

If Statements and Conditionals with Python Turtles

Textbook

If Statements and Conditionals with Python Turtles



Now let's create a way for the user to determine what happens on the program. This can be done with an if statement.

Let's add an input statement.

```
1 response = input("Should the turtle begin? Type yes or no")
```

Now let's make the program react to what the user puts in.

If Statement

Let's add a program that checks the user's response.

```

1 response = input("Should the turtle begin? Type yes or no")
2 if response == "yes":
3     turtle.forward(100)
4     turtle.right(45)
5     turtle.forward(100)

```

In this program, if the user's response is `yes`, the turtle will move forward. Anything that you want to happen if the response is yes needs to be indented one tab key.

Note: Indentation is VERY important in Python! If you don't indent correctly, the program will not work as desired.

String

In the if statement above, the word `"yes"` is surrounded by quotation marks. This is because the response the user is giving is a word. Words or statements are a kind of data type in Python called a string. They need to be surrounded by quotation marks.

Conditional

See how there are 2 equals signs between `response == yes`? This creates a [conditional](#). A conditional checks to see the relationship between two values.

- One equals sign is used to assign variables. `my_variable = 12`
- Two equals signs is to create conditionals. `if my_variable == 12:`

Other conditional values:

Sign	What it Checks For
<code>==</code>	Equal
<code>></code>	Greater Than
<code><</code>	Less Than
<code>>=</code>	Greater Than or Equal to
<code><=</code>	Less Than or Equal to
<code>≠</code>	Not Equal
<code>!=</code>	Not Equal

(`≠` can be done by using the option and equal keys on Mac, ALT and type 8800 on Windows.)

Else

You'll notice with this program that if the user doesn't type `yes`, it will automatically not run the code in the if statement. You can also specify what happens if they don't type yes. This is done with an `else` statement.

```

1 response = input("Should the turtle begin? Type yes or no")
2 if response == yes:
3     turtle.forward(100)
4     turtle.right(45)
5     turtle.forward(100)
6 else:
7     print("Ok, we won't start yet")

```

With the `else` statement, we can add functionality to the program if they don't type `yes`. In this case, we added a print statement.

Integer Input

If the input is an integer value, you need to make sure to convert it to an integer. Do this by adding the following to your input.

```
int( )
```

Your code would look something like this.

```

1 response = int(input("How many times should the turtle move?"))

```

Try it!

Then you can use the input as a number.

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!

Checkpoint

If Statements and Conditionals with Python Turtles

Create a program that will ask the user if the turtle should move forward or backward. Depending on the response, the turtle will move that direction.

1. Include the necessary code to start up a Python screen. (Import the library and generate a screen.)
2. Create a variable named `response` and assign it to an input that asks the user `Should the turtle move forward or backward?`
3. Create an if statement that checks to see if the input response was `"forward"`.
4. If the response was `"forward"`, move the turtle forward 50.
5. Else, move the turtle backward 50.

Requirements:

- Include the necessary code to start up a Python screen. (Import the library and generate a screen.)

- Create a variable named `response` and assign it to an input that asks the user `Should the turtle move forward or backward?`
- Create an if statement that checks to see if the input response was `"forward"` .
- If the response was `"forward"` , move the turtle forward 50. Don't forget to indent your code!
- Else, move the turtle backward 50. Don't forget to indent your code!

Questions (10)

1. Why is correct indentation emphasized in Python, especially within an if statement?

MULTIPLE CHOICE

Choose the correct answer:

- A. It makes the code look neat and organized
- B. Incorrect indentation may lead to program errors
- C. It improves the program's performance
- D. It makes code shorter

2. In Python, what is a string?

MULTIPLE CHOICE

Choose the correct answer:

- A. A numerical value
- B. A sequence of characters enclosed in quotation marks
- C. A conditional statement
- D. An indentation technique

3. What does the conditional response `== "yes"` check for?

MULTIPLE CHOICE

Choose the correct answer:

- A. If the response variable is assigned the value "yes"
- B. If the response variable is not equal to "yes"
- C. If the user input is greater than "yes"
- D. If the response variable is True

MULTIPLE CHOICE

4. Which symbol is used to check if two values are equal?

Choose the correct answer:

- A. =
- B. ==
- C. ===
- D. !=

SELECT MULTIPLE

5. Which two symbols are used to check if values are NOT equal? Select 2

Select all that apply:

- A. ≠
- B. !=
- C. ==
- D. #=

MULTIPLE CHOICE

6. Which symbol is used to mean "greater than"?

Choose the correct answer:

- A. >
- B. <
- C. =
- D. ==

MULTIPLE CHOICE

7. Which symbol is used to mean "less than?"

Choose the correct answer:

- A. >
- B. <
- C. =
- D. ==

8. What will the following code print out?

age = 10 if age < 5: print("You can ride") else: print("You cannot ride")

Choose the correct answer:

- A. 10
- B. 5
- C. You can ride
- D. You cannot ride

9. What will the following code print out?

favorite = "blue" if favorite = "blue": print("You get a navy shirt") else: print("You get a red shirt")

Choose the correct answer:

- A. You get a navy shirt
- B. You get a red shirt
- C. It will throw an error
- D. blue

10. Debug the following code:

Code to Debug:

```
1 dinner = "tacos"
2
3 if dinner = "pizza":
4     print("We need pepperoni!")
5 else:
6     print("We need taco shells!")
```


Challenges (5)

1. Color Choice

Create a program that will ask the user if they want the circle to be orange or not. If they do, draw an orange circle. If they don't, draw a different color of circle.

1. Include the necessary code to start up a Python screen. (Import the library and generate a screen.)
2. Create a variable named `response` and assign it to an input that asks the user `Do you want the circle to be orange? yes or no`.
3. Create an if statement that checks to see if the input response was `"yes"`.
4. If the response was `"yes"`, make the fillcolor `orange`.
5. Else, make the fillcolor something other than `orange`.

Requirements:

- Include the necessary code to start up a Python screen. (Import the library and generate a screen.)
- Create a variable named `response` and assign it to an input that asks the user `Do you want the circle to be orange? yes or no`.
- Create an if statement that checks to see if the input response was `"yes"`.
- If the response was `"yes"`, make the fillcolor `orange`.
- Else, make the fillcolor something other than `orange`.

2. Circle or Square

Create a program that asks the user if they want to draw a circle or a square. Then, it will draw that shape, depending on the response.

1. Include the necessary code to start up a Python screen. (Import the library and generate a screen.)
2. Create a variable named `response` and assign it to an input that asks the user `Do you want a circle or a square?`.
3. Create an if statement that checks to see if the input response was `"circle"`.
4. If the response was `"circle"`, draw a circle.
5. Else, draw a square. For an extra challenge, try doing it by using a for loop. (You will need to not only indent the for loop as a whole, but everything inside the for loop. This is why indentation is so important.)

Requirements:

- Include the necessary code to start up a Python screen. (Import the library and generate a screen.)
- Create a variable named `response` and assign it to an input that asks the user `Do you want a circle or a square?`.
- Create an if statement that checks to see if the input response was `"circle"`.
- If the response was `"circle"`, draw a circle.
- Else, draw a square. For an extra challenge, try doing it by using a for loop. (You will need to not only indent the for loop as a whole, but everything inside the for loop. This is why indentation is so important.)

3. Turtle or No?

Create a program that asks the user if they want to draw using a turtle shape. If so, the shape of the turtle will become a turtle. If not, the shape will stay classic.

1. Include the necessary code to start up a Python screen. (Import the library and generate a screen.)
2. Create a variable named `response` and assign it to an input that asks the user `Do you want to draw with a turtle shape? yes or no`.
3. Create an if statement that checks to see if the input response was `"yes"`.
4. If the response was `"yes"`, update your `turtle.shape` to be `turtle`.
5. Else, update your `turtle.shape` to be `classic`.

Requirements:

- Include the necessary code to start up a Python screen. (Import the library and generate a screen.)
- Create a variable named `response` and assign it to an input that asks the user `Do you want to draw with a turtle shape? yes or no`.
- Create an if statement that checks to see if the input response was `"yes"`.
- If the response was `"yes"`, update your `turtle.shape` to be `turtle`.
- Else, update your `turtle.shape` to be `classic`.

4. How Many Circles?

Create a program that asks the user how many circles to draw. If the response is greater than 0, draw that many circles. Else, draw a straight line.

1. Include the necessary code to start up a Python screen. (Import the library and generate a screen.)
2. Create a variable named `response` and assign it to an input that asks the user `How many circles should I draw? Enter a number 0 - 5`. Make sure to convert the input to an integer using `int()`.
3. Create an if statement that checks to see if the input `response` was greater than `0`. (Reminder that you don't need quotation marks for an integer.)
4. If `response` was greater than 0, create a for loop with the variable named `response` as the range.
5. Inside the for loop, include a circle command and a right or a left command.
6. Else, draw a straight line.

Requirements:

- Include the necessary code to start up a Python screen. (Import the library and generate a screen.)
- Create a variable named `response` and assign it to an input that asks the user `How many circles should I draw? Enter a number 0 - 5`. Make sure to convert the input to an integer using `int()`.
- Create an if statement that checks to see if the input `response` was greater than `0`. (Reminder that you don't need quotation marks for an integer.)
- If `response` was greater than 0, create a for loop with the variable named `response` as the range.
- Inside the for loop, include a circle command and a right or a left command.
- Else, draw a straight line.

5. Custom Title?

Create a program that asks the user if they want to create a custom title. If so, the program will ask the user what they want the title to be. Then it will enter that title as the custom title.

1. Include the necessary code to start up a Python screen. (Import the library and generate a screen.)
2. Create a variable named `response` and assign it to an input that asks the user `Do you want a custom title for your turtle screen? yes or no` .
3. Create an if statement that checks to see if the input response was `"yes"` .
4. If the response was `"yes"` , create a variable named `custom_title` and assign it to an input that asks the user `What do you want the title to be?` .
5. After the second input, use the variable named `custom_title` in a `turtle.title()` command.
6. Else, print `No custom title` .

Requirements:

- Include the necessary code to start up a Python screen. (Import the library and generate a screen.)
- Create a variable named `response` and assign it to an input that asks the user `Do you want a custom title for your turtle screen? yes or no` .
- Create an if statement that checks to see if the input response was `"yes"` .
- If the response was `"yes"` , create a variable named `custom_title` and assign it to an input that asks the user `What do you want the title to be?` .
- After the second input, use the variable named `custom_title` in a `turtle.title()` command.
- Else, print `No custom title` .

Answer Keys & Solutions

Checkpoint Solutions

If Statements and Conditionals with Python Turtles

```
1 import turtle
2 turtle.getscreen()
3
4 response = input("Should the turtle move forward or backward?")
5
6 if response == "forward":
7     turtle.forward(50)
8 else:
9     turtle.backward(50)
```

Questions

1. Why is correct indentation emphasized in Python, especially within an if statement?

MULTIPLE CHOICE

Correct Answer:

- A. It makes the code look neat and organized ✗ Incorrect
- B. Incorrect indentation may lead to program errors ✓ Correct
- C. It improves the program's performance ✗ Incorrect
- D. It makes code shorter ✗ Incorrect

Explanation:

Code must be indented correctly to work

2. In Python, what is a string?

MULTIPLE CHOICE

Correct Answer:

- A. A numerical value ✗ Incorrect
- B. A sequence of characters enclosed in quotation marks ✓ Correct

C. A conditional statement

✗ Incorrect

D. An indentation technique

✗ Incorrect

Explanation:

This is an example of a string "Hey"

3. What does the conditional response == "yes" check for?

MULTIPLE CHOICE

Correct Answer:

A. If the response variable is assigned the value "yes"

✓ Correct

B. If the response variable is not equal to "yes"

✗ Incorrect

C. If the user input is greater than "yes"

✗ Incorrect

D. If the response variable is True

✗ Incorrect

Explanation:

Two equals signs check for equivalence.

4. Which symbol is used to check if two values are equal?

MULTIPLE CHOICE

Correct Answer:

A. =

✗ Incorrect

B. ==

✓ Correct

C. ===

✗ Incorrect

D. !=

✗ Incorrect

Explanation:

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

5. Which two symbols are used to check if values are NOT equal? Select 2

SELECT MULTIPLE

Correct Answers:

A. \neq

✓ Correct

B. $!=$

✓ Correct

C. $==$

✗ Incorrect

D. $\# =$

✗ Incorrect

Explanation:

Two equals signs check for equivalence.

6. Which symbol is used to mean "greater than"?

MULTIPLE CHOICE

Correct Answer:

A. $>$

✓ Correct

B. $<$

✗ Incorrect

C. $=$

✗ Incorrect

D. $==$

✗ Incorrect

Explanation:

The symbol opens to the left.

7. Which symbol is used to mean "less than?"

MULTIPLE CHOICE

Correct Answer:

A. $>$

✗ Incorrect

B. $<$

✓ Correct

C. $=$

✗ Incorrect

D. $==$

✗ Incorrect

Explanation:

The symbol opens to the right.

8. What will the following code print out?

Correct Answer:

- A. 10 ✗ Incorrect
- B. 5 ✗ Incorrect
- C. You can ride ✗ Incorrect
- D. You cannot ride ✓ Correct

Explanation:

The variable named age is not less than 5.

9. What will the following code print out?

Correct Answer:

- A. You get a navy shirt ✗ Incorrect
- B. You get a red shirt ✗ Incorrect
- C. It will throw an error ✓ Correct
- D. blue ✗ Incorrect

Explanation:

This code only has one equal sign for the comparison

10. Debug the following code:

Incorrect Code:

```
1 dinner = "tacos"
2
3 if dinner = "pizza":
4     print("We need pepperoni!")
5 else:
6     print("We need taco shells!")
```

Correct Solution:

```
1 dinner = "tacos"
2
3 if dinner == "pizza":
4     print("We need pepperoni!")
```

```
5 else:
6     print("We need taco shells!")
```

Explanation:

This code is missing an equals sign

Challenges

1. Color Choice

Solution:

```
1 import turtle
2 turtle.getscreen()
3
4 response = input("Do you want the circle to be orange? yes or no")
5
6 if response == "yes":
7     turtle.fillcolor("orange")
8 else:
9     turtle.fillcolor("purple")
10
11
12 turtle.begin_fill()
13
14 turtle.circle(50)
15
16 turtle.end_fill()
```

2. Circle or Square

Solution:

```
1 import turtle
2 turtle.getscreen()
3
4 response = input("Do you want a circle or a square")
5
6 if response == "circle":
7     turtle.circle(30)
8 else:
9     for my_counter in range(4):
10         turtle.forward(30)
11         turtle.left(90)
```

3. Turtle or No?

Solution:

```
1 import turtle
2 turtle.getscreen()
3
4 response = input("Do you want to draw with a turtle shape? yes or no")
5
6 if response == "yes":
7     turtle.shape("turtle")
8 else:
9     turtle.shape("classic")
```

4. How Many Circles?

Solution:

```
1 import turtle
2 turtle.getscreen()
3
4 response = int(input("How many circles should I draw? Enter a number 0 - 5"))
5
6 if response > 0:
7     for my_counter in range(response):
8         turtle.circle(20)
9         turtle.left(20)
10 else:
11     turtle.forward(20)
```

5. Custom Title?

Solution:

```
1 import turtle
2 turtle.getscreen()
3
4 response = input("Do you want a custom title for your turtle screen? yes or no")
5
6 if response == "yes":
7     custom_title = input("What do you want the title to be?")
8     turtle.title(custom_title)
9 else:
10     print("No custom title")
```