

## The Print Statement & Variables

---

### Textbook

---

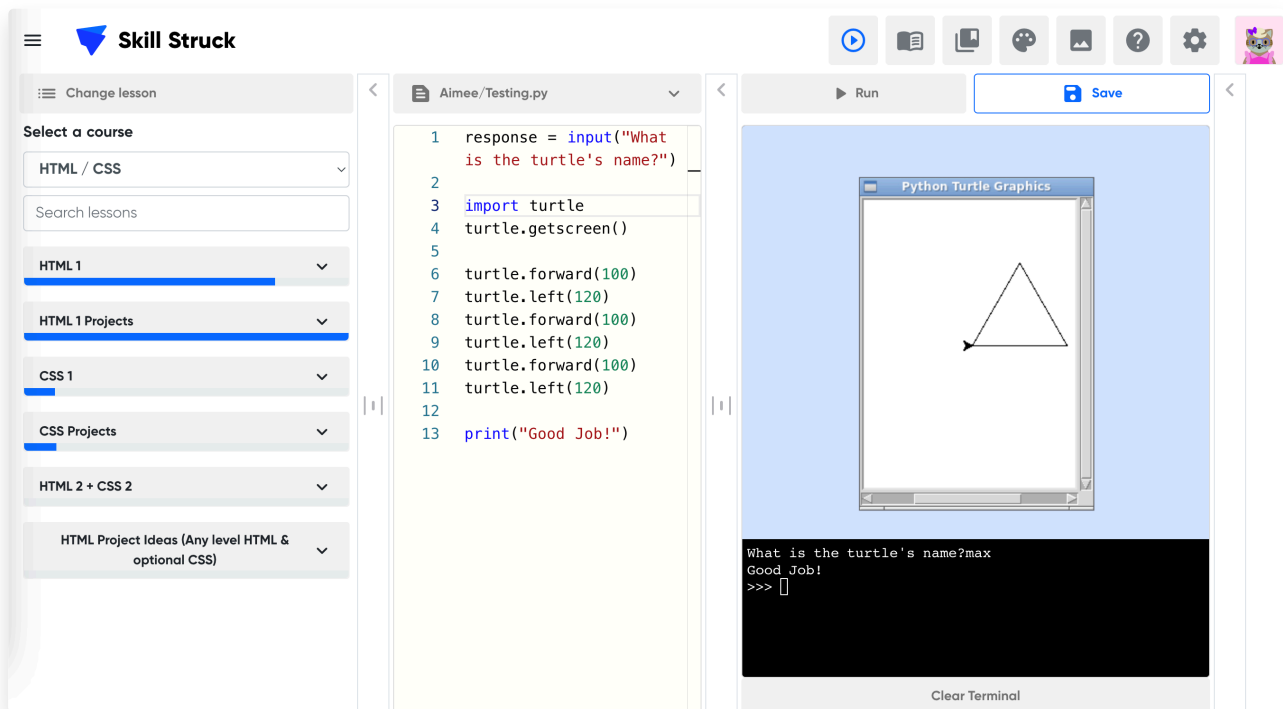
## The Print Statement & Variables



We can also print information out by using Python code! When we print something in Python, it gets written down in a notebook called a console.

### The Console

The [console](#) is like a notebook that only the programmer sees. It's a place for the programmer to take notes and make sure the program is working correctly. The console can be found on the right of the Skill Struck Screen below where the turtle screen shows.



The black box in the bottom right of the screen is showing the console.

The console also shows errors and other helpful information for the programmer.

## Print a String

Now let's print out a string! In python, words or phrases are called [strings](#). A string is a word or phrase surrounded by quotation marks.

Here are 3 examples of strings:

```
"This is a string"
```

```
"This is also a string."
```

```
"Hello"
```

Now let's print these strings out! We do this by using a print statement that looks like this:

```
print( )
```

Whatever you want to print is put inside the parentheses.

```
1 import turtle
2 turtle.getscreen()
3
4 print("Good job turtle!")
```

This will then print `Good job turtle!` to the console.

## Print a Variable

We can also print out variables.

Remember that [variables](#) are like an empty box. We can put whatever we want inside.

```
1 import turtle
2 turtle.getscreen()
```

```
3
4 my_name = "Jane"
5 print(my_name)
```

In this example we printed out the variable named `my_name` , which is holding the string `Jane` .

So `Jane` is printed to the console.

## Print Both a String and a Variable

Here's an example where we print both a string and a variable:

```
1 import turtle
2 turtle.getscreen()
3
4 my_name = "Jane"
5 print(my_name)
6 print("I love the color yellow.")
```

In this example, we print the variable named `name` and a string. Notice that when you print a variable, you don't use quotation marks. The quotation marks are for printing a new string. Remember that [strings](#) are surrounded by quotation marks.

## Rules for Naming Variables

Here's a reminder for the rules for naming variables.

When deciding what to name your variables, try to name them something that refers to what it's doing. It is [best practice](#) to name your variables something that refers to what the variable will hold.

Here are a few more rules when it comes to naming variables.

- Variables cannot be more than one word long (We get around this rule by adding an underscore character `_` to join multiple words together. For example `first_name` is a valid variable name. This is called snake case.
- A variable name must start with a letter or an underscore character `_` .
- A variable name cannot start with a number.
- A variable name can only use capital A-Z, lowercase a-z, numbers 0-9, and the underscore character `_` . No special characters are allowed.

## Your Turn to Be a Detective!

Sometimes, to truly understand a problem or a story, you need to look beyond just what's directly stated. Just like a detective uses clues to figure out what happened, you can use the information you have to figure out what's implied or what might happen next.

Try these questions to sharpen your detective skills using what you've learned about printing and variables:

- You see the code `print(message)` and then in the console, "Hello, World!" appears. What do you think is stored inside the `message` variable?
- If a program has `user_input = "25"` and later `print(user_input * 2)` shows "2525" in the console, what does this tell you about the type of information stored in `user_input` ?

- You are looking at a program's console and see an error message you've never encountered before. What is the first step you would take to try and understand what the error means?
- If you wanted to print out someone's age and their favorite color, and you already have both pieces of information stored in variables, how would you combine a string and these variables in a single print statement to make a clear message?

## Checkpoint

---

### The Print Statement & Variables

Practice printing information to the console.

1. Include the necessary code to start up a Python screen (import the library and generate a screen).
2. Create a variable named `name` and assign it to the string `"Ajash"`.
3. Create a variable named `color` and assign it to the string `"blue"`.
4. Print the variable named `name`.
5. Print the variable named `color`.
6. Print the string `"Today is cloudy."`
7. Print the string `"Thanks for coming!"`

### Requirements:

- Include the necessary code to start up a Python screen (import the library and generate a screen).
- Create a variable named `name` and assign it to the string `"Ajash"`.
- Create a variable named `color` and assign it to the string `"blue"`.
- Print the variable named `name`.
- Print the variable named `color`.
- Print the string `"Today is cloudy."`
- Print the string `"Thanks for coming!"`

## Questions (10)

---

### 1. What is the primary purpose of the console in Python programming?

MULTIPLE CHOICE

Choose the correct answer:

- A. Displaying output to the end user.
- B. Providing a space for programmers to take notes.
- C. Storing variables for later use.
- D. Running Python code in real-time.

**2. Where can the console be found on the Skill Struck Screen.**

**Choose the correct answer:**

- A. Left side, top.
- B. Left side, bottom.
- C. Right side, top.
- D. Right side, bottom.

**3. What is a string in Python?**

**Choose the correct answer:**

- A. A numerical variable.
- B. A word or phrase enclosed in quotation marks.
- C. A reserved keyword in Python.
- D. A special character used in printing.

**4. Which is the correct way to write a print statement in Python?**

**Choose the correct answer:**

- A. `print(" ")`
- B. `console.print()`
- C. `echo(" ")`
- D. `display.print(" ")`

**5. Consider the following code, what would actually display in the console?**

```
print("Good Morning")
```

**Choose the correct answer:**

- A. "Good Morning".
- B. Good Morning
- C. ("Good Morning.")
- D. `print("Good Morning.")`

**6. Consider the following code, what will actually display in the console?**

```
my_name = "Rio" print(my_name)
```

**Choose the correct answer:**

- A. my\_name
- B. "Rio"
- C. Rio
- D. "my\_name"

**7. Which of the following is NOT a rule for naming variables in Python?****Choose the correct answer:**

- A. Start with a letter or underscore.
- B. Not more than one word separated by spaces.
- C. Using only A-Z, a-z, 0-9, and underscore.
- D. Start with a number.

**8. According to best practices, what should be considered when naming variables in Python?****Choose the correct answer:**

- A. Use special characters for uniqueness.
- B. Keep it unrelated to its purpose.
- C. Choose names that reflect the variable's purpose.
- D. Start with a number for clarity.

**9. How is the rule of variables not being more than one word addressed in Python?****Choose the correct answer:**

- A. By using hyphens between words.
- B. By using underscores between words.
- C. By adding spaces between words.
- D. Only use one word for variable names.

## 10. Debug the following code:

DEBUG CODE

### Code to Debug:

```
1 greeting = "Good Afternoon!"
2
3 print(greeting)
```

## Challenges (4)

### 1. I Love Coding

Practice generating print statements.

1. Include the necessary code to start up a Python screen (import the library and generate a screen).
2. Create 5 print statements on separate lines of code.
3. The first print statement will say `I` .
4. The next print statement will say `love` .
5. The next print statement will say `coding` .
6. The next print statement will say `with` .
7. The next print statement will say `turtles!` .

#### Requirements:

- Include the necessary code to start up a Python screen (import the library and generate a screen).
- Create 5 print statements on separate lines of code.
- The first print statement will say `I`
- The next print statement will say `love`
- The next print statement will say `coding`
- The next print statement will say `with`
- The next print statement will say `turtles!`

### 2. Super Long Name

Practice creating a variable with a really long name!

1. Include the necessary code to start up a Python screen (import the library and generate a screen)
2. Create a variable with a name that has at least 5 underscores and assign it to a string that says `"hey"`
3. Print the variable

#### Requirements:

- Include the necessary code to start up a Python screen (import the library and generate a screen)
- Create a variable with a name that has at least 5 underscores and assign it to a string that says `"hey"`
- Print the variable

### 3. Favorites

Practice creating and printing variables of your favorite things!

1. Include the necessary code to start up a Python screen (import the library and generate a screen).
2. Create a variable named `hobby` and assign it to a string.
3. Create a variable named `sport` and assign it to a string.
4. Create a variable named `food` and assign it to a string.
5. Print the variable named `hobby`.
6. On a separate line of code, print the variable named `sport`.
7. On a separate line of code, print the variable named `food`.

#### Requirements:

- Include the necessary code to start up a Python screen (import the library and generate a screen).
- Create a variable named `hobby` and assign it to a string.
- Create a variable named `sport` and assign it to a string.
- Create a variable named `food` and assign it to a string.
- Print the variable named `hobby`.
- On a separate line of code, print the variable named `sport`.
- On a separate line of code, print the variable named `food`.

### 4. Cheering On Your Turtle

Practice creating print statements. Your print statements will cheer on your turtle.

1. Include the necessary code to start up a Python screen (import the library and generate a screen).
2. Move your turtle forward, turn once, then forward again.
3. Create 3 print statements on separate lines of code.
4. The first print statement should say `Good job turtle!`
5. The second print statement should say `You rock!`
6. The third print statement should say `You can do it!`

#### Requirements:

- Include the necessary code to start up a Python screen (import the library and generate a screen).
- Move your turtle forward, turn once, then forward again.
- Create 3 print statements on separate lines of code.
- The first print statement should say `Good job turtle!`
- The second print statement should say `You rock!`
- The third print statement should say `You can do it!`



## Answer Keys & Solutions

### Checkpoint Solutions

#### The Print Statement & Variables

```
1 import turtle
2 turtle.getscreen()
3
4 name = "Ajash"
5 color = "blue"
6
7 print(name)
8 print(color)
9 print("Today is cloudy.")
10 print("Thanks for coming!")
```

### Questions

#### 1. What is the primary purpose of the console in Python programming?

MULTIPLE CHOICE

##### Correct Answer:

- A. Displaying output to the end user. ✗ Incorrect
- B. Providing a space for programmers to take notes. ✓ Correct
- C. Storing variables for later use. ✗ Incorrect
- D. Running Python code in real-time. ✗ Incorrect

##### Explanation:

The console is for the programmer to see, not the user

#### 2. Where can the console be found on the Skill Struck Screen.

MULTIPLE CHOICE

##### Correct Answer:

- A. Left side, top. ✗ Incorrect
- B. Left side, bottom. ✗ Incorrect

C. Right side, top.

✗ Incorrect

D. Right side, bottom.

✓ Correct

#### Explanation:

It's the black area on the right side of the screen.

### 3. What is a string in Python?

MULTIPLE CHOICE

#### Correct Answer:

A. A numerical variable.

✗ Incorrect

B. A word or phrase enclosed in quotation marks.

✓ Correct

C. A reserved keyword in Python.

✗ Incorrect

D. A special character used in printing.

✗ Incorrect

#### Explanation:

This is a string "Hello There"

### 4. Which is the correct way to write a print statement in Python?

MULTIPLE CHOICE

#### Correct Answer:

A. `print(" ")`

✓ Correct

B. `console.print()`

✗ Incorrect

C. `echo(" ")`

✗ Incorrect

D. `display.print(" ")`

✗ Incorrect

#### Explanation:

It's a simple command without a period.

### 5. Consider the following code, what would actually display in the console?

MULTIPLE CHOICE

### Correct Answer:

- A. "Good Morning". ✗ Incorrect
- B. Good Morning ✓ Correct
- C. ("Good Morning.") ✗ Incorrect
- D. print("Good Morning.") ✗ Incorrect

### Explanation:

Only the value inside the string will print.

## 6. Consider the following code, what will actually display in the console?

MULTIPLE CHOICE

### Correct Answer:

- A. my\_name ✗ Incorrect
- B. "Rio" ✗ Incorrect
- C. Rio ✓ Correct
- D. "my\_name" ✗ Incorrect

### Explanation:

The variable value will print

## 7. Which of the following is NOT a rule for naming variables in Python?

MULTIPLE CHOICE

### Correct Answer:

- A. Start with a letter or underscore. ✗ Incorrect
- B. Not more than one word separated by spaces. ✗ Incorrect
- C. Using only A-Z, a-z, 0-9, and underscore. ✗ Incorrect
- D. Start with a number. ✓ Correct

### Explanation:

You cannot begin your variable name with a number.

## 8. According to best practices, what should be considered when naming variables in Python?

MULTIPLE CHOICE

**Correct Answer:**

- A. Use special characters for uniqueness. ✗ Incorrect
- B. Keep it unrelated to its purpose. ✗ Incorrect
- C. Choose names that reflect the variable's purpose. ✓ Correct
- D. Start with a number for clarity. ✗ Incorrect

### Explanation:

Variable names that help describe what the variable does are best.

## 9. How is the rule of variables not being more than one word addressed in Python?

MULTIPLE CHOICE

**Correct Answer:**

- A. By using hyphens between words. ✗ Incorrect
- B. By using underscores between words. ✓ Correct
- C. By adding spaces between words. ✗ Incorrect
- D. Only use one word for variable names. ✗ Incorrect

### Explanation:

Here's an example of a valid variable name `first_example`

## 10. Debug the following code:

DEBUG CODE

**Incorrect Code:**

```
1 greeting = "Good Afternoon!"  
2
```

```
3 print(greeting)
```

**Correct Solution:**

```
1 greeting = "Good Afternoon!"  
2  
3 print(greeting)
```

## Challenges

### 1. I Love Coding

**Solution:**

```
1 import turtle  
2 turtle.getscreen()  
3  
4 print("I")  
5 print("love")  
6 print("coding")  
7 print("with")  
8 print("turtles!")
```

### 2. Super Long Name

**Solution:**

```
1 import turtle  
2 turtle.getscreen()  
3  
4 here_is_my_very_long_variable_name = "Hey"  
5  
6 print(here_is_my_very_long_variable_name)
```

### 3. Favorites

**Solution:**

```
1 import turtle  
2 turtle.getscreen()  
3  
4 hobby = "dancing"  
5 sport = "soccer"  
6 food = "cereal"  
7  
8 print(hobby)  
9 print(sport)  
10 print(food)
```

## 4. Cheering On Your Turtle

**Solution:**

```
1 import turtle
2 turtle.getscreen()
3
4 turtle.forward(50)
5 turtle.left(90)
6 turtle.forward(100)
7
8
9 print("Good job turtle!")
10
11 print("You rock!")
12
13 print("You can do it!")
```