

## Else If Statements with Python Turtles

### Textbook

## Else If Statements with Python Turtles



Let's add some further customization to our if statements. Let's start with the following code as an example.

```
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")
```

But what if we want to respond differently if the user enters an input of `no` ?

### Elif Statement

We can add an else if statement. In Python, this is shortened to `elif` .

```

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 elif response == "no":
7     print("Ok, we won't start yet")
8 else:
9     print("Please respond with yes or no")

```

Now we check if the user enters `yes` and move the turtle. We check if the user enters `no` and add a certain print statement. We also add a print statement if the user enters anything other than `yes` or `no`.

Note: pay attention to indentation. See how each command is indented after the `if`, `elif`, and `else` statements? This is important for Python to work.

## Multiple Elif Statements

You can also have as many elif statements as you want.

```

1 response = input("Should the turtle begin? Type yes, no, or maybe")
2 if response == "yes":
3     turtle.forward(100)
4 elif response == "no":
5     print("Ok, we won't start yet")
6 elif response == "maybe":
7     print("Okay, take some time to think about it")
8 else:
9     print("Please respond with yes or no")

```

This gives the user even more options to choose from.

## Assessing the Reasonableness of Solutions

Finding an answer to a math problem or a coding challenge is great, but a truly skilled problem-solver doesn't stop there. They also assess the reasonableness of their solutions. This means asking yourself, "Does this answer actually make sense in the real world, given the problem I'm trying to solve?"

To do this, you can start by estimating to discover possible solutions even before you begin calculating. Think about what a sensible answer might look like. You can also use benchmark quantities – like knowing that a price can't be negative, or that a turtle moving forward 100 steps will draw a longer line than moving 10 steps – to determine if your solution makes sense. Throughout the process, check your calculations as you work, and always verify your possible solutions by explaining the methods used. Finally, evaluate your results based on the given context. Does your answer fit the story of the problem? If your calculation tells you that a single cookie recipe yields 5,000 cookies, you'd know to double-check your work! By always questioning if your solution is reasonable, you become much more accurate and confident in your problem-solving.

---

## Checkpoint

### Else If Statements with Python Turtles

Create a program where the turtle moves forward, then asks the user if the turtle should turn left or right. Depending on the answer, the turtle will turn, and then move forward again.

1. Include the necessary code to start up a Python screen. (Import the library and generate a screen.)
2. Move the turtle forward 30.
3. Create a variable named `response` and assign it to an input that asks the user `Should the turtle turn left or right?`
4. Create an if statement that checks to see if the input response was `"left"`.
5. If the response was `"left"`, turn the turtle to the left 45.
6. Create an elif statement that checks to see if the input response was `"right"`.
7. If the response was `"right"`, turn the turtle to the right 45.
8. Else, print `invalid response`.
9. After all the if statement options, move the turtle forward 30 again.

### Requirements:

- Include the necessary code to start up a Python screen. (Import the library and generate a screen.)
- Move the turtle forward 30.
- Create a variable named `response` and assign it to an input that asks the user `Should the turtle turn left or right?`
- Create an if statement that checks to see if the input response was `"left"`.
- If the response was `"left"`, turn the turtle to the left 45.
- Create an elif statement that checks to see if the input response was `"right"`.
- If the response was `"right"`, turn the turtle to the right 45.
- Else, print `invalid response`.
- After all the if statement options, move the turtle forward 30 again.

### Questions (10)

#### 1. How is the syntax for "else if" written in Python?

MULTIPLE CHOICE

Choose the correct answer:

- A. elseif
- B. elif
- C. elseif statement
- D. else if

## 2. Why is indentation important in Python?

Choose the correct answer:

- A. It makes the code look neat and organized
- B. It determines the speed of the code
- C. It's necessary for Python code to work correctly
- D. Indentation is not important in Python

## 3. What will be printed if the user enters "maybe" in the last code block?

```
response = input("Should the turtle begin? Type yes, no, or maybe") if response == "yes": t.forward(100) elif response == "no": print("Ok, we won't start yet") elif response == "maybe": print("Okay, take some time to think about it") else: print("Please respond with yes or no")
```

Choose the correct answer:

- A. Ok, we won't start yet
- B. Please respond with yes or no
- C. Okay, take some time to think about it
- D. Should the turtle begin? Type yes, no, or maybe

## 4. What action does the turtle take if the user input is "yes"?

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

Choose the correct answer:

- A. Move forward by 100 units
- B. Move backward by 100 units
- C. Turn right by 90 degrees
- D. Turn left by 20 degrees

**5. What is the purpose of the "else" statement in this code block?**

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

**Choose the correct answer:**

- A. It checks if the user input is "yes"
- B. It prints a statement if the user input is not "yes"
- C. It prompts the user for input
- D. It moves the turtle forward by 100 units

**6. What will be printed if the user enters "nope" in the following code block?**

```
response = input("Should the turtle begin? Type yes, no, or maybe") if response == "yes": turtle.forward(100) elif response
== "no": print("Ok, we won't start yet") elif response == "maybe": print("Okay, take some time to think about it") else:
print("Please respond with yes or no")
```

**Choose the correct answer:**

- A. Please respond with yes or no
- B. Okay, take some time to think about it
- C. Ok, we won't start yet
- D. Should the turtle begin? Type yes, no, or maybe

**7. In the following code block, what happens if the user enters "no"?**

```
response = input("Should the turtle begin? Type yes or no") if response == "yes": turtle.forward(100) turtle.right(45)
turtle.forward(100) elif response == "no": print("Ok, we won't start yet") else: print("Please respond with yes or no")
```

**Choose the correct answer:**

- A. The turtle moves forward by 100 units
- B. The program terminates
- C. It will print: Please respond with yes or no
- D. It will print: Ok, we won't start yet

## 8. What is the advantage of using multiple elif statements?

MULTIPLE CHOICE

Choose the correct answer:

- A. It simplifies the code
- B. It allows for multiple conditions to be checked
- C. It increases the turtle's speed
- D. It reduces bugs

## 9. Debug the following code:

DEBUG CODE

Code to Debug:

```
1 response = input("Should the turtle move forward? yes or no")
2 if response == "yes":
3     turtle.forward(100)
4 elif response = "no":
5     print("We won't move the turtle")
6 else:
7     print("Invalid response")
```

## 10. Debug the following code:

DEBUG CODE

Code to Debug:

```
1 response = input("Should the turtle move forward? yes or no")
2 if response == "yes":
3     turtle.forward(100)
4 elif response == "no"
5     print("We won't move the turtle")
6 else:
7     print("Invalid response")
```

## Challenges (4)

### 1. Pen Color

Create a program that asks the user if the turtle should draw in green or blue. Depending on the answer, the turtle pencolor will change.

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 pen draw green or blue?` .
3. Create an if statement that checks to see if the input response was `"green"` .
4. If the response was `"green"` , set the pencolor to `green` .
5. Create an elif statement that checks to see if the input response was `"blue"` .
6. If the response was `"blue"` , set the pencolor to `blue` .
7. Else, print `invalid response` .
8. After all the if statement options, move the turtle forward 30.

#### 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 pen draw green or blue?` .
- Create an if statement that checks to see if the input response was `"green"` .
- If the response was `"green"` , set the pencolor to `green` .
- Create an elif statement that checks to see if the input response was `"blue"` .
- If the response was `"blue"` , set the pencolor to `blue` .
- Else, print `invalid response` .
- After all the if statement options, move the turtle forward 30.

## 2. Warm Color Cool Color

Create a program that asks the user if they want the background of their turtle screen to be a warm or a cool color. Warm colors are red, orange, and yellow. Cool colors are green, blue, and purple.

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 background color be a warm or a cool color?`.
3. Create an if statement that checks to see if the input response was `"warm"`.
4. If the response was `"warm"`, set the background to a warm color.
5. Create an elif statement that checks to see if the input response was `"cool"`.
6. If the response was `"cool"`, set the background to a cool color.
7. Else, print `invalid response`.

### 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 background color be a warm or a cool color?`.
- Create an if statement that checks to see if the input response was `"warm"`.
- If the response was `"warm"`, set the background to a warm color.
- Create an elif statement that checks to see if the input response was `"cool"`.
- If the response was `"cool"`, set the background to a cool color.
- Else, print `invalid response`.

## 3. How many Angles?

Create a program that asks the user how many angles to draw 1, 2, or 3. Depending on the answer, the turtle will draw that many angles.

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 angles should I draw? Enter number 1-3`. Make sure to convert this input into an integer using `int()`.
3. Create an if statement that checks to see if the input response was `1`. If the response was `1`, draw one angle.
4. Create an elif statement that checks to see if the input response was `2`. If the response was `2`, draw two angles.
5. Create an elif statement that checks to see if the input response was `3`. If the response was `3`, draw three angles.
6. Else, print `invalid response`.

### 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 angles should I draw? Enter number 1-3`. Make sure to convert this input into an integer using `int()`.
- Create an if statement that checks to see if the input response was `1`. If the response was `1`, draw one angle.
- Create an elif statement that checks to see if the input response was `2`. If the response was `2`, draw two angles.
- Create an elif statement that checks to see if the input response was `3`. If the response was `3`, draw three angles.
- Else, print `invalid response`.



#### 4. Create your own If Statement

Create your own if statement program!

1. Include the necessary code to start up a Python screen. (Import the library and generate a screen.)
2. Include an input statement.
3. Include an if statement.
4. Include an else if statement.
5. Include an else statement.

##### **Requirements:**

- Include the necessary code to start up a Python screen. (Import the library and generate a screen.)
- Include an input statement.
- Include an if statement.
- Include an else if statement.
- Include an else statement.

## Answer Keys & Solutions

### Checkpoint Solutions

#### Else If Statements with Python Turtles

```
1 import turtle
2 turtle.getscreen()
3
4 turtle.forward(30)
5
6 response = input("Should the turtle turn left or right?")
7
8 if response == "left":
9     turtle.left(45)
10 elif response == "right":
11     turtle.right(45)
12 else:
13     print("invalid response")
14
15
16 turtle.forward(30)
```

### Questions

#### 1. How is the syntax for "else if" written in Python?

MULTIPLE CHOICE

Correct Answer:

- A. elseif ✗ Incorrect
- B. elif ✓ Correct
- C. elseif statement ✗ Incorrect
- D. else if ✗ Incorrect

#### Explanation:

The word else is shortened.

#### 2. Why is indentation important in Python?

MULTIPLE CHOICE

Correct Answer:

A. It makes the code look neat and organized

✗ Incorrect

B. It determines the speed of the code

✗ Incorrect

C. It's necessary for Python code to work correctly

✓ Correct

D. Indentation is not important in Python

✗ Incorrect

#### Explanation:

Indentation is a part of correct Python syntax

### 3. What will be printed if the user enters "maybe" in the last code block?

MULTIPLE CHOICE

#### Correct Answer:

A. Ok, we won't start yet

✗ Incorrect

B. Please respond with yes or no

✗ Incorrect

C. Okay, take some time to think about it

✓ Correct

D. Should the turtle begin? Type yes, no, or maybe

✗ Incorrect

#### Explanation:

Which print statement comes after the conditional that checks for "maybe"?

### 4. What action does the turtle take if the user input is "yes"?

MULTIPLE CHOICE

#### Correct Answer:

A. Move forward by 100 units

✓ Correct

B. Move backward by 100 units

✗ Incorrect

C. Turn right by 90 degrees

✗ Incorrect

D. Turn left by 20 degrees

✗ Incorrect

#### Explanation:

What code is indented after the conditional response == "yes"

### 5. What is the purpose of the "else" statement in this code block?

Correct Answer:

- A. It checks if the user input is "yes" ✗ Incorrect
- B. It prints a statement if the user input is not "yes" ✓ Correct
- C. It prompts the user for input ✗ Incorrect
- D. It moves the turtle forward by 100 units ✗ Incorrect

#### Explanation:

In this code, the else statement runs if the user enters anything other than "yes"

### 6. What will be printed if the user enters "nope" in the following code block?

MULTIPLE CHOICE

Correct Answer:

- A. Please respond with yes or no ✓ Correct
- B. Okay, take some time to think about it ✗ Incorrect
- C. Ok, we won't start yet ✗ Incorrect
- D. Should the turtle begin? Type yes, no, or maybe ✗ Incorrect

#### Explanation:

This print statement is triggered by the else statement.

### 7. In the following code block, what happens if the user enters "no"?

MULTIPLE CHOICE

Correct Answer:

- A. The turtle moves forward by 100 units ✗ Incorrect
- B. The program terminates ✗ Incorrect
- C. It will print: Please respond with yes or no ✗ Incorrect
- D. It will print: Ok, we won't start yet ✓ Correct

### Explanation:

What print statement follows the conditional that checks for an input of "no"

## 8. What is the advantage of using multiple elif statements?

MULTIPLE CHOICE

### Correct Answer:

- A. It simplifies the code ✗ Incorrect
- B. It allows for multiple conditions to be checked ✓ Correct
- C. It increases the turtle's speed ✗ Incorrect
- D. It reduces bugs ✗ Incorrect

### Explanation:

Else if statements are available if the original condition is not met

## 9. Debug the following code:

DEBUG CODE

### Incorrect Code:

```
1 response = input("Should the turtle move forward? yes or no")
2 if response == "yes":
3     turtle.forward(100)
4 elif response = "no":
5     print("We won't move the turtle")
6 else:
7     print("Invalid response")
```

### Correct Solution:

```
1 response = input("Should the turtle move forward? yes or no")
2 if response == "yes":
3     turtle.forward(100)
4 elif response == "no":
5     print("We won't move the turtle")
6 else:
7     print("Invalid response")
```

### Explanation:

This code is missing an equals sign

## 10. Debug the following code:

### Incorrect Code:

```
1 response = input("Should the turtle move forward? yes or no")
2 if response == "yes":
3     turtle.forward(100)
4 elif response == "no"
5     print("We won't move the turtle")
6 else:
7     print("Invalid response")
```

### Correct Solution:

```
1 response = input("Should the turtle move forward? yes or no")
2 if response == "yes":
3     turtle.forward(100)
4 elif response == "no":
5     print("We won't move the turtle")
6 else:
7     print("Invalid response")
```

### Explanation:

This code is missing a colon

## Challenges

### 1. Pen Color

#### Solution:

```
1 import turtle
2 turtle.getscreen()
3
4 response = input("Should the pen draw green or blue?")
5
6 if response == "green":
7     turtle.pencolor("green")
8 elif response == "blue":
9     turtle.pencolor("blue")
10 else:
11     print("invalid response")
12
13
14 turtle.forward(30)
```

### 2. Warm Color Cool Color

**Solution:**

```
1 import turtle
2 turtle.getscreen()
3
4 response = input("Should the background color be a warm or a cool color?")
5
6 if response == "warm":
7     turtle.bgcolor("yellow")
8 elif response == "cool":
9     turtle.bgcolor("lightblue")
10 else:
11     print("invalid response")
```

**3. How many Angles?****Solution:**

```
1 import turtle
2 turtle.getscreen()
3
4 response = int(input("How many angles should I draw? Enter number 1-3"))
5
6 if response == 1:
7     turtle.forward(50)
8     turtle.left(45)
9     turtle.forward(50)
10 elif response == 2:
11     turtle.forward(50)
12     turtle.left(45)
13     turtle.forward(50)
14     turtle.left(45)
15     turtle.forward(50)
16 elif response == 3:
17     turtle.forward(50)
18     turtle.left(45)
19     turtle.forward(50)
20     turtle.left(45)
21     turtle.forward(50)
22     turtle.left(45)
23     turtle.forward(50)
24 else:
25     print("invalid response")
```

**4. Create your own If Statement****Solution:**

```
1 import turtle
2 turtle.getscreen()
3
4 response = input("Want to draw with a turtle or a square?")
5
```

```
6 if response == "turtle":  
7     turtle.shape("turtle")  
8 elif response == "square":  
9     turtle.shape("square")  
10 else:  
11     print("invalid response")
```