

Input/Output

Textbook

Input/Output

Important!!!

As you go through the lessons, the checkpoints and challenges will be checking to see if your print statements match the requirements EXACTLY. It needs to match capitalization, punctuation, and spacing.

That being said, the checkpoints and challenges will be reading ALL of your print statements and checking to see if they match. In your program, if the requirements are asking for a *specific* output, with an input example and an output example, make sure that you include ONLY the print statement that you want to submit for your output answer.

Input

Writing programs that can get input from the user is a fundamental skill for any programmer. Imagine playing a video game that didn't use your input! You wouldn't be able to play it.



There are many different types of input that Python can work with. When we receive input from a user, we always assign it to a variable.

Getting Input

Let's say we want the user to input their first name into our program so we know what to call them. To do this, we create a variable called "first_name" and assign the input to that variable, as you can see below.

```
first_name = input("What is your first name?")
```

This code will ask the user for their first name, and pause the program until the user types in their name and presses "Enter."

Output

A print statement is a kind of output. An output is something the computer generates from information it received. We put in an input, the computer does computations and generates an output. There are many kinds of outputs in programming, but we'll focus on print statements for now.

Now that we know how to get input from the user, we can use it to send a response. To do this, we need to learn how to combine variables and text in `print` statements.

```
1 favorite_animal = input('What is your favorite animal?')
2
3 print(favorite_animal)
```

Here is another example

```
1 color = input('What is your favorite color?')
2
3 print(color)
```

Try it!

Important Note: The input question and answer will show in the console even though a print statement wasn't used. This is how Python terminals work in the industry. The autograder that looks for certain outputs only looks for print statements specifically. Only items printed from a print statement will be considered.

White Space

Good use of white space in a Python program makes the code easier to read and understand. Python uses indentation to show where code blocks start and end, like in loops or functions, so proper white space is important for the program to work correctly. Leaving spaces between functions or different parts of the code helps separate them, making the code cleaner and easier to follow. Using white space well helps keep the code organized and simple to change later.

Making Bigger Sentences

You can combine variables and text in `print` statements by using a "+" sign between them. See the code below for an example, and assume that the user typed in "blue" for the input.

```
1 color = input('What is your favorite color?')
2
3 print("My favorite color is " + color)
```

Try it!

Notice that we had to add a space after the word `is` to make sure there is a space in the print statement.

Now let's try adding an exclamation point

```
1 favorite_animal = input('What is your favorite animal?')
2
3 print('You said your favorite animal is a ' + favorite_animal + '!!')
```

As you can see, this code combines text in your print statement with the variable typed in by the user. Combining these two types of code is called concatenation, which you'll learn more about later.

Clear Prompts

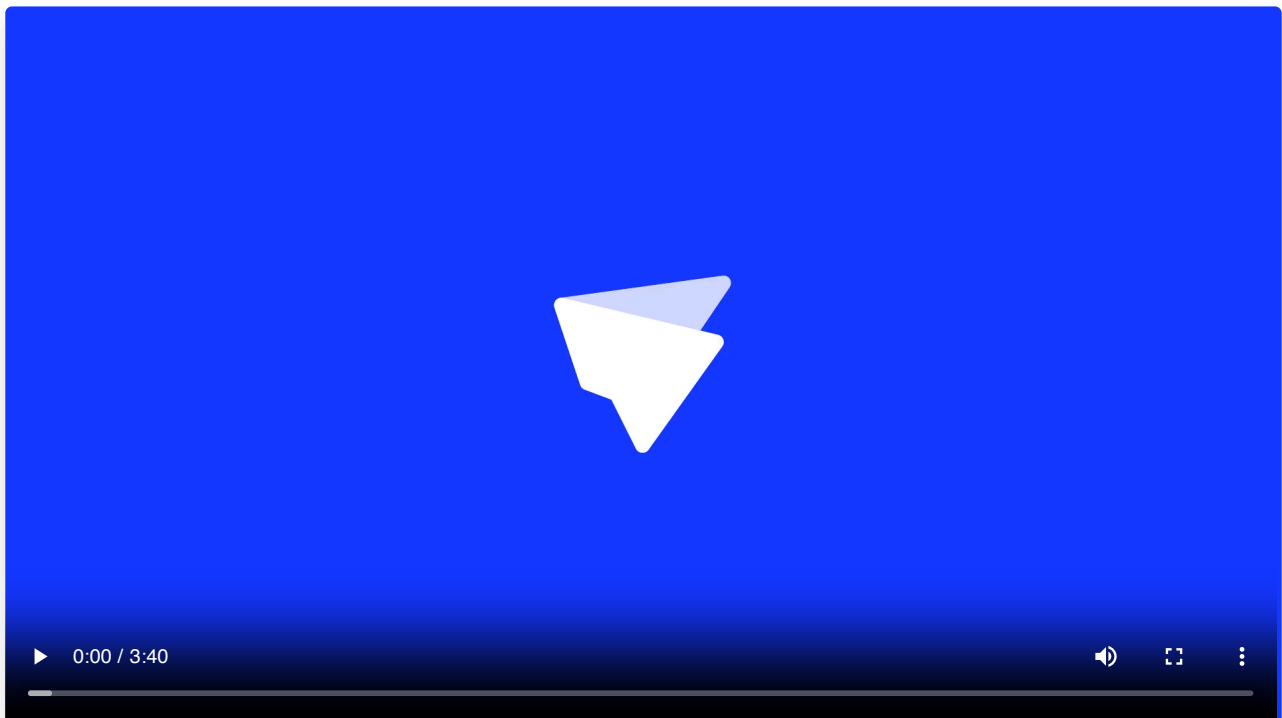
User prompts are the instructions or questions a program presents to the user, guiding them on what input is expected. For a program to be effective and user-friendly, these prompts **must be clear and promote usability**, ensuring the user understands exactly what to do and how to do it correctly. A well-defined prompt prevents errors and frustration, making the program intuitive and efficient to use.

Example: Correcting an Age Verification Prompt

Paolo's age verification application crashes when users enter their birth month incorrectly (e.g., typing out "January" instead of a number). To correct this, a clear and usable user prompt should guide the user to the expected input format.

Original, problematic prompt (implied): "Enter your birth month:" **Corrected, clear, and usable prompt:** "Please enter your birth month as a number (e.g., 1 for January, 2 for February):"

This revised prompt explicitly states the *expected format* (a number) and provides *examples* for clarity, preventing the program from crashing due to unexpected text input and significantly improving the user experience.



Checkpoint

Print the Input

Write a program that asks the user what their favorite color is.

Print their favorite color in a sentence that says `Your favorite color is` + their inputted color.

Match the output EXACTLY! Spacing, capitalization, and punctuation all matter. Make sure to include a space after the word "is".

For example:

Input: `blue`

Output: `Your favorite color is blue`

Another example:

Input: `red`

Output: `Your favorite color is red`

READ THIS! Important information about the autograder!

As you know, the autograder is very detail oriented. If the challenge or checkpoint asks for a specific **input** and a specific **output**, the method of grading is a bit different. The autograder is checking to see that your program will present the correct output with certain inputs.

Follow these tips and tricks to make sure your correct code can pass the autograder.

-If the program is looking for a specific output, make sure you have only **one** print statement! The auto grader is going to be looking at all the print statements on your code page, so make sure you just have one to represent your answer!

-The autograder will also be checking for the required answer EXACTLY. Make sure to match capitalization, spacing, and punctuation. If you miss a period or a space, it won't pass. This helps to show just how detail-oriented coding is.

-Keep in mind that just because the example input is generating the correct output, it doesn't mean your code works for other inputs. Make sure a variety of inputs follow the requirements of the challenge or checkpoint in order to pass.

Questions (9)

1. Which line of code correctly asks the user for their name and assigns it to a variable?

MULTIPLE CHOICE

Choose the correct answer:

- A. `name = input "What's your name?"`
- B. `name = input("What's your name?")`
- C. `name = input("What's your name?")`
- D. `name = input("What's your name?)`

2. How do you combine text and a variable in a print() statement?

MULTIPLE CHOICE

Choose the correct answer:

- A. Use the "+" sign.
- B. Include the variable in the quotation marks.
- C. Put the variable outside the print() statement.
- D. Put the variable before the print() statement.

MULTIPLE CHOICE

3. What key does the user need to press after typing in their input?

Choose the correct answer:

- A. +
- B. =
- C. Enter
- D. -

MULTIPLE CHOICE

4. True or False: A print statement is a kind of output.

Choose the correct answer:

- A. True
- B. False

DEBUG CODE

5. Edit the text box below to debug (fix) the code:

Code to Debug:

```
1 age = input "How old are you?"
```

DEBUG CODE

6. Debug the following code to create an input:

Code to Debug:

```
1 animal = "What is your favorite animal?"
```

DEBUG CODE

7. Edit the text box below to debug (fix) the code:

Code to Debug:

```
1 height = input(How tall are you?)
```

8. What will the following code snippet output if the user inputs "cat" for favorite_animal?

MULTIPLE CHOICE

```
favorite_animal = input('What is your favorite animal?') print('You said your favorite animal is a ' + favorite_animal + '!')
```

Choose the correct answer:

- A. You said your favorite animal is a cat
- B. You said your favorite animal is a cat?
- C. You said your favorite animal is a cat!
- D. You said your favorite animal is a cat.

9. How do you assign user input to a variable in Python?

MULTIPLE CHOICE

Choose the correct answer:

- A. `create_variable = input("prompt")`
- B. `input("prompt") = create_variable`
- C. `create_variable("prompt")`
- D. `input(create_variable)`

Challenges (5)

1. Hello, Harry!

Write a program that asks a user for their first name, then prints out `Hello,` and their first name.

For example:

Input: `Harry`

Output: `Hello, Harry!`

Another example:

Input: `Jasmine`

Output: `Hello, Jasmine!`

Don't forget to match the output EXACTLY! Spacing, capitalization, and punctuation all matter. One space after a comma.

Don't forget to include the **exclamation point**.

READ THIS! Important information about the autograder!

As you know, the autograder is very detail oriented. If the challenge or checkpoint asks for a specific **input** and a specific **output**, the method of grading is a bit different. The autograder is checking to see that your program will present the correct output with certain inputs.

Follow these tips and tricks to make sure your correct code can pass the autograder.

-If the program is looking for a specific output, make sure you have only **one** print statement! The auto grader is going to be looking at all the print statements on your code page, so make sure you just have one to represent your answer!

-The autograder will also be checking for the required answer EXACTLY. Make sure to match capitalization, spacing, and punctuation. If you miss a period or a space, it won't pass. This helps to show just how detail oriented coding is.

-Keep in mind that just because the example input is generating the correct output, it doesn't mean your code works for other inputs. Make sure a variety of inputs follow the requirements of the challenge or checkpoint in order to pass.

2. Ice Cream Order

What is your all time favorite way to eat ice cream? Do you like it in a bowl or on a cone? With fruit or chocolate? Do you like toppings?

1. Create a variable named `flavor` and assign it to an input that asks the user for the ice cream flavor.
2. Create a second variable named `mix_in` and assign it to an input that asks the user for their mix in.
3. Create a third variable named `topping` and assign it to an input that asks the user for their topping.
4. Print out each variable in a separate print statement.

Requirements:

- Create a variable named `flavor` and assign it to an input that asks the user for the ice cream flavor.
- Create a second variable named `mix_in` and assign it to an input that asks the user for their mix in.
- Create a third variable named `topping` and assign it to an input that asks the user for their topping.
- Print out each variable in a separate print statement.

3. Compliments

What kind of compliment makes you feel good? Create a program that gives compliments!

Create **2 inputs**. 1 input will ask for your name. 1 input will ask for your favorite compliment.

Create an **output** that uses the inputs in a way that will compliment the user in the following format.

Make sure you use concatenation for your inputs with the words `"Hello"` and `"I would like to give you a compliment."`

For example:

Inputs: `James` , `You are so kind`

Output: `Hello James, I would like to give you a compliment. You are so kind`

Another example:

Inputs: `Holly` , `You work so hard`

Output: `Hello Holly, I would like to give you a compliment. You work so hard`

Don't forget the period after compliment and then a single space after the period and before the next sentence!

Make sure to concatenate the extra words to the outputs by using the `+` symbol!

4. Guitar Class

You are signing up for an advanced guitar class. As you are signing up for the class, they ask you some questions about your music experience so they can put you in the correct class.

Create a program that asks for your name as an input and asks what kind of music experience you have.

The program will then print the output in the following format in one print statement.

Inputs: `George` , `I played the piano for 2 years` .

Output: `Hello George welcome to the guitar class! Your music experience: I played the piano for 2 years` .

Another example:

Inputs: `Natalie` , `I have been a singer my whole life` .

Output: `Hello Natalie welcome to the guitar class! Your music experience: I have been a singer my whole life` .

Don't forget to match the output exactly! Check capitalization, spelling, punctuation, and spacing. One space after the end of a sentence.

5. About Me!

Write a program that asks the user **five** different questions about themselves in input statements.

Some questions you should ask are:

What do you like to do?

What's your favorite movie?

When is the BEST time of day?

What pet peeves do you have?

When they've answered the questions, choose your favorite answer and print it out.

Requirements:

- Ask the user 5 questions
- Assign each answer to a unique variable.
- Choose an answer and print it out.

Answer Keys & Solutions

Checkpoint Solutions

Print the Input

```
1 favorite_color = input("What is your favorite color?")
2
3 print("Your favorite color is " + favorite_color)
```

Questions

1. Which line of code correctly asks the user for their name and assigns it to a variable?

MULTIPLE CHOICE

Correct Answer:

- A. name = input "What's your name?" ✗ Incorrect
- B. name = input("What's your name?" ✗ Incorrect
- C. name = input("What's your name?") ✓ Correct
- D. name = input("What's your name?) ✗ Incorrect

Explanation:

Opening and closing quotation marks and parentheses.

2. How do you combine text and a variable in a print() statement?

MULTIPLE CHOICE

Correct Answer:

- A. Use the "+" sign. ✓ Correct
- B. Include the variable in the quotation marks. ✗ Incorrect
- C. Put the variable outside the print() statement. ✗ Incorrect
- D. Put the variable before the print() statement. ✗ Incorrect

Explanation:

It's similar to "adding" them together.

3. What key does the user need to press after typing in their input?

MULTIPLE CHOICE

Correct Answer:

A. +

✗ Incorrect

B. =

✗ Incorrect

C. Enter

✓ Correct

D. -

✗ Incorrect

Explanation:

It's kind of like submitting your answer.

4. True or False: A print statement is a kind of output.

MULTIPLE CHOICE

Correct Answer:

A. True

✓ Correct

B. False

✗ Incorrect

Explanation:

Outputs are ways to see or use variables

5. Edit the text box below to debug (fix) the code:

DEBUG CODE

Incorrect Code:

```
1 age = input "How old are you?"
```

Correct Solution:

```
1 age = input("How old are you?")
```

Explanation:

Input statements use parentheses.

6. Debug the following code to create an input:

DEBUG CODE

Incorrect Code:

```
1 animal = "What is your favorite animal?"
```

Correct Solution:

```
1 animal = input("What is your favorite animal?")
```

Explanation:

Input statements use parentheses.

7. Edit the text box below to debug (fix) the code:

DEBUG CODE

Incorrect Code:

```
1 height = input(How tall are you?)
```

Correct Solution:

```
1 height = input("How tall are you?")
```

Explanation:

Input statements use parentheses and quotation marks.

8. What will the following code snippet output if the user inputs "cat" for favorite_animal?

MULTIPLE CHOICE

Correct Answer:

A. You said your favorite animal is a cat

✗ Incorrect

B. You said your favorite animal is a cat?

✗ Incorrect

C. You said your favorite animal is a cat!

✓ Correct

D. You said your favorite animal is a cat.

✗ Incorrect

Explanation:

Which punctuation is concatenated here?

9. How do you assign user input to a variable in Python?

MULTIPLE CHOICE

Correct Answer:

- | | |
|---|-------------|
| A. <code>create_variable = input("prompt")</code> | ✓ Correct |
| B. <code>input("prompt") = create_variable</code> | ✗ Incorrect |
| C. <code>create_variable("prompt")</code> | ✗ Incorrect |
| D. <code>input(create_variable)</code> | ✗ Incorrect |

Explanation:

An equals sign is needed for creating a variable

Challenges

1. Hello, Harry!

Solution:

```
1 name = input("What is your first name?")
2
3 print("Hello, " + name + "!")
```

2. Ice Cream Order

Solution:

```
1 flavor = input("What flavor of ice cream do you want?")
2
3 mix_in = input("What kind of mix-in do you want?")
4
5 topping = input("What kind of topping do you want?")
6
7 print(flavor)
8 print(mix_in)
9 print(topping)
```

3. Compliments

Solution:

```
1 name = input("What is your name?")
2 compliment = input("What is your favorite compliment?")
3
4 print("Hello " + name + " I would like to give you a compliment. " + compliment)
```

4. Guitar Class

Solution:

```
1 name = input("What is your name?")
2 experience = input("What is your music experience?")
3
4 print("Hello " + name + " welcome to the guitar class! Your music experience: " + experience)
```

5. About Me!

Solution:

```
1 question1 = input("What is your dream?")
2
3 question2 = input("What do you want to be when you grow up?")
4
5 question3 = input("What is your all time favorite thing to do?")
6
7 question4 = input("Where do you want to live?")
8
9 question5 = input("What is your dream pet?")
10
11 print(question1)
```