Documentation Template

| ***You MUST provide evidence showing how the problem has been decomposed, how the components have been developed and trialled, and of how they have been assembled and tested to create a final, working outcome.*** |
| --- |

### Outline / Decomposition

*Please write down your task decomposition here (a numbered list is a good idea)*

1. *Create variables and list*
2. *Introduce the game and how to play*
3. *Create a loop that loops the game until the user does not want to play again.*
4. *Generate a random integer between 1-100 and store it in a variable*
5. *Ask the user to input a guess*
6. *Store the user guess in a list.*
7. *Determine whether the guess is higher or lower than the guess.*
8. *If the guess is correct, congratulate the user and ask if they want to play again*
9. *If not, tell the user if the mystery number is higher or lower than their guess.*
10. *Ask them to guess again.*
11. *Keep looping until the user guesses correctly.*
12. *Ask the user if they want to play again.*

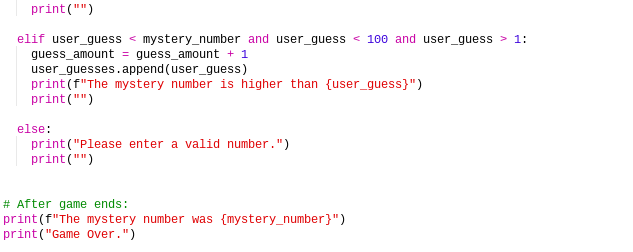
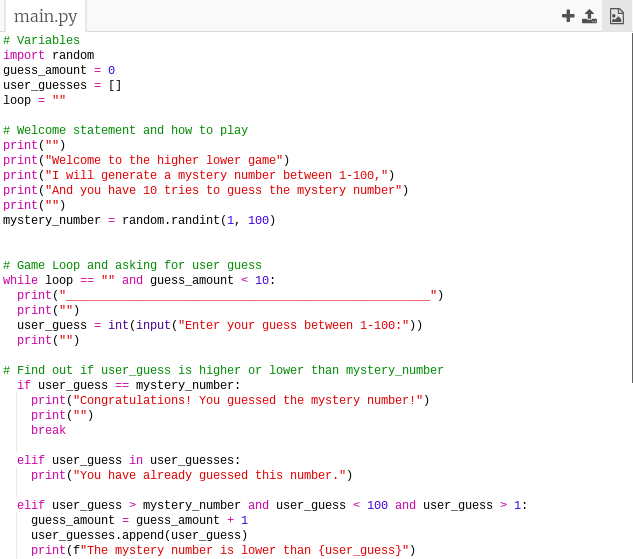
*Variations:*

1. *Ask the user for how many rounds they would like to play. Maximum = 10 rounds*
2. *Create variables for the amount of rounds, rounds played, and scores.*
3. *Create a list to store the scores for each round.*
4. *At the end of the given number of rounds, tell the user what their best, worst and average scores were.*
5. *Ask them if they wish to play again.*

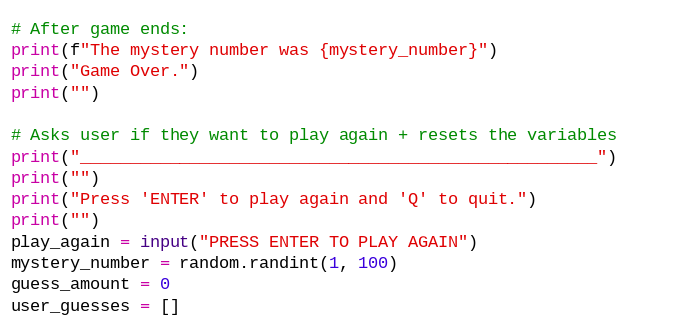
### Version Log

*Your version log should go here. Annotated screenshots are a good idea at this point*

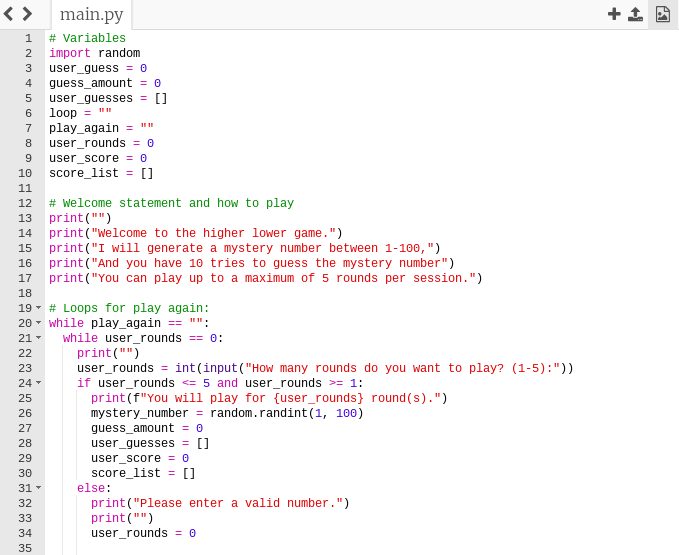
***Below - The base version of the game with no additional variations. Runs through the game once***

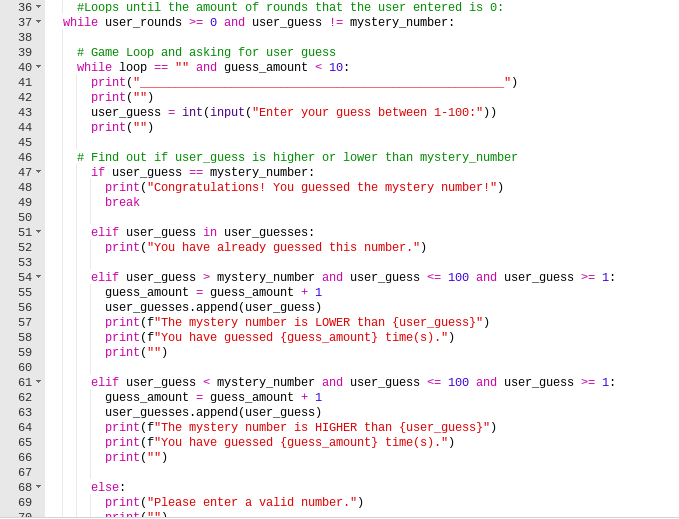
**

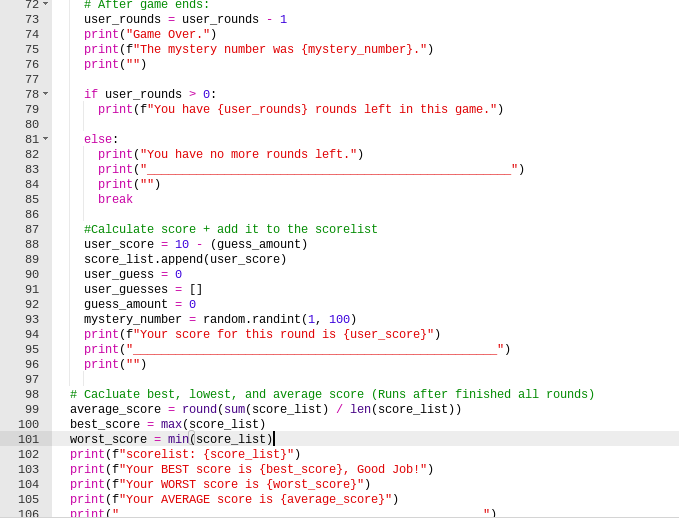
***Below - added the feature to ask user to play again***

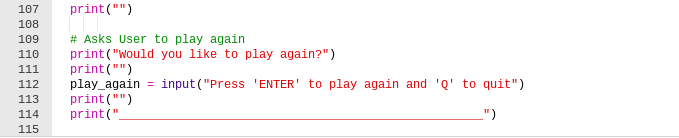
******

***Below - added multiple variations to the game:***

******

******

******

******

### Component Testing

*Show that you have tested each component here. You should have a test plan and then screenshot proof for each component. You should also include notes that justify the major decisions you made.*

***Major decisions made:***

*1. Using .append() to add the user\_guess into the usr\_guesses list:*

*It works properly over using user\_guesses = user\_guesses + user\_guess*

*2. Using a score list to store each score:  
To store scores to calculate best, worst, and average.*

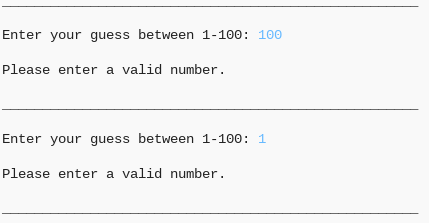
***Component Testing:***

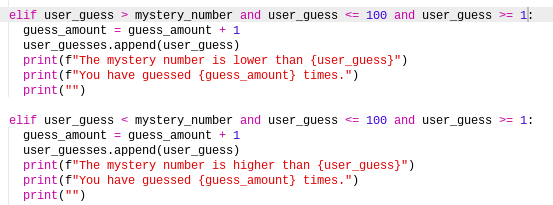
*Tested entering 1 and 100 as a guess. Fixed the mistake by*

*changing < 100 and > 1 to <=100 and >= 1*

*Tested the loops before changing it to two different loops,*

*1 for looping asking the user for their guess, and another for refreshing each round (after the user guesses the correct number).*

**

**

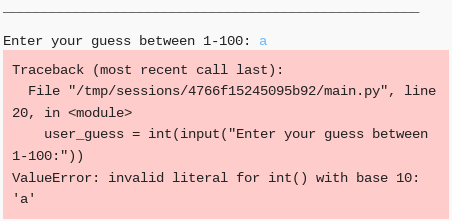
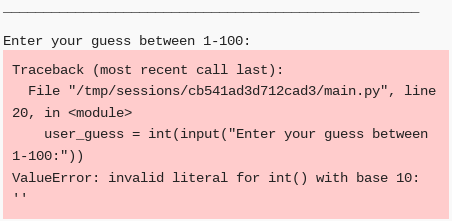
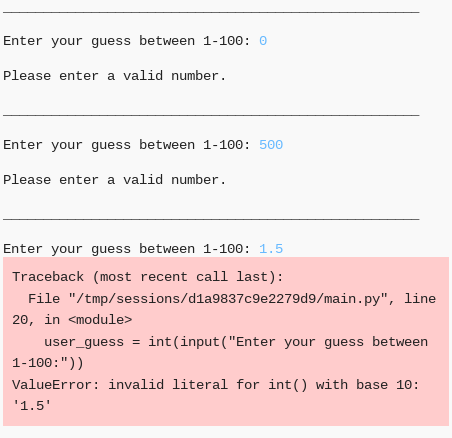
### Assembled Outcome Testing

*Please show testing for your assembled outcome below. This should include a test plan followed by screenshot proof*

***Test plan:*** *test the expected user inputs for the user guess as well as unexpected inputs such as letters, floats, blanks, and numbers over 100 or under 1.*

***Test result:*** *all inputs that are not integers return a ValueError and if you enter an integer above 100 or below 1, it asks the user to enter a valid number.*

***Screenshot proofs below:***

**

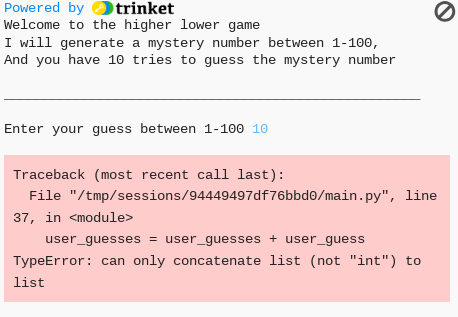
### Usability Testing

*Write a list of things improvements which need to be made based on your usability testing. Then write down what you changed.*

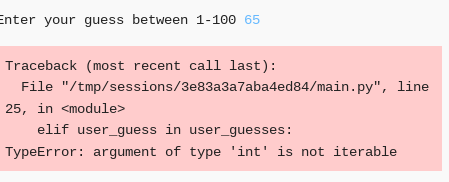
***Changes that need to be made:***

*Adding the user\_guess to the user\_guesses list*

***Error:***

**

*Checking if the user\_guess is already in the user\_guesses list*

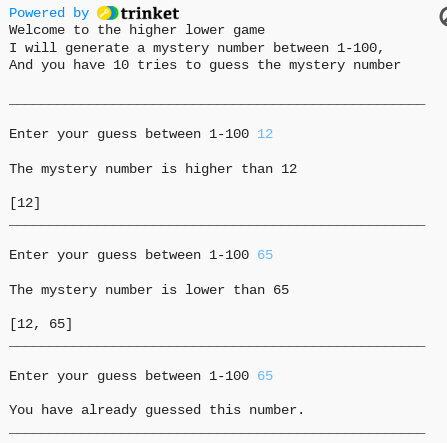
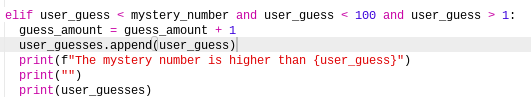
**

### Post Usability Test…

*Show that your post usability testing program works correctly*

***Changing user\_guess = user\_guesses + user\_guess to .append()***

*The images below show how using the .append() function correctly adds the inputted guess from the user to the list..*

**

### Social and End User Considerations…

**How did you ensure that your task was suitable for your chosen audience?**

*I ensured that my task was suitable for my audience because I made it easy to use and have clear instructions on what to do.*

**How have you honoured copyright?**

*I haven’t*

**How did you make your quiz easy to use?**

*I made my quiz easy to use because the system gives you a statement on if the mystery number is higher or lower than what you have guessed and it also prints out the amount of guesses you have done. I also made it easy to understand in general and have clear instructions.*