LEARN PYTHON

Number Guess

Wanna play a game? In this project, we'll build a program that rolls a pair of dice and asks the user to guess the sum. If the user's guess is equal to the total value of the dice roll, the user wins! Otherwise, the computer wins.

The program should do the following:

1. Roll a pair of dice.
2. Add the values of the roll.
3. Ask the user to guess a number.
4. Compare the user's guess to the total value.
5. Determine the winner (user or computer).

Let's begin!

Number Guess

1. Begin by writing a multi-line comment that starts on line 1.

   Describe what this program does.

2. To make sure that the rolls are random, we will need some Python code that isn't built-in – we need to import the `randint` function, from the `random` module.

   The syntax for importing is:

   ```
   from module import function
   ```

3. You'll also need to import more code that will be used to simulate dice rolls.

   Import the `sleep` function, from the `time` module.

4. We have imported the code that we will use later, let's move on!

   First, we'll have to prompt the user for their guess.

   Create a function called `get_user_guess()`. The function should take no arguments.

5. Inside of the function, prompt the user for their guess. Store the input into a variable called `guess`.

   Don't forget to use proper Python indentation!

6. By default, using `raw_input` alone will store the user's input as a string. Since the user is guessing a whole number, we will need an integer, not a string.

   Wrap the `raw_input("Guess a number: ")` part of your code with `int()`.

7. The name of this function is `get_user_guess()`, which implies that when the function is called, it should get, or `return`, the user's guess.

   On the next line, return the user's guess.

8. Great! This function is complete and we will use it later to complete our entire program.

   Now it's time to start building the rest of the game.

   Create a second function called `roll_dice()`.

   The `roll_dice` function will be used to simulate the rolling of a pair of dice.

9. We'd like to specify the number of sides that a single die will have.

   Modify the function to include a parameter called `number_of_sides`.

10. *Inside* the function, let's simulate the first die roll.

    A single die can land on any value that's at least 1 and no greater than the number of sides.

    Use the `randint` function that you imported earlier to generate a random integer between `1` and `number_of_sides`. The syntax for the function looks like:

    ```
    x = randint(low, high)
    ```

    Set the result equal to a variable called `first_roll`.

11. On the next line, simulate the second roll.

    This line of code will look almost identical to the code from the last task. However, this time, set the result equal to a variable called `second_roll`.

12. Now let's calculate the maximum value the program can possibly roll. This will help us set some rules for the game later.

    On the next line, create a variable called `max_val` and set it equal to `number_of_sides` times 2 (since there are two dice).

13. On the next line, let the user know what the maximum possible value is by using string formatting to print the `max_val`.

    Remember, `max_val` is an integer so use `%d`.

14. On the next line, call the `get_user_guess()` function. Remember that the function will return the user's guess after prompting the user.

    Store the returned value into a variable called `guess`.

15. Great! We have written code that simulates a dice roll and asks the user for their guess. Now it's time to write the rules that will determine the winner of the game.

    But what if the user guesses a number that's larger than the total possible value of the dice roll? That shouldn't be allowed...

    On the next line, write an `if` statement that checks if the user's guess is greater than the maximum value.

16. Within the `if` block, let the user know that their guess is invalid by printing an appropriate message.

17. This is a good time pause and test your program!

    Somewhere outside of the `roll_dice` function, call the `roll_dice` function:

    ```
    roll_dice(6)
    ```

    Click Save. Then, in the terminal, type the following command and press `enter`:

    ```
    python NumberGuess.py
    ```

    Test the code by guessing a number higher than `12`, the maximum possible number.

18. Let's continue on!

    Add to the `if` statement by starting an `else` block.

    Inside of the `else` block, print the message `Rolling...` to the user.

19. Staying inside the `else` block, `sleep` the program for 2 seconds on the next line to simulate the dice rolling:

    ```
    sleep(2)
    ```

20. On the next line, use string formatting to print the first roll.

    Remember that you stored the first roll in an `int` variable earlier, so use `%d`.

    Then, on the next line, sleep the program for 1 second.

21. On the next line, `print` the value of the second roll.

    On the line after that, `sleep` the program again for 1 second.

22. To determine a winner, we will need to use the total value of the dice roll.

    On the next line, create a variable called `total_roll` and set it equal to the sum of the first roll and the second roll.

23. On the next line, `print` the total roll to the user.

    On the following line, `print` the message `Result...` to the user.

    Directly after that, `sleep` the program for 1 second to build suspense!

24. Keeping inside of the `else` block, add an `if` statement checks if the user's guess is equal to the total roll.

    If it is, `print` a friendly message to the user informing them that they won.

25. What if the user's guess is not equal to the total roll?

    Inside of an `else` block, `print` a message to the user informing them that they lost.

26. Great! We're almost done! For this program to run, we have to call the function.

    Somewhere outside of the `roll_dice` function, call the `roll_dice` function.

    Make sure to specify the number of sides a single die has as the argument!

27. Finally, let's play Number Guess!

First, click Save. Then, in the terminal, type the following command and press `enter`:

```
python NumberGuess.py
```

Did you win or lose?