

Python 9 Worksheet: String Manipulation

Student Resource



604KB

Python Worksheet 9 - String Manipulation.docx



254KB

Python Worksheet 9 - String Manipulation.pdf
pdf

Starter Activity

Which of these variables are strings?

```
name = "Fred"  
name = input("What is your name? ")  
code = "123"  
value = input("Enter a number ")  
age = int(input("How old are you? "))  
happy = "True"  
weekday = True
```

Activity 1: String Functions

The `len()` function is used to find out how many characters there are in a string.

```
name = input("Enter your name ")  
length = len(name)  
print("Your name has", length, "characters in it")
```

```
Enter your name
```

```
Ada
```

```
Your name has 3 characters in it
```

The `.upper()` function returns the uppercase version of a string.

The `.lower()` function returns the lowercase version of a string.

Notice that neither of these change the string itself. If you want to do this, you need to assign the result to the variable as in line 2 below.

```
surname = input("What is your surname? ")
surname = surname.upper()
print("Your surname in uppercase is", surname)

surname = surname.lower()
print("Your surname in lowercase is", surname)
```

```
What is your surname?
```

```
Lovelace
```

```
Your surname in uppercase is LOVELACE
```

```
Your surname in lowercase is lovelace
```

Use the examples above to help you with the following exercises. For each exercise, paste in a screenshot of your program and testing output.

1. Change Case

Write a program that asks the user to enter their first name and surname. Convert their surname to uppercase and output it in the following format `<firstname>` `<surname>`.

E.g. If the user enters *Ada* and *Lovelace*, your program should output *Ada*
LOVELACE

2. Show in a different case

Write a program that asks the user to enter a phrase. Then output the phrase in lower case followed by the original phrase with the case unaltered.

E.g. If the user enters the phrase, *Hello there*, your program should output *hello there*, followed by *Hello there*.

3. Length of a phrase

Write a program that asks the user to enter a phrase. Then output the number of characters in the string.

E.g. If the user enters the phrase, *Hello there*, your program should output *There are 11 characters in the phrase*

Activity 2: String Formatting

```
name = input("What is your name? ")
age = int(input("How old are you? "))
hobby = input("What is your favourite hobby? ")
print(f"{name} is {age} years old and enjoys {hobby}")

print(f"You will be 18 in {18 - age} years")

print(f"There are {len(name)} characters in your name")
```

Use the examples above to help you with these exercises. For each exercise, paste in a screenshot of your program and testing output.

1. Ask the user to enter their name, hometown and number of siblings and then use f-strings to output a message.

2. Extend your answer to exercise 1 to output the length of their hometown and how many children there are in the family. E.g. if the user enters York for their hometown and 2 for the number of siblings, your program should output:

There are 4 letters in York

There are 3 children in your home

Activity 3: Strings Index Investigation

Copy this code and run it. Then answer the questions and explain the reasons for the output in the table that follows.

```
name = input("What is your name? ")
length = len(name)

print(name)
print(length)
print(name[1])
print(name[2])
print(name[0])
print(name[length-1])
print(name[length])
```

What name did you enter? _____

What was the length? _____

Code	Output	Explanation
<code>print(name[1])</code>		
<code>print(name[2])</code>		
<code>print(name[0])</code>		
<code>print(name[length-1])</code>		

Code	Output	Explanation
<pre>print(name[length])</pre>		

Activity 4: Iterating over a string

For each exercise, paste in a screenshot of your program and testing output.

1. Iterating over a string

Complete this code to output each letter in a string using a `while` loop.

```
name = input("What is your name? ")
length = len(name)

index = ??
while index < ?? :
    print(name[??])
    ??
```

2. Using indexes

Write a program that asks the user to enter a phrase and then print out the first, sixth and last character in that phrase. You can assume that the phrase is at least six characters long.

3. Every second character

Change your answer to exercise 1 so that it prints every second character instead. E.g. if the user enters *Freddie*, your program should output *Fede*

4. Backwards

Now change your answer to exercise 3 so that it prints the phrase backwards. E.g. if the user enters *Mary*, your program should output *y r a M*

Hints:

-

- What is the index of the first character you want to output?
- What is the index of the last character you want to output?
- How should you change the index each time the loop repeats?

Activity 5: Selection in a Loop

You may not use the `in` keyword for these exercises. For each exercise, paste in a screenshot of your program and testing output.

1. Count vowels

Write a program that asks the user to enter a phrase and counts the number of vowels in the phrase. The vowels are a, e, i, o and u.

Examples:

If the user enters *Hello there*, your program should output 4

Extension:

Extend your program so that it counts both uppercase and lowercase vowels. E.g. if the user enters *ALPHAbet*, your program should output 3

2. Find a letter

Ask the user to enter a phrase, then ask the user to enter a letter to find. Find out if that letter exists in the phrase and output an appropriate message. Your program should be case-sensitive, so it should not find an E if you are looking for an e.

Examples:

If the user enters *Hello* and *e*, your program should output:

Found the letter e in Hello

If the user enters *Hello* and *a*, your program should output:

The letter a was not found in Hello

Hint: You know if you have found a letter as soon as you see it, but you do not know that a letter is not there until you have finished checking every single letter.