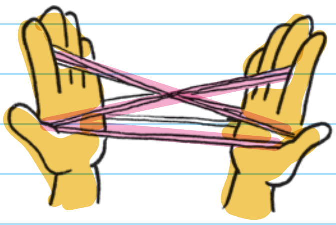
Chapter 26

STRINGS

STRINGS are a group of sequenced characters within either single or double quotation marks. You use strings to show messages or text on-screen in your program. When Python sees quotation marks around text, it reads it as a string.



FOR EXAMPLE: These are all strings because they are surrounded by quotation marks:

"Game Over"

'time: 1:06'

"Lives left:"

"Welcome LOL. Can I tell you a joke?"

Don't mix single and double quotes. For example, **'board game night"** doesn't work.

To assign a variable the value of a string, name the variable, add the assignment operator (the equal symbol), then add the string in quotation marks.

FOR EXAMPLE, for a joke-telling program, we can name a variable "welcome," add the assignment operator, then add the value "Welcome LOL. Can I tell you a joke?" like this:

```
welcome = "Welcome LOL. Can I tell you a joke?"
```

USING THE PRINT FUNCTION WITH A STRING

To display the value of a variable using the print function:

- First assign the variable a value.

```
welcome = "Welcome LOL. Can I tell you a joke?"
```

- Then type the name of the variable inside the parentheses of the print function.

```
print(welcome)
```

VARIABLE

FUNCTION

This will make the shell window display the welcome message:

```
Welcome LOL. Can I tell you a joke?
```

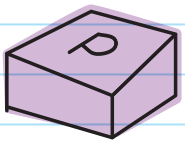
IMPORTANT: if you used quotation marks around the variable name, like in the code below, the result would be different.

```
welcome = "Welcome LOL. Can I tell you a joke?"
print("welcome")
```

quotation marks included

This will make the shell window print: `welcome`

When you put text in quotation marks, Python recognizes it as a string instead of a variable. So instead of looking up the variable value, it just prints out the word "welcome" as a string.



Displaying Text on Multiple Lines

To make a string automatically display on multiple lines, use three single quotation marks (''') and type the text on new lines the way you want it to appear.

FOR EXAMPLE:

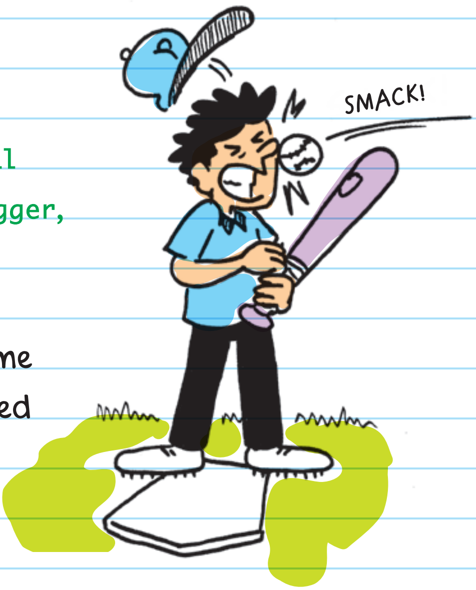
```
joke = '''I was wondering why the ball
kept getting bigger and bigger,
and then it hit me.'''

print(joke)
```

This will print:

```
I was wondering why the ball
kept getting bigger and bigger,
and then it hit me.
```

The output will have the same line breaks that were entered in the string.



Changing the Value of a String Variable

You can change the value of a variable by adding another line of code that assigns the variable a new value in the same way you assigned the original value.

FOR EXAMPLE, to replace "LOL" with "meh":

```
reaction = "lol"
```

Value of reaction: lol

```
reaction = "meh"
```

Value of reaction: meh

"meh" replaces "lol" as the value of reaction.

```
print(reaction)
```

New value of reaction: meh

This will print: `meh`

The example prints "meh" because that was the latest value of reaction.

Once the value of a variable is changed, the old value is gone for good.

FORMATTING STRINGS

You can style strings so that they include the same type of punctuation and presentation as regular typed text.

Quotation Marks

All strings have quotation marks around them. So if you want to put quotation marks inside your string, you'll need to **ESCAPE** the interior quotation marks by placing a backslash (\) before each of them.

FOR EXAMPLE:

BACKSLASH

```
tornado_joke = "What do you get if you say \"tornado"\ ten
times backward and forward? . . . A real tongue twister!"

print(tornado_joke)
```

This will print:

```
What do you get if you say "tornado" ten times backward and
forward? . . . A real tongue twister!
```

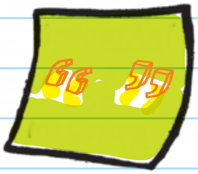
When you **escape a character**, you are telling Python to treat that character as part of the string instead of as a special character.

EXCEPTION! If you use single quotation marks at the beginning and end of your string, then you can use double quotation marks as part of your string without needing to escape them. (Or, if you use double at the beginning and end, you can use single as part of the string.)

FOR EXAMPLE, you could assign the tornado_joke variable the following value and get the same result as before:

```
tornado_joke = 'What do you get if you say "tornado" ten
times backward and forward? . . . A real tongue twister!'
```

But you should still know how to escape characters in case you need to use both single and double quotation marks or other characters in a string.



Line Breaks

To print on a new line, use an escaped "n" like this:

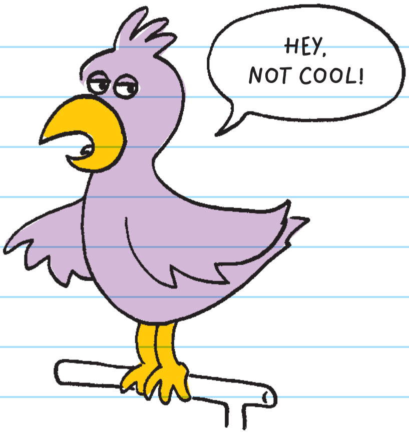
```
print("\n")
```

FOR EXAMPLE, you could break up the lines of a joke to emphasize the punch line:

```
print("I had a talking parrot.\n But it didn't say it was
hungry,\n\n so it died.")
```

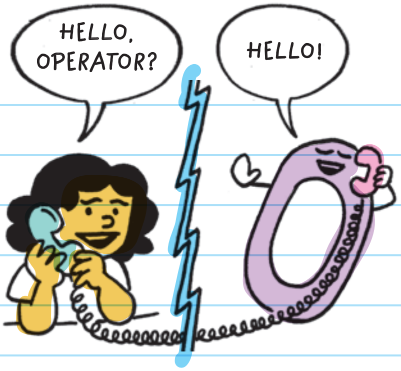
This will print:

```
I had a talking parrot. But it
didn't say it was hungry,
     so it died.
```



OPERATORS

OPERATORS are symbols that represent actions that manipulate values. Many operators come from math.



***** is a mathematical operator that represents multiplying.
+ is a mathematical operator that represents adding.

You can use operators with strings as well as numbers.

FOR EXAMPLE, to say "Hello!" 3 times, you could use:

```
print("Hello!" * 3)
```

MULTIPLICATION OPERATOR

"Hello!" * 3 means "Hello!" × 3

This would print: `Hello!Hello!Hello!`

To add space between each "Hello!" add a space at the end of the string:

```
print("Hello! " * 3 )
```

SPACE

This will print: `Hello! Hello! Hello!`

String Addition

Operators can also be used to add strings together to make a sentence.

FOR EXAMPLE:

SPACE

```
part1 = "Why did the chicken cross the road? "

part2 = "To get to the other side."

print(part1 + part2)
```

This will display: `Why did the chicken cross the road? To get to the other side.`

Another way to add strings together is by creating a new variable that is the sum of two other variables.

```
part1 = "Why did the chicken cross the road? "
part2 = "To get to the other side."
whole_joke = part1 + part2
```

This would assign the variable whole_joke the value of `Why did the chicken cross the road? To get to the other side.`

Addition operators can combine both variables and strings into a new string variable value like this:

```
animal = "alligator"
joke = "Why did the " + animal + " cross the road?"
print(joke)
```

The spaces around the + signs make the code easier to read. Because they aren't part of a string, the computer ignores them. You could also correctly type this:

```
"Why did the "+animal+" cross the road?"
```

and it would display the same thing:

```
Why did the alligator cross the road?
```

Variables are helpful because they can change values, making programs more flexible.

Sometimes you want a user to input information.

The **input()** function displays the text you add between the parentheses and waits for the user to type something.

```
input("add your text here")
```

The shell window will display "add your text here" and wait for the user to type something in the window and press enter.

You can save the information the user types in the shell window to a variable.

To do this, before the input function, type the name of the variable you want to use and also add the assignment operator:

Variable that will store the user's answer

```
variable = input("add your text here")
```

Your text or question to display for the user

FOR EXAMPLE, you can use the input() function to ask the user to give a new value to the "animal" variable from the joke example:

```
animal = input("What's your favorite animal? ")
joke = "Why did the " + animal + " cross the road?"
print(joke)
```

The program will ask the question, then wait for the user to type in their answer and press **ENTER**.

```
What's your favorite animal? Alligator
```

After the user inputs their answer, this line will print using the information stored in the "animal" variable.

```
Why did the Alligator cross the road?
```

This part is the answer the user entered.

String Functions

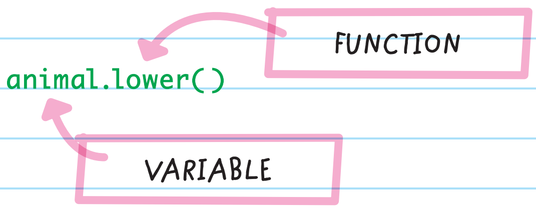
Python has built-in functions that change strings in helpful ways. There are functions to capitalize the first letter of a word or sentence, join multiple strings into a single string, convert a string to all lowercase letters, replace part of a string, format a string into a nicer-looking output, and more.

STRING FUNCTIONS help change data into a more usable format. For example, if you ask a user what their favorite animal is and they type "ALLIGATOR" in all caps, the lower() function can change all the capital letters to lowercase.

Use string functions by using this format:

```
variable.function()
```

If you have a variable called "animal" and you want to make sure the value is in lowercase, use the lower() function on the name variable:



```
animal.lower()
```

You can use the string functions along with a string almost anywhere in a program.

Examples of different placements of the lower() function:

```
animal = input("What's your favorite animal? ").lower()
print("Why did the " + animal + " cross the road?")
```

```
animal = input("What's your favorite animal? ")
animal = animal.lower()
print("Why did the " + animal + " cross the road?")
```

```
animal = input("What's your favorite animal? ")
print("Why did the " + animal.lower() + " cross the road?")
```

All three examples will print the same output:

```
Why did the alligator cross the road?
```

SOME OF THE MOST COMMON STRING FUNCTIONS

capitalize(): Changes the first letter to uppercase and the rest to lowercase

lower(): Changes all uppercase letters to lowercase

swapcase(): Changes capital letters to lowercase and lowercase letters to capital

upper(): Converts lowercase letters to uppercase

CHECK YOUR KNOWLEDGE

1. Which of these is NOT a string:
   - A. "string cheese"
   - B. 'shoelace'
   - C. "She said, \"keep pulling the thread\""
   - D. 5 + 9

2. What will the Python code below print?

```
art = "painting"
print("my favorite kind of art is: " + art)
```

3. Explain how to make a string print on multiple lines.

4. If you want to ask a user their favorite color, what would be a good function to use? How would you write it?

5. What does it mean to escape a quotation mark in a string?

6. What code could you add as input to the print function below so that it prints on a new line?

```
print(______)
```

7. What will the following code print?

```
flavor1 = "chocolate"
flavor2 = "strawberry"
flavor3 = "vanilla"
print("My favorite sundae has these 3 flavors: " +
flavor1 + ", " + flavor2 + ", and " + flavor3 + ".")
```

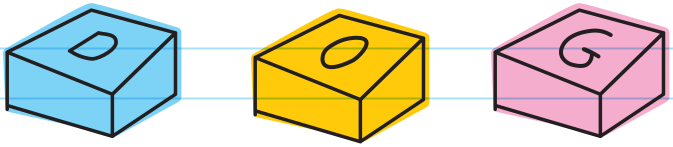
8. Suppose you want to ask a user to input their dog's name, which you store in the variable "dog." You also want to capitalize the first letter of the dog's name. Which of the following will NOT capitalize the dog's name?

A. `capitalize( ) = dog`

B. `dog = dog.capitalize( )`

C. `dog = input("What’s your dog’s name?").capitalize( )`

D. `print("Your dog’s name is: " + dog.capitalize( ))`



CHECK YOUR ANSWERS

1. D

2. my favorite kind of art is: painting

3. You can surround the string in three single quotation marks (''') and enter the text on new lines in the way you want it to appear. Or you can insert "\n" wherever you want a new line to start.

4. The input function, written like this: input("What's your favorite color?")

5. When you escape a quotation mark, you are making it part of the string instead of using it as a special character to mark the beginning or end of a string.

6. "\n"

7. My favorite sundae has these three flavors: chocolate, strawberry, and vanilla.

8. A