

Strings

Textbook

Strings

Now that you've had some experience with data types, it's time to learn a little more about [strings](#).



A string is a piece of information with quotation marks around it. Here is an example of a variable named `pet` that is set equal to a string that says `dog`.

```
pet = "dog"
```

A string can be declared or created using either single or double quotes. Both work equally well.

```
pet = 'dog'
```

```
pet = "dog"
```

Strings can also be many words long.

```
motto = "Always keep trying no matter what."
```

Strings can also have numbers in them.

```
candy = "I have 10 chocolate candies and 13 strawberry candies."
```

Strings can also just have numbers in them.

```
age = "15"
```

This may look like an integer data type, but it's actually a string.

IMPORTANT NOTE! Strings can look a lot like numbers, like in the example above! But when we try to do math with a string, we run into problems. This is why converting between data types is important! Review the lesson on converting if you need a refresher on converting between data types.

Strings can also be called [string literals](#).

Adding Quotes to Your String

But what if you want to include a quotation mark in your string? Perhaps you are quoting someone. Your code might look like this.

```
1 quote = "And the president said "Welcome""
2 print(quote)
```

Try it!

This will throw an error.

Or what if you want to include an apostrophe and use single quotes for your string?

```
1 idea = 'Let's go to the store.'
2 print(idea)
```

Try it!

This will also throw an error.

This is where we use an escape backslash. \

This key can be found above your "return" key. If you include this backslash key right before the symbol you want to include in the string, the Python code will not consider it the end of a string.

So the quote example would look like this.

```
1 quote = "And the president said \"Welcome\""
2 print(quote)
```

Try it!

And the apostrophe example would look like this.

```
1 idea = 'Let\'s go to the store.'
2 print(idea)
```

Try it!

Alternate Method

Another way to include quotes in your print statement is to just alternate the use of single and double quotes.

```
1 quote = 'And the president said "Welcome"'
2 print(quote)
```

Try it!

End = Statement

By default, Python prints each print statement on a new line.

```
1 print("Good morning")
2 print("Friends")
```

Try it!

The default behavior is for this to print two statements on a separate line.

Good morning

Friends

If you want these to print on the same line, you can use the `end=` statement.

The `end=` statement tells the next print statement to attach onto the same line.

```
1 print("Good Morning", end=' ')
2 print("Friends")
```

Try it!

*Note: You must include a **space** between the quotes in the `end=` statement.*

This will print the second print statement on the same line as the first.

Good Morning Friends

You can also add a word inside the `end=` statement and it will attach between the two statements.

```
1 print("Good Morning", end='my best')
2 print("Friends")
```

Try it!

This will print out

Good Morningmy bestFriends

Now let's add spaces in the `end=` statement to make a smooth print statement

```
1 print("Good Morning", end=' my best ')
2 print("Friends")
```

Try it!

This will print out

```
Good Morning my best Friends!
```

Sep = Statement

You can also print multiple items in a row.

```
1 print("Movies", "tonight")
```

Try it!

This will automatically print both these statements on the same line separated by a space.

```
Movies tonight
```

You can specify what you want to separate the print statements with by using the `sep=` statement.

```
1 print("Movies", "tonight", sep='!')
```

Try it!

This will print out

```
Movies!tonight
```

Multiline Strings

Strings can also hold a large amount of text that goes across multiple lines of code. These are called [multiline strings](#). To create a multiline string you must use three sets of quotes, double or single, before and after the desired text. Both ways are shown below.

```
1 string1 = """I am part of a long string.
2 This the second line of code.
3 Look, I can keep going to the third line
4 and I am still being
5 assigned to the same variable."""
6
7
8
9 string2 = '''I am part of a long string.
10 This the second line of code.
11 Look, I can keep going to the third line
12 and I am still being
13 assigned to the same variable.'''
14
15
```

MULTI-LINE STRINGS

Python



Skill
Struck TM

▶ 0:00 / 1:11



Checkpoint

Strings

Create a **variable** and assign it to a **multiline string**.

Print the variable/multiline string.

Requirements:

- Multi-line string variable

Questions (8)

1. Which use of quotation marks is correct when creating strings in Python?

MULTIPLE CHOICE

Choose the correct answer:

- A. 'Hello World'
- B. "Hello World"
- C. ""Hello World""
- D. They are all valid ways to create strings.

2. How many quotation marks are required at the beginning and end when creating a multi-line string?

MULTIPLE CHOICE

Choose the correct answer:

- A. 1
- B. 2
- C. 3
- D. 4

3. True or False: Strings must ALWAYS have double quotation marks "like this" around strings.

MULTIPLE CHOICE

Choose the correct answer:

- A. True
- B. False

4. True or False: This is a string. "46"

MULTIPLE CHOICE

Choose the correct answer:

- A. True
- B. False

5. True or False: You will run into problems if you try to do math with this value. "3"

MULTIPLE CHOICE

Choose the correct answer:

- A. True
- B. False

6. Debug the following code. Assume you are trying to assign the variable to a string.

DEBUG CODE

Code to Debug:

```
1 subject = "I like to go to dance class."
```

7. Debug the following code. Assume you are trying to assign the variable to a string.

DEBUG CODE

Code to Debug:

```
1 name = Savannah
```

8. True or False: Strings can also be called string literals

MULTIPLE CHOICE

Choose the correct answer:

- A. True
- B. False

Challenges (5)

1. Favorite Quotes

What quotes do you find inspirational? If you don't know of any particular quotes, take a minute to look some up.

1. Declare a variable named `short_quote` and a variable named `long_quote`.
2. The first should have a **one-line string** with a short quote you like.
3. The second should have a **multi-line string** with a long quote you like.
4. **Print both variables** out in print statements.

Requirements:

- Declare a one-line string variable named `short_quote`.
- Declare a multi-line string variable named `long_quote`.
- Print `short_quote` in its own print statement
- Print `long_quote` in its own print statement

2. Journal Entry

Writing in a journal helps people understand and express gratitude for their day. Studies have shown that keeping a journal is linked to greater life satisfaction.

1. Declare a variable called `journal` and write a brief summary of your day in a **multi-line string**.
2. **Print** that variable.

Requirements:

- Declare a variable called `journal` with a multi-line string.
- Print the journal variable.

3. Project Ideas

Do you have an idea for a video game, a website, or an app?

Create one multi-line string variable named `ideas` .

In this variable, write out some ideas for future Python programs you'd like to develop.

Requirements:

- Declare a multi-line string variable named `ideas`
- Write about your goals for learning Python. What do you hope to develop one day? Print out your multi-line string.

4. Best Names

What are your all time favorite names? Did you ever wish you had a different name? Or maybe there is a name that you've always thought sounded awesome.

1. Choose 4 names that you really love and assign them to variables.
2. Create and print **4 variables**.
3. Assign the variables to strings:

`name1`

`name2`

`name3`

`name4`

Requirements:

- Create and print the variable named `name1`
- Create and print the variable named `name2`
- Create and print the variable named `name3`
- Create and print the variable named `name4`

5. Garden and Orchard

You're planting the world's best garden and orchard. Think of all the awesome fruits and veggies that are in the world. If you could have ANY fresh fruits or vegetables you want, what would you have?

Create **6 variables** and assign them to a string of the best produce:

```
fruit1
```

```
fruit2
```

```
fruit3
```

```
veggie1
```

```
veggie2
```

```
veggie3
```

Print each variable.

Requirements:

- Create and print the variable `fruit1`
- Create and print the variable `fruit2`
- Create and print the variable `fruit3`
- Create and print the variable `veggie1`
- Create and print the variable `veggie2`
- Create and print the variable `veggie3`

Answer Keys & Solutions

Checkpoint Solutions

Strings

```
1 idea = """I have this really great idea but it would take several lines to write
  about.
2 I'm going to throw a pool party and invite all my friends.
3 We'll need food for sure.
4 Food makes the best parties."""
5
6 print(idea)
```

Questions

1. Which use of quotation marks is correct when creating strings in Python?

MULTIPLE CHOICE

Correct Answer:

- A. 'Hello World' ✗ Incorrect
- B. "Hello World" ✗ Incorrect
- C. ""Hello World"" ✗ Incorrect
- D. They are all valid ways to create strings. ✓ Correct

Explanation:

Strings can use 1, 2, or 3 quotation marks

2. How many quotation marks are required at the beginning and end when creating a multi-line string?

MULTIPLE CHOICE

Correct Answer:

- A. 1 ✗ Incorrect
- B. 2 ✗ Incorrect

C. 3

✓ Correct

D. 4

✗ Incorrect

Explanation:

3. True or False: Strings must ALWAYS have double quotation marks "like this" around strings.

MULTIPLE CHOICE

Correct Answer:

A. True

✗ Incorrect

B. False

✓ Correct

Explanation:

Strings can use 1, 2, or 3 quotation marks.

4. True or False: This is a string. "46"

MULTIPLE CHOICE

Correct Answer:

A. True

✓ Correct

B. False

✗ Incorrect

Explanation:

Strings have quotation marks around them. Sometimes strings can look like numbers.

5. True or False: You will run into problems if you try to do math with this value. "3"

MULTIPLE CHOICE

Correct Answer:

A. True

✓ Correct

B. False

✗ Incorrect

Explanation:

If you multiplied that value by 5, it will duplicate the string to look like this: 33333

6. Debug the following code. Assume you are trying to assign the variable to a string.

DEBUG CODE

Incorrect Code:

```
1 subject = "I like to go to dance class.
```

Correct Solution:

```
1 subject = "I like to go to dance class."
```

Explanation:

Did you close your quotation marks?

7. Debug the following code. Assume you are trying to assign the variable to a string.

DEBUG CODE

Incorrect Code:

```
1 name = Savannah
```

Correct Solution:

```
1 name = "Savannah"
```

Explanation:

Strings can use either single or double quotation marks.

8. True or False: Strings can also be called string literals

MULTIPLE CHOICE

Correct Answer:

A. True

✓ Correct

B. False

✗ Incorrect

Explanation:

It's another word for strings.

Challenges

1. Favorite Quotes

Solution:

```
1 short_quote = "Be where your feet are."
2
3 long_quote = """Learning is the beginning of wealth.
4 Learning is the beginning of health.
5 Learning is the beginning of spirituality.
6 Searching and learning is where the miracle process all begins."""
7
8 print(short_quote)
9 print(long_quote)
```

2. Journal Entry

Solution:

```
1 journal = """Today I went to school.
2 I really like my schedule!
3 The classes I have with my friends are my favorite.
4 I don't like PE."""
5
6 print(journal)
```

3. Project Ideas

Solution:

```
1 ideas = """I enjoy learning Python!
2 I hope to be able to build an app someday.
3 I think an app that would keep track of meteor showers would be so awesome.
4 I want it to alert people when there will be a meteor shower that night, so they will
5 never miss it!"""
6 print(ideas)
```

4. Best Names

Solution:

```
1 name1 = "Savannah"
2 name2 = "Chloe"
3 name3 = "Cody"
4 name4 = "River"
5
6 print(name1)
7 print(name2)
8 print(name3)
9 print(name4)
```

5. Garden and Orchard

Solution:

```
1 fruit1 = "guava"
2 fruit2 = "pineapple"
3 fruit3 = "mango"
4 veggie1 = "broccoli"
5 veggie2 = "eggplant"
6 veggie3 = "tomato"
7
8 print(fruit1)
9 print(fruit2)
10 print(fruit3)
11 print(veggie1)
12 print(veggie2)
13 print(veggie3)
```