ("is x less than y?       :" , x < y)
6 print ("is x greater than y?     :" , x > y)
7 print ("is x less than or equal to y? :" , x <= y)
8 print ("is x greater than or equal to y? :" , x >= y)
9
```

```
1 is x equal to y?      : False
2 is x not equal to y?   : True
3 is x less than y??    : True
4 is x greater than y?   : False
5 is x less than or equal to y? : True
6 is x greater than or equal to y? : False
7
```



Comparison Operators

▶ Task :

- ▷ Create two sets (using `set()` function) with the given string values below.
- ▷ Compare these sets and print out 'We are the same!' if they are equal, do nothing if they are not.

- "TWELVE PLUS ONE"
- "ELEVEN PLUS TWO"



Comparison Operators

- ▶ The code might be like :

```
1 set1 = set("TWELVE PLUS ONE")
2 set2 = set("ELEVEN PLUS TWO")
3
4 if set1 == set2:
5     print("We are the same!")
6
7
```

Discuss in-class! How they can be the same.

Output

We are the same!

Comparison Operators

- ▶ **Task : Convert string "Yes" to boolean True, convert string "No" to boolean False.**
 - ▷ Write a program that ;
 - ▷ Takes the word **Yes** or **No** from the users and **converts** it into **boolean** type.
 - ▷ Yes → **True**
 - ▷ No (or other than Yes) → **False**
 - ▷ Print the result i.e.(if yes): "You entered True"
 - ▷ You may not use if-statements



Comparison Operators

- ▶ The code might be like :

```
1 convert = input("Enter Yes or No : ").title().strip() == "Yes"
2 print("You entered", convert)
3
```



3

if-else Statements

if-else Statements(review)

- ▶ The simple structure  of an **if-else** statement is :

```
if condition1:  
    execute body1  
else:  
    execute body2
```

if-else Statements(review)

- Let's take a look at this **pre-class** example of an **if-else** statement :

```
1 course = 'clarusway'  
2  
3 if course == "clarusway":  
4     print("you guaranteed the job")  
5 else:  
6     print("think about it again")  
7
```

if-else Statements

- Let's take a look at this example of an **if-else** statement :

```
1 course = 'clarusway'  
2  
3 if course == "clarusway":  
4     print("you guaranteed the job")  
5 else:  
6     print("think about it again")  
7
```

```
1 you guaranteed the job  
2
```

if-else Statements(review)

- Here's another **pre-class** example of an **if-else** statement :

```
1 number = 5
2 if number <= 3:
3     print("Number is smaller than or equal to 3")
4 else: # Optional clause (you can only have one else)
5     print("Number is bigger than 3")
6
```

What is the output? Try to figure out in your mind...



if-else Statements

- Here's another example of an **if-else** statement:

```
1 number = 5
2 if number <= 3:
3     print("Number is smaller than or equal to 3")
4 else: # Optional clause (you can only have one else)
5     print("Number is bigger than 3")
6
```

```
1 Number is bigger than 3
2
```

if-else Statements

- ▶ **Task : Python Program to Check if a Number is Odd or Even**
 - ▷ Write a program to check whether a number entered by the user is even or odd.
 - ▷ Print the result such as : “**2 is even**”

if-else Statements

- ▶ The code might be like :

```
1 num = int(input("Enter a number: "))
2 if (num % 2) == 0:
3     print("{0} is Even".format(num))
4 else:
5     print("{0} is Odd".format(num))
6
7 |
```

if-else Statements

- ▶ **Task : Python Program to Check if a Number is Negative or Positive.**
 - ▷ Write a program to check whether a number entered by the user is *negative* or *positive*. Number zero is not acceptable.
 - ▷ Print the result such as : '**Positive number**'



USWY[®]
Students, write your response!

REINVENT YOURSELF

Pear Deck Interactive Slide
Do not remove this bar

if-else Statements

- ▶ The code might be like :

```
1 num = float(input("Enter a number: "))
2 if num > 0:
3     print("Positive number")
4 else:
5     print("Negative number")
6
7 |
```

if-else Statements

- ▶ **Task : Python Program to Check which number is larger.**
 - ▷ Write a program that prints which of the two numbers the user entered is large.
 - ▷ Print the result such as : “The large number is 4”

if-else Statements

- The code might be like :

```
1 num1 = float(input("Enter first number: "))
2 num2 = float(input("Enter second number: "))
3
4 if (num1 > num2) :
5     larger = num1
6 else:
7     larger = num2
8
9 print("The large number is", larger)
10
11
```

Option-1

if-else Statements

- The code might be like :

```
1 num1 = float(input("Enter first number: "))
2 num2 = float(input("Enter second number: "))
3
4 if (num1 > num2) :
5     larger = num1
6 else:
7     larger = num2
8
9 print("The large number is", larger)
10
11
```

Option-1

```
1 num1 = float(input("Enter first number: "))
2 num2 = float(input("Enter second number: "))
3
4 if (num1 > num2) :
5     print("The large number is", num1)
6 else:
7     print("The large number is", num2)
8
```

Option-2

if-else Statements



- ▶ **Task :** Convert boolean True to string value of "Yes", convert boolean False to string value of "No".
 - ▷ Write a program that ;
 - ▷ Converts the type of the variable `bool_value` which keeps `True / False` to `Yes` or `No`.
 - ▷ `True → "Yes"`
 - ▷ `False → "No"`

if-else Statements

- ▶ The code might be like :

```
1 | bool_value = False # can be True or False
2 |
3 | if bool_value:
4 |     print("Yes")
5 | else :
6 |     print("No")
7 |
```

Output

No



4

if-elif-else Statements

What do you know about statements:

if-elif-else



Pear Deck

Pear Deck Interactive Slide
Do not remove this bar



Students, write your response!

if-elif-else Statements (review)

- ▶ You can define a series of conditionals.
- `if` for the **first** one,
- `elif` for the **rest**, up until the final (optional),
- `else` for **anything not caught by the other conditionals.**

if-elif-else Statements (review)

- The simple and common structures of an if-elif-else statement are:

```
if condition1:  
    execute body1  
  
elif condition2:  
    execute body2  
  
else:  
    execute body3
```

```
if condition_1:  
    action_1  
  
elif condition_2:  
    action_2  
    .  
    .  
    .  
    .  
elif condition_n:  
    action_n  
  
else:  
    action_(n+1)
```



here you can
add as many
elifs as you need

if-elif-else Statements (review)

- ▶ Consider this pre-class example :

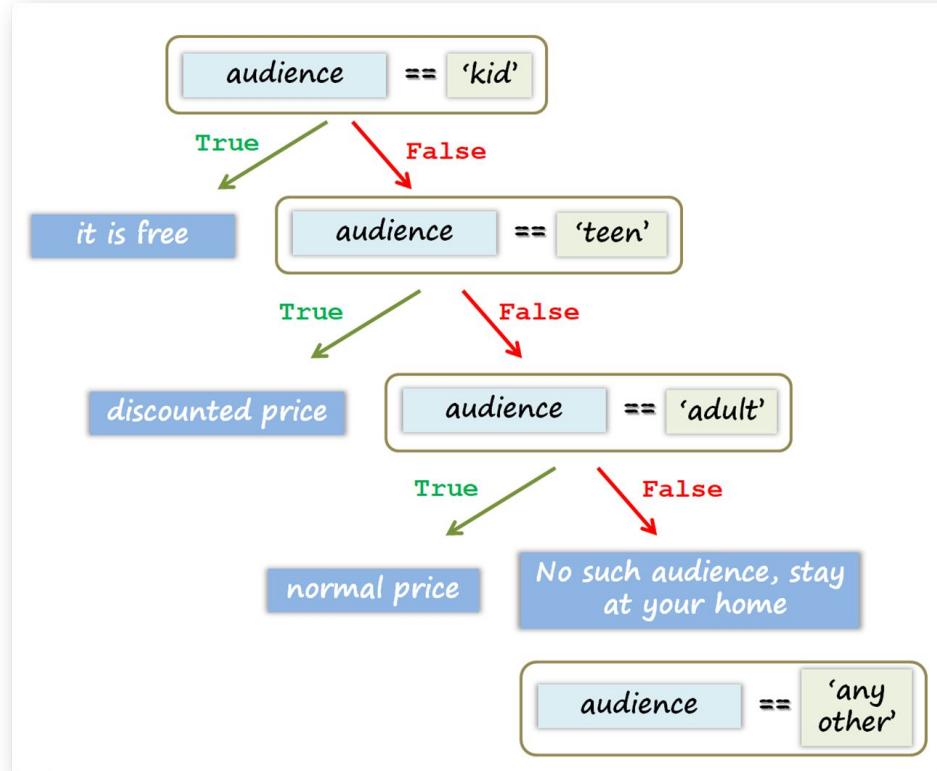
```
1 audience = "baby"
2
3 if audience == "kid":
4     print("it is free to go to cinema")
5 elif audience == "teen":
6     print("discounted price!")
7 elif audience == "adult":
8     print("normal price")
9 else:
10    print("No such audience, stay at your home!")
11
```

What is the output? Try to figure out in your mind...



if-elif-else Statements (review)

- Let's examine this diagram of previous example :



if-elif-else Statements (review)

- The output :

```
1 audience = "baby"
2
3 if audience == "kid":
4     print("it is free to go to cinema")
5 elif audience == "teen":
6     print("discounted price!")
7 elif audience == "adult":
8     print("normal price")
9 else:
10    print("No such audience, stay at your home!")
11
```

```
1 No such audience, stay at your home!
2
```

if-elif-else Statements

- ▶ **Task : Write Python Program to Find the Largest Among Three Numbers**
 - ▷ Write a program that prints which of the three numbers the user entered is the largest.
 - ▷ Print the result such as : “The largest number is 4”

if-elif-else Statements

- The code might be like :

```
1 num1 = float(input("Enter first number: "))
2 num2 = float(input("Enter second number: "))
3 num3 = float(input("Enter third number: "))
4
5 if (num1 >= num2) and (num1 >= num3):
6     largest = num1
7 elif (num2 >= num1) and (num2 >= num3):
8     largest = num2
9 else:
10    largest = num3
11
12 print("The largest number is", largest)
13
```

if-elif-else Statements

- ▶ **Task : Write Python Program to Check if a Number is Negative, Positive or Zero.**
 - ▷ Write a program to check whether a number entered by the user is negative, positive or zero.
 - ▷ Print the result such as : “**Negative number**” or “**Zero**”.



Students, write your response!

REINVENT YOURSELF

Pear Deck Interactive Slide
Do not remove this bar

if-elif-else Statements

- ▶ The code might be like :

```
1 num = float(input("Enter a number: "))
2 if num > 0:
3     print("Positive number")
4 elif num == 0:
5     print("Zero")
6 else:
7     print("Negative number")
8
```



5

Nested if-elif-else Statements

Nested if-elif-else Statements

- ▶ Nested structure of pre-class examples.

```
1 audience_group = 'kid', 'teen', 'adult'
2
3 audience = "teen"
4
5 if audience in audience_group:
6     if audience == "kid":
7         print("it is free to go to cinema")
8     elif audience == "teen":
9         print("discounted price!")
10    else: # audience == "adult":
11        print("normal price")
12 else:
13     print("No such audience, stay at your home!")
```

Nested if-elif-else Statements

- In this case, the output is :

```
1 audience_group = 'kid', 'teen', 'adult'
2
3 audience = "teen"
4
5 if audience in audience_group:
6     if audience == "kid":
7         print("it is free to go to cinema")
8     elif audience == "teen":
9         print("discounted price!")
10    else: # audience == "adult":
11        print("normal price")
12 else:
13     print("No such audience, stay at your home!")
```

```
1 discounted price!
2
```

Nested if-elif-else Statements

- Let's write a program that asks you to enter your exam score and calculates the range in which your degree is based on your exam score. The output would be: e.g, "**Your degree is B+**"
 - 95 and above ►► "A+"
 - 90-94 ►► "A"
 - 85-89 ►► "B+"
 - 80-84 ►► "B"
 - 79 and below ►► "below B" or "B-"



Use nested **if**-statement.

Nested if-elif-else Statements

- The one of the solution code may be like :

```
1 score = int (input("Enter your score :"))
2
3 if score >= 90:
4     if score >= 95:
5         Score_letter="A+"
6     else:
7         Score_letter="A"
8 elif score >= 80:
9     if score >= 85:
10        Score_letter="B+"
11    else:
12        Score_letter="B"
13 else:
14     Score_letter="below B"
15
16 print ("Your degree: %s" % Score_letter)
17
```

THANKS!

End of the Lesson

(Conditional Statements)

next Lesson



Loops



click above

