

Testing Code

Textbook

Testing Code



Making sure a Python program gives the right answers and works well in all situations is really important. Validating the outputs and testing the robustness of a Python program are crucial steps to ensure its reliability and accuracy. Validation involves checking that the outputs produced by the program are correct and meet the expected requirements.

We will explain the following methods of testing by using the following code example.

```
1 candies = int(input("How many candies?"))
2
3 people = int(input("How many people?"))
4
5 print(candies/people)
```

Try it!

Boundary Conditions

These are the edges of what the program can handle. For example, if it works with numbers, we check how it handles really big or really small numbers.

For the following code, will it work if the user entered 4 billion candies? What about .000024 candies?

```
1 candies = int(input("How many candies?"))
2
3 people = 8
4
5 print(candies/people)
```

Try it!

If the input was `4,000,000` candies, the output would be `500,000`. So large numbers work fine.

If the input was `.000024`, the output would be `.000003`, so small numbers also work.

Invalid Inputs

Sometimes people might enter things that the program doesn't expect, like letters instead of numbers. The program needs to handle these situations without crashing or giving weird results.

For the following code, what would happen if the user entered something other than a number?

```
1 candies = int(input("How many candies?"))
2
3 people = 8
4
5 print(candies/people)
```

Try it!

If the input was `basketball`, this program would throw an error. So the programmer needs to write in an alert of some kind to tell the user to enter a number.

Unexpected Scenarios

This is about testing what happens when things happen that the program wasn't designed for. Like if it expects a list of numbers but gets a mix of numbers and words. We check how it reacts and if it can still do its job correctly.

Incorrect Results

We need to check if the program always gives the right answers. If it's supposed to calculate something, we compare what it gives us to what we expect. If they don't match, we know there might be a problem.

To do this, we use different kinds of tests. Some test small parts of the program by themselves (unit tests), others check how different parts work together (integration tests), and some test the whole program like it's being used for real (system tests).

Quality Assurance

By doing all this testing, we make sure the program works well in all situations and gives accurate results, which is really important for making sure it's reliable and does what we need it to do.

Automated Testing

Automated software testing significantly cuts down the cost of testing software. By using automated tests, especially with Python tools like `unittest` or `pytest`, you replace tasks that used to be done by hand. These automated tests run fast and reliably, even thousands of times, with almost no extra cost after they're set up. This means you find problems earlier in the development process, and fixing them then is much cheaper than fixing them later.

This efficiency allows human testers to spend their time on more complicated testing that needs human thought. The result is better quality software, fewer problems after the software is released, and getting the product to users faster. All these benefits lead to big cost savings over time.

Questions (8)

1. What is the primary objective of validating outputs in a Python program?

MULTIPLE CHOICE

Choose the correct answer:

- A. Shortening Code
- B. Checking code syntax
- C. Verifying correctness and meeting expected requirements
- D. Enhancing user interface

2. What are "boundary conditions" in testing a Python program?

MULTIPLE CHOICE

Choose the correct answer:

- A. Testing unexpected scenarios
- B. Testing different parts of the program
- C. Testing extremes of program inputs
- D. Testing integration of program modules

3. What should a Python program do if a user enters an invalid input?

MULTIPLE CHOICE

Choose the correct answer:

- A. Print a warning message and invite them to enter another input
- B. Ignore the input
- C. Continue running without any change
- D. Throw an error

4. What does "unexpected scenarios" refer to in testing a Python program?

MULTIPLE CHOICE

Choose the correct answer:

- A. Testing for incorrect results
- B. Testing integration of program modules
- C. Testing scenarios outside of normal usage
- D. Testing small parts of the program

5. When validating a Python program, why is checking for "incorrect results" important?

MULTIPLE CHOICE

Choose the correct answer:

- A. To enhance user interface
- B. To ensure the program always gives the right answers
- C. To test unexpected scenarios
- D. To check program syntax

6. What role does "quality assurance" play in Python programming?

MULTIPLE CHOICE

Choose the correct answer:

- A. Ensuring the program works well in all situations
- B. Improving program performance
- C. Enhancing user interface
- D. Checking for syntax errors

7. Why is testing for "invalid inputs" crucial in Python programming?

MULTIPLE CHOICE

Choose the correct answer:

- A. To test integration of program modules
- B. To handle unexpected scenarios
- C. To check for incorrect results
- D. To ensure the program handles user errors

8. What is the purpose of conducting "system tests" in Python programming?

Choose the correct answer:

- A. Testing boundary conditions
- B. Testing small parts of the program
- C. Testing unexpected scenarios
- D. Testing the entire program as it would be used in real life

Answer Keys & Solutions

Questions

1. What is the primary objective of validating outputs in a Python program?

MULTIPLE CHOICE

Correct Answer:

- | | |
|--|-------------|
| A. Shortening Code | ✗ Incorrect |
| B. Checking code syntax | ✗ Incorrect |
| C. Verifying correctness and meeting expected requirements | ✓ Correct |
| D. Enhancing user interface | ✗ Incorrect |

Explanation:

Testing assures that code works right.

2. What are "boundary conditions" in testing a Python program?

MULTIPLE CHOICE

Correct Answer:

- | | |
|---|-------------|
| A. Testing unexpected scenarios | ✗ Incorrect |
| B. Testing different parts of the program | ✗ Incorrect |
| C. Testing extremes of program inputs | ✓ Correct |
| D. Testing integration of program modules | ✗ Incorrect |

Explanation:

This includes really large and really small numbers.

3. What should a Python program do if a user enters an invalid input?

MULTIPLE CHOICE

Correct Answer:

A. Print a warning message and invite them to enter another input

✓ Correct

B. Ignore the input

✗ Incorrect

C. Continue running without any change

✗ Incorrect

D. Throw an error

✗ Incorrect

Explanation:

Programs should tell the user what they are supposed to do.

4. What does "unexpected scenarios" refer to in testing a Python program?

MULTIPLE CHOICE

Correct Answer:

A. Testing for incorrect results

✗ Incorrect

B. Testing integration of program modules

✗ Incorrect

C. Testing scenarios outside of normal usage

✓ Correct

D. Testing small parts of the program

✗ Incorrect

Explanation:

Users will enter all kinds of inputs, so we need to test for that.

5. When validating a Python program, why is checking for "incorrect results" important?

MULTIPLE CHOICE

Correct Answer:

A. To enhance user interface

✗ Incorrect

B. To ensure the program always gives the right answers

✓ Correct

C. To test unexpected scenarios

✗ Incorrect

D. To check program syntax

✗ Incorrect

Explanation:

A program should produce the correct results.

6. What role does "quality assurance" play in Python programming?

MULTIPLE CHOICE

Correct Answer:

- | | |
|--|-------------|
| A. Ensuring the program works well in all situations | ✓ Correct |
| B. Improving program performance | ✗ Incorrect |
| C. Enhancing user interface | ✗ Incorrect |
| D. Checking for syntax errors | ✗ Incorrect |

Explanation:

The best programs work well all the time

7. Why is testing for "invalid inputs" crucial in Python programming?

MULTIPLE CHOICE

Correct Answer:

- | | |
|--|-------------|
| A. To test integration of program modules | ✗ Incorrect |
| B. To handle unexpected scenarios | ✗ Incorrect |
| C. To check for incorrect results | ✗ Incorrect |
| D. To ensure the program handles user errors | ✓ Correct |

Explanation:

The user could sometimes make a mistake

8. What is the purpose of conducting "system tests" in Python programming?

MULTIPLE CHOICE

Correct Answer:

- | | |
|--------------------------------|-------------|
| A. Testing boundary conditions | ✗ Incorrect |
|--------------------------------|-------------|

B. Testing small parts of the program

✗ Incorrect

C. Testing unexpected scenarios

✗ Incorrect

D. Testing the entire program as it would be used in real life

✓ Correct

Explanation:

Systems tests make sure programs run well in relation to other systems