

Python Conditionals

Textbook

Python Conditionals



A condition compares values using a [comparison operator](#). You've probably seen these before in math class, but we'll go over them again here:

- Greater than: >
- Greater than or equal to: >=
- Less than: <
- Less than or equal to: <=

Like in math, you can check to see if a number is greater than, less than, greater than or equal to, or less than or equal to another number. You do this with the same symbols as math. You can also check if two values are equal or not equal.

Check if Two Values are Equal

IMPORTANT!! If you want to check to see if a piece of data is equal to another piece of data, you can compare them by putting **two equal signs (==)** between them. A common mistake new programmers make is forgetting two equals signs and only using one when checking if two things are equal, so watch out!

```
1 friend_pet = "poodle"
```

```
2 your_pet = "fish"
3
4 if friend_pet == your_pet:
5     print("You and your friend have the same pet!")
6 else:
7     print("You and your friend have different pets.")
8
```

Try it!

Check if Two Values are NOT Equal

If you want to check to see if two values are not equal, combine an exclamation and equals symbol (!=) between the values.

```
1 friend_pet = "poodle"
2 your_pet = "fish"
3
4 if friend_pet != your_pet:
5     print("You and your friend have different pets!")
6 else:
7     print("You and your friend have the same pet!")
```

Try it!

AND Condition

If you want to check if TWO different conditions are both true, you can combine two condition statements by using an [AND condition](#).

You do this by putting the word `and` between conditions.

condition1 `and` condition2

Here is an example:

```
1 your_pet= "gerbil"
2 your_need = "exercise"
3
4 if your_pet == "dog" and your_need == "animal companion":
5     print("You are allowed to bring your pet inside the store.")
6 else:
7     print("You cannot bring your pet inside the store.")
8
```

Try it!

OR Condition

If you want to check if ONE of TWO different conditions are true then you can combine them by using an [OR condition](#).

You do this by putting the word `or` between conditions.

condition1 `or` condition2

Here is an example:

```
1 your_pet= "gerbil"
2
3 if your_pet == "fish" or your_pet == "reptile":
4     print("You are allowed to bring your pet inside the classroom.")
5 else:
6     print("You cannot bring your pet inside the classroom.")
```

Try it!

Selection Structures

In Python, conditional statements, also known as selection structures, are like decision-making tools for your program. They let your code choose different paths to follow based on whether a certain condition is true or false.

For example, you might want your program to do one thing if a user enters a specific number, and something entirely different if they enter another. These "if-then-else" logic structures allow programs to be dynamic and responsive, rather than just running a fixed set of instructions.

They're essential for creating programs that can adapt to different inputs, handle various situations, and respond intelligently to data, making them a fundamental building block for any complex software.

Boolean Logic

In programming, conditions that are either true or false are at the heart of how programs make decisions. This concept is called **Boolean logic**, named after mathematician George Boole. At its simplest, Boolean logic deals with just two values: `True` and `False`. These are special keywords in Python (and many other languages) that represent these two states.

When you use comparison operators (like `==`, `!=`, `<`, `>`), the result of that comparison is always a Boolean value. For example, `5 > 3` evaluates to `True`, while `10 == 7` evaluates to `False`.

Boolean logic also involves **logical operators** like `and`, `or`, and `not` which you've already seen.

- The `and` operator gives `True` only if *both* conditions it connects are `True`. If even one is `False`, the result is `False`.
- The `or` operator gives `True` if *at least one* of the conditions it connects is `True`. It only results in `False` if *both* conditions are `False`.
- The `not` operator simply reverses a Boolean value. If something is `True`, `not` makes it `False`, and vice-versa.

Understanding Boolean logic is fundamental because it's how programs evaluate conditions, control the flow of execution, and make intelligent decisions based on data. It's the "brain" behind `if` statements and other selection structures, allowing your code to react differently to various situations.

The screenshot shows the Skill Struck web application. On the left, a lesson titled "Python Conditionals" is displayed. It explains that a condition compares values using a comparison operator and lists four types: Greater than (>), Greater than or equal to (>=), Less than (<), and Less than or equal to (<=). Below this, a video player shows a progress bar at 0:00 / 4:11 with a "Start Checkpoint" button. The central code editor, titled "Aimee.py", contains the following Python code:

```
1 tickets = 5
2
3 if tickets > 4:
4     print("You can ride the carousel")
5     print("You have enough tickets")
6 else:
7     print("You can't ride")
8
```

On the right, there is a "Run" button and a "Save" button. Below these is a terminal window with a black background and a white prompt ">>>". At the top right of the interface, there are links for "Glossary", a help icon, and a user profile icon.

Checkpoint

Python Conditionals

Create a program to check if a person is able to drive. Compare two variables and print a response depending on how the variables compare.

1. Use an **input** function to ask the user for their `age` .
2. Create a variable named `license` that holds a boolean for if they have their license.
3. Create the IF statement with the following conditions:
4. If age is greater than or equal to 16 **and** they have their license, print out `You are old enough to drive.`
5. Else, print out `You are not able to drive.`

Remember to use two equal signs for the license variable in your IF statement.

Requirements:

- Create the Input.
- Create the IF statement. Make sure to check for age first and use the word and to also check the license boolean.
- Create the else statement.
- Create a variable named `license` that holds a boolean

Questions (8)

1. What is the correct way to see if two values are equal?

MULTIPLE CHOICE

Choose the correct answer:

- A. `value = value`
- B. `value == value`
- C. `value is value?`
- D. `value === value`

2. What is the correct way to see if one condition AND another are met in Python?

MULTIPLE CHOICE

Choose the correct answer:

- A. `condition1 and condition2`
- B. `condition1 & condition2`
- C. `condition1 ^ condition2`
- D. `condition1 && condition2`

3. How do you check to see if two values are NOT equal?

MULTIPLE CHOICE

Choose the correct answer:

- A. `value /= value`
- B. `value x= value`
- C. `value NOT= value`
- D. `value != value`

4. How do you check to see if one condition OR another condition is met?

MULTIPLE CHOICE

Choose the correct answer:

- A. `condition1 ^^ condition2`
- B. `condition1 or condition2`
- C. `condition1 || condition2`
- D. `condition1 // condition2`

MULTIPLE CHOICE

5. What will the following code print out?

```
dogs = 5 if dogs > 5: print("You have a lot of dogs") else: print("You have a few dogs")
```

Choose the correct answer:

- A. You have a lot of dogs
- B. You have a few dogs
- C. 5
- D. dogs

MULTIPLE CHOICE

6. What will the following code print out?

```
candies = 100 if candies >= 100: print("You have enough candy for the party!") else: print("You need to get more candy.")
```

Choose the correct answer:

- A. You have enough candy for the party!
- B. You need to get more candy.
- C. candies
- D. 100

DEBUG CODE

7. Edit the text box below to debug (fix) the code:**Code to Debug:**

```
1 age = 14
2
3 if age > 10
4     print("You can go to the movies.")
5 else:
6     print("You cannot go to the movies.")
```

DEBUG CODE

8. Edit the text box below to debug (fix) the code:**Code to Debug:**

```
1 age == 14
2
3 if age > 10:
4     print("You can go to the movies.")
5 else:
6     print("You cannot go to the movies.")
```

Challenges (3)

1. Can I Ride?

You are at a theme park and you really want to ride the tallest, most extreme roller coaster in the world. You get to the gate and realize there are rules as to who is allowed to ride the roller coaster. You must be at least 48 inches tall.

1. Create a program that checks to see if you can ride or not.
2. Create an input that asks the user to enter their **height in inches**.
3. Create outputs that let the user know if they can ride: **"You can ride the roller coaster."** or **"You can't ride the roller coaster."**

For example:

Input: `40`

Output: `You can't ride the roller coaster.`

Another example:

Input: `50`

Output: `You can ride the roller coaster.`

Don't forget to match the output exactly! Check spelling, punctuation, and spacing. One space after a period.

2. More than average?

1. Write a program that receives a number as an input. The number will be any number between **0 and 100**.
2. If the number is less than or equal to 50, print `Fewer than average`
3. If the number is greater than 50, print `More than average`

For example:

Input: `25`

Output: `Fewer than average`

Another example:

Input: `90`

Output: `More than average`

Don't forget to match the output exactly! Check spelling, punctuation, and spacing. One space after a period.

3. Black Square

1. Write a program that takes in two inputs – both integers. These integers will reflect the coordinates of a square on a chessboard.
2. For example, the top left corner will be **(1,1)**, and the bottom right corner would be **(8,8)**.
3. **The first integer corresponds to the row, the second corresponds to the column.**
4. If the square is black, print **Yes** , otherwise print **No** if it's white.

For example:

Inputs: **2** , **6**

Output: **No**

Another Example:

Inputs: **3** , **3**

Output: **No**

Hint: If the sum of the coordinates is odd, it's a black square.

Answer Keys & Solutions

Checkpoint Solutions

Python Conditionals

```
1 age = int(input("How old are you?"))
2 license = True
3 if age >= 16 and license == True:
4     print("You are old enough to drive.")
5 else:
6     print("You are not able to drive")
```

Questions

1. What is the correct way to see if two values are equal?

MULTIPLE CHOICE

Correct Answer:

- A. value = value ✗ Incorrect
- B. value == value ✓ Correct
- C. value is value? ✗ Incorrect
- D. value === value ✗ Incorrect

Explanation:

One equals sign is for assigning a variable.

2. What is the correct way to see if one condition AND another are met in Python?

MULTIPLE CHOICE

Correct Answer:

- A. condition1 and condition2 ✓ Correct
- B. condition1 & condition2 ✗ Incorrect
- C. condition1 ^ condition2 ✗ Incorrect
- D. condition1 && condition2 ✗ Incorrect

Explanation:

The word and is typed out.

3. How do you check to see if two values are NOT equal?

MULTIPLE CHOICE

Correct Answer:

- | | |
|---------------------|-------------|
| A. value /= value | ✗ Incorrect |
| B. value x= value | ✗ Incorrect |
| C. value NOT= value | ✗ Incorrect |
| D. value != value | ✓ Correct |

Explanation:

The exclamation point means not in programming.

4. How do you check to see if one condition OR another condition is met?

MULTIPLE CHOICE

Correct Answer:

- | | |
|-----------------------------|-------------|
| A. condition1 ^^ condition2 | ✗ Incorrect |
| B. condition1 or condition2 | ✓ Correct |
| C. condition1 condition2 | ✗ Incorrect |
| D. condition1 // condition2 | ✗ Incorrect |

Explanation:

The word "or" is typed out.

5. What will the following code print out?

MULTIPLE CHOICE

Correct Answer:

- | | |
|---------------------------|-------------|
| A. You have a lot of dogs | ✗ Incorrect |
|---------------------------|-------------|

B. You have a few dogs

✓ Correct

C. 5

✗ Incorrect

D. dogs

✗ Incorrect

Explanation:

If dogs is not greater than 5, then the else statement will trigger.

6. What will the following code print out?

MULTIPLE CHOICE

Correct Answer:

A. You have enough candy for the party!

✓ Correct

B. You need to get more candy.

✗ Incorrect

C. candies

✗ Incorrect

D. 100

✗ Incorrect

Explanation:

Since candies is greater than or equal to 100, the first section of code will run.

7. Edit the text box below to debug (fix) the code:

DEBUG CODE

Incorrect Code:

```
1 age = 14
2
3 if age > 10
4     print("You can go to the movies.")
5 else:
6     print("You cannot go to the movies.")
```

Correct Solution:

```
1 age = 14
2
3 if age > 10:
4     print("You can go to the movies.")
5 else:
6     print("You cannot go to the movies.")
```

Explanation:

There's a missing colon.

8. Edit the text box below to debug (fix) the code:

DEBUG CODE

Incorrect Code:

```
1 age == 14
2
3 if age > 10:
4     print("You can go to the movies.")
5 else:
6     print("You cannot go to the movies.")
```

Correct Solution:

```
1 age = 14
2
3 if age > 10:
4     print("You can go to the movies.")
5 else:
6     print("You cannot go to the movies.")
```

Explanation:

One equals sign is for assigning a variable. Two is for checking for equivalence.

Challenges

1. Can I Ride?

Solution:

```
1 height = int(input("Enter your height in inches. "))
2
3 if height >= 48:
4     print("You can ride the roller coaster.")
5 else:
6     print("You can't ride the roller coaster.")
```

2. More than average?

Solution:

```
1 first = int(input("Enter your number. "))
2
3 if first <= 50:
4     print("Fewer than average")
5 else:
```

```
6 print("More than average")
```

3. Black Square

Solution:

```
1 first = int(input("Enter your first number"))
2 second = int(input("Enter your second number"))
3
4 # if the sum of the coordinates is odd, it's a black square
5
6 math = (first + second)
7
8 if math % 2 == 1:
9     print("Yes")
10 else:
11     print("No")
```