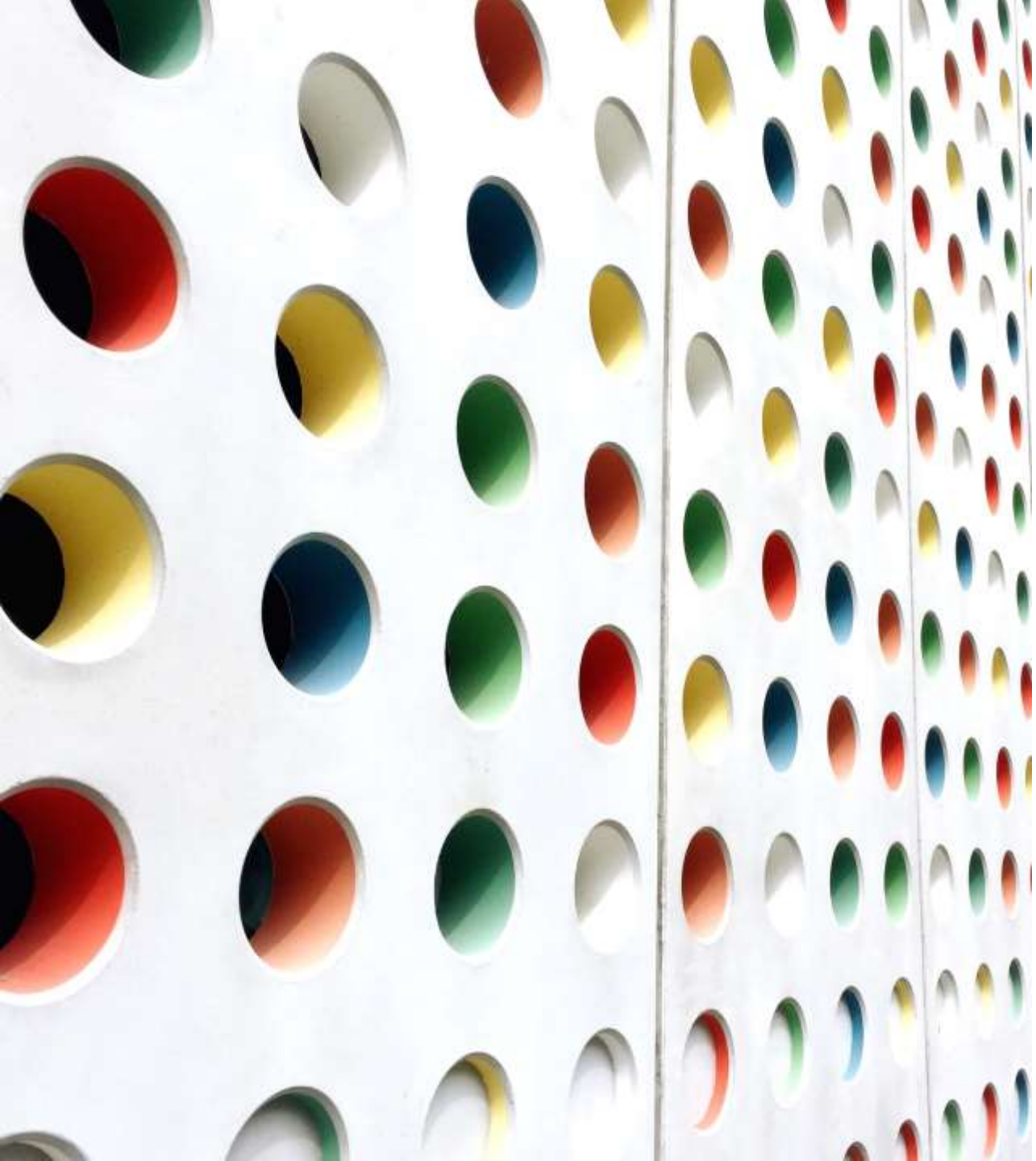


Enhancing The Interactivity and Visualization Features of The Online Python Tutor Web Application

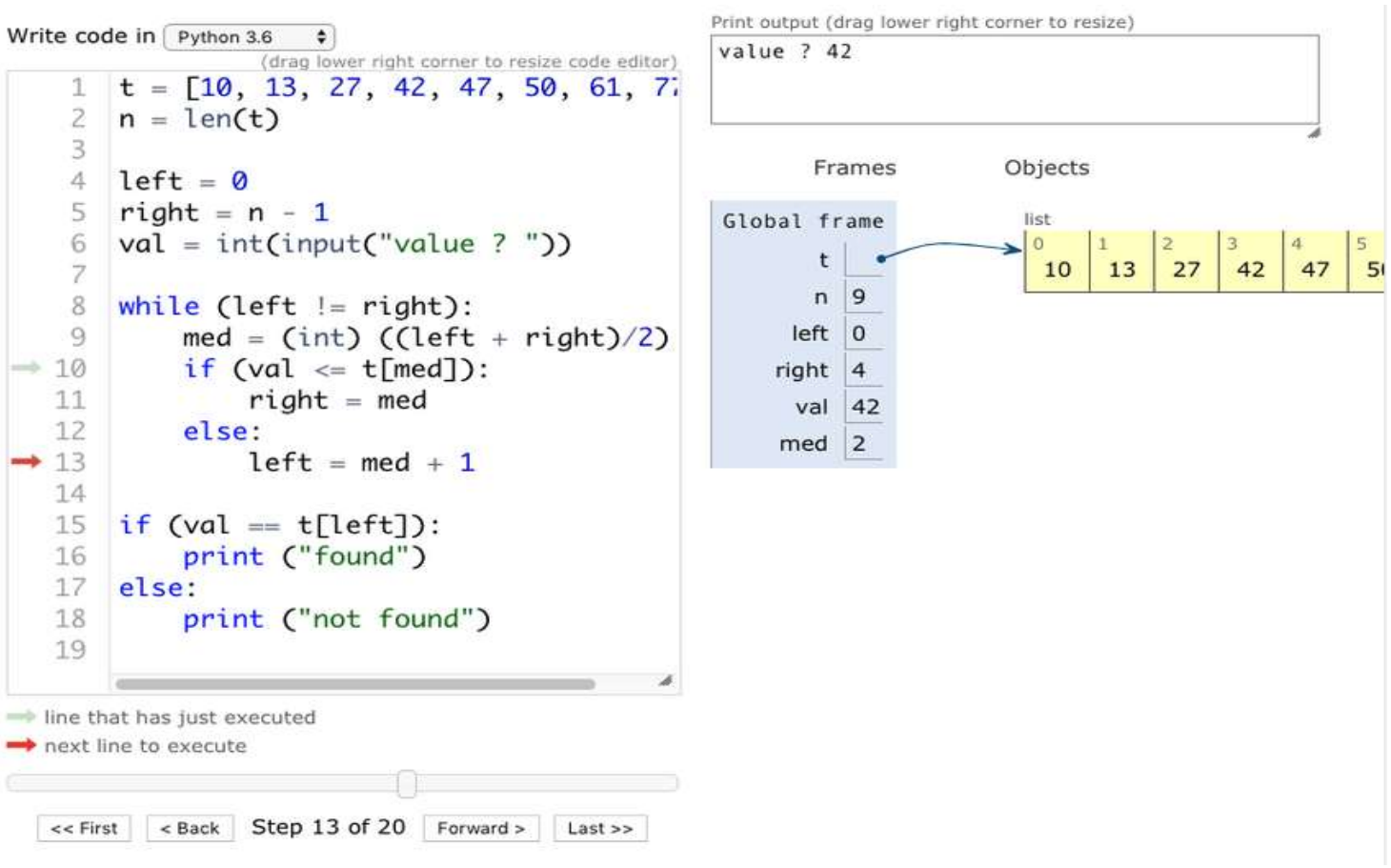
FINAL DEMONSTRATION:

SIVIWE QOLOHLE

Python Learning Tools

- Existing tools should provide foundational introductions to programming for beginners.
- Visualisations are need to make programming less abstract.
- Increase interactivity to check and deepen user understanding.

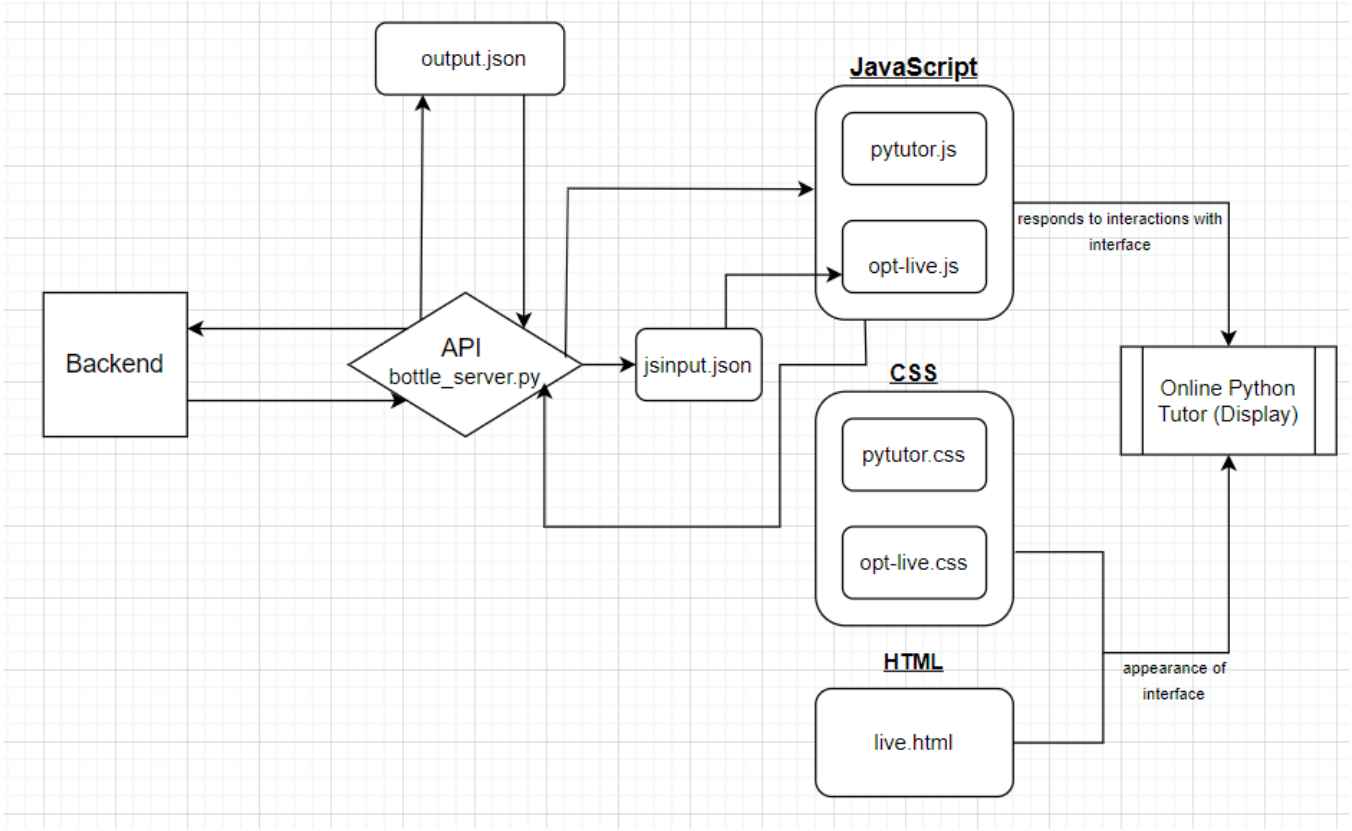
Online Python Tutor



Requirements

- **Highlighting Feature:**
  - Make changes in the editor more visible
- **Variable Testing Feature:**
  - Test users understanding of code.
- **Description Keywords:**
  - Reminder of structure and definitions of Python keywords.

Implementation: Architecture



Implementation: Interactivity

For the following code:

```
x=10
v=4
```

The following trace file is created (output.json):


```
{"code": "x=10 \nv=4 \n", "trace": [{"line": 1, "event": "step_line", "func_name":
"<module>", "globals": {}, "ordered_globals": [], "stack_to_render": [], "heap":
{}, "stdout": ""}, {"line": 2, "event": "step_line", "func_name": "<module>",
"globals": {"x": 10}, "ordered_globals": ["x"], "stack_to_render": [], "heap": {},
"stdout": ""}, {"line": 2, "event": "return", "func_name": "<module>", "globals":
{"x": 10, "v": 4}, "ordered_globals": ["x", "v"], "stack_to_render": [], "heap":
{}, "stdout": ""}]}
```


The following values are extracted into the jsinput.json file:


```
{
    "1": "N/A , N/A",
    "2": "x , 10",
    "3": "v , 4"
}
```

Implementation: Interactivity and Visualisation

VARIABLE TESTING

This feature allows you to test your understanding of the effect of each line, on the variables.

Instructions:

Predict which variable will change in the following line, as well as the value it will change to.

The **red arrow** in the gutter points to the line you should predict for.

If there is no change, type **N/A** in both the "Changed Variable" and the "New Value" sections

Please see example below

```
1  x=3
2  c=4
3  m=3
4  s=3
5  for i in range(2,15,2):
6      if (i%3)==0:
7          x=30
8      else:
9          c=20
10         m=x+c
```

Global frame

x	3
c	20
m	3
s	3
i	2

Close Tester

Changed Variable: m New Value: 23 Mark Answer

Result: correct value

Solution Step:7: Variable m Changed to the Value: 23

Implementation: Keyword Descriptions

```
1  x=3
2  c=4
3  m=3
4  s=3
5  for i in range(2,15,2):
6      if (i%3)==0:
7          x=30
8      else:
9          c=20
10         m=x+c
11 for
```

The 'for' loop is used to iterate over a sequence of values. Structure: In 'for i in range(x):' 'x' represents the number of iterations. In 'for i in range(v,y)' 'v' is the starting value and 'y' is the end value. In 'for i in range(c,z,r)' 'c' is the starting value, 'z' is the ending and 'r' is the increment.

Line that has Executed

Next Line to Execute

```
1  x=3
2  c=4
3  m=3
4  s=3
5  for i in range(2,15,2):
6      if (i%3)==0:
7          x=30
8      else:
9          c=20
10         m=x+c
11 if
```

This keyword checks if a condition is met. Example: x=6 if x==6: print('yes')

Line that has Executed

Next Line to Execute

Testing and Evaluation

- **Overall Application Testing**
- **Unit Testing:** Testing each modification made to the original Online Python Tutor.
- **User Testing:** Testing with participants.

User Testing

- **Participants:**
  - CSC1010H and CSC1011H students.

- **Testing Procedures:**
  - Observe Navigation of users through Interactivity Feature and Visualisation Feature (to determine significance of the features and their understandability)
  - Survey Questions for each feature.

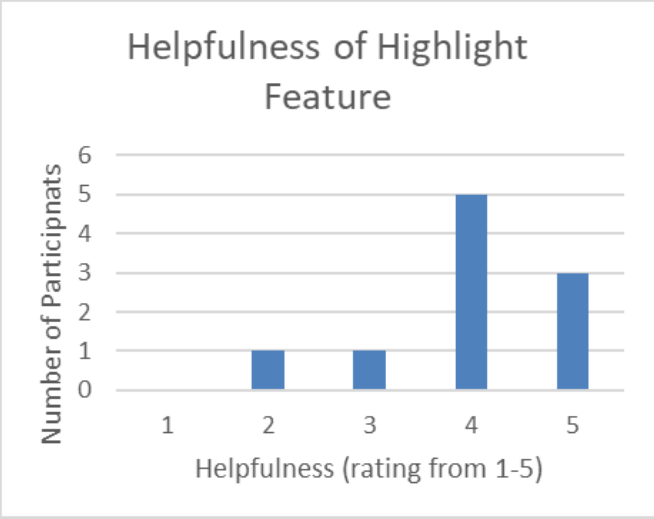
Findings: Observations

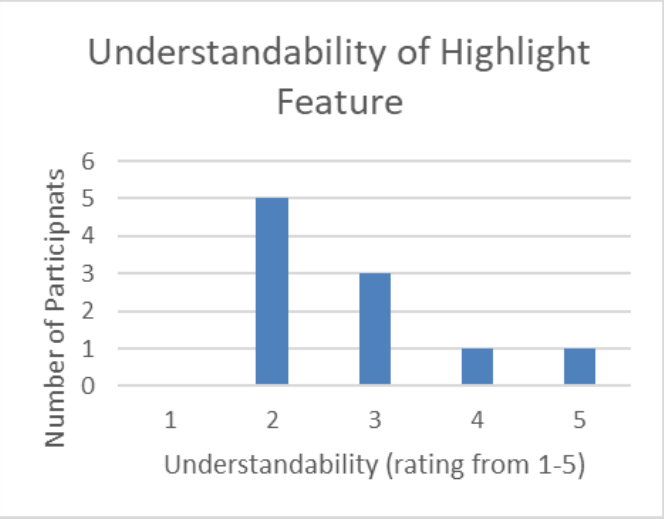
- Initial Instructions are required for the web application (tutorial video was suggested).
- In depth explanations were needed for the interactivity feature.
- Correlation between the Interactivity feature and the highlighting was not seen initially.

Findings: Survey

Highlighting Feature:

- Initially Difficult to Understand Feature.
- Once understood the helpfulness was seen.

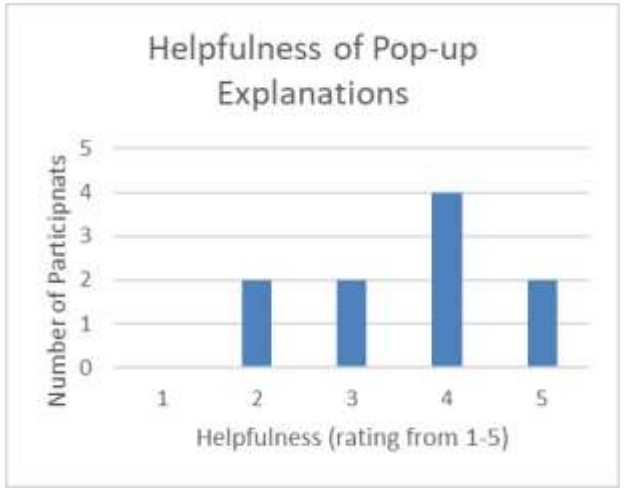


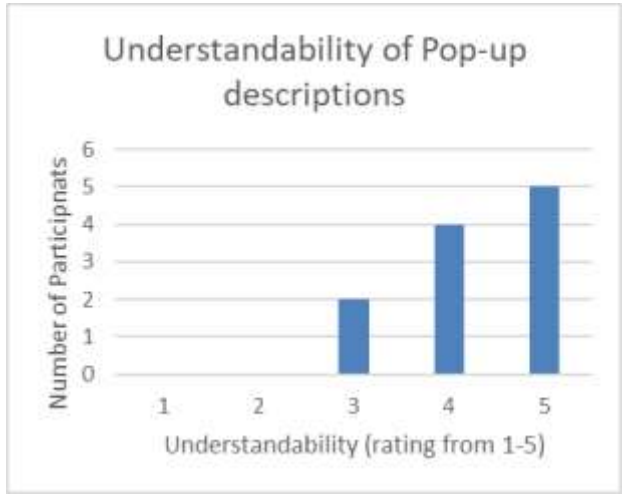


Findings: Survey

Description Pop-ups:

- Overall rating of helpfulness was due to the position of pop-ups, but that was later corrected.
- Overall rating was due to the understandability of the pop-ups (especially for beginners).

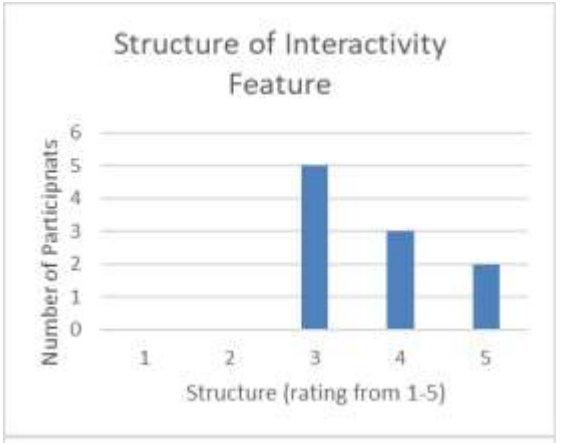


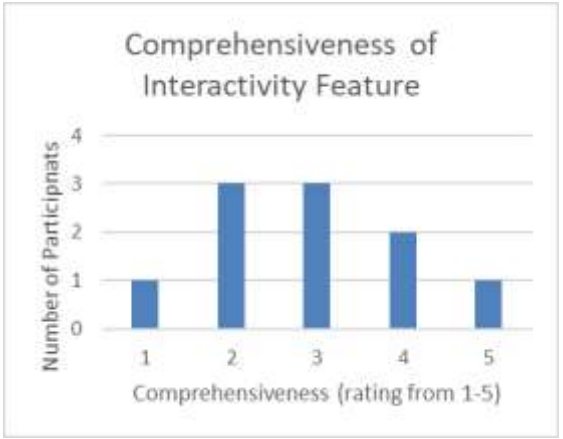


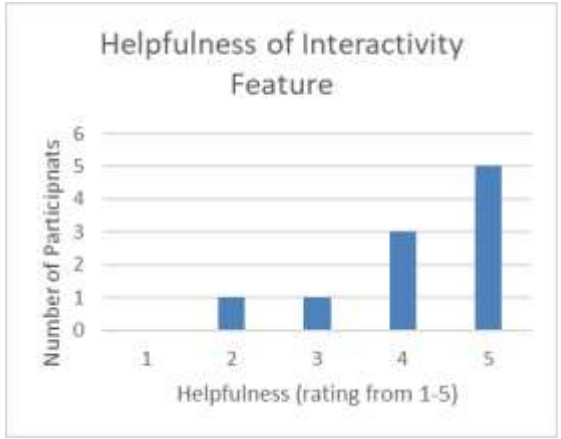
Findings: Survey

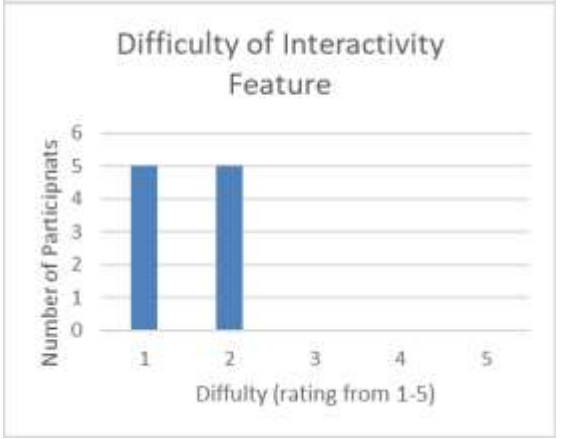
Interactivity Feature:

- Comprehensiveness: Initially difficult to understand from given explanation and instructions.
- Helpfulness: beneficial for checking Understanding and Debugging
- Difficulty: Better for beginners and code should be given.









Discussion

- **Effectiveness of Visualisation Features:**
  - Helpful
  - Highlight vs Description Pop Ups: Pops are more necessary proving need for comprehensive explanations of keywords.
- **Interactivity Feature:**
  - Useful for debugging but it was suggested that it would be helpful if codes were given to the users. Codes would be of level easy, medium and hard.
- Target users should change to high school students
- Interactivity is preferred to Visualisation.
- **Limitations:**
  - Input is not taken.
  - Only caters to Python users.

Conclusions

- Change in variables assists in determining the effect of each line of code.
- Improving Interactivity is more helpful than improving the visualisations.
- The enhancements are better for beginners, unlike the chosen participants who have had more that 6 months worth of experience.
- Target Users could include high school students.