Task: The String Datatype

www.hyperiondev.com

Introduction

Welcome to the String Datatype Task!

Strings are some of the most important and useful data types in programming. Why? Let's think about it like this. When you were born your parents did not immediately teach you to do mathematical sums, not 1+1 or standard deviation. The first thing the taught you to do was to speak, to say words, to construct full sentences, to say "Mom" or "Dad". Well this is why we are going to "teach" the computer to first be able to communicate with the user and the only way to do this is to have a good grasp of strings.

Connect with your mentor

CONNECT

Remember that with our courses - you're not alone! You can contact your mentor to get support on any aspect of your course.

The best way to get help is to login to www.hyperiondev.com/support to start a chat with your mentor. You can also schedule a call or get support via email.

Your mentor is happy to offer you support that is tailored to your individual career or education needs. Do not hesitate to ask a question or for additional support!

A note from the Hyperion Team...

South Africa is often considered a late bloomer, as far as technology is concerned. More often than not, technologies pioneered on the other side of the Atlantic, or generally off-continent, take time to reach our shores. It's because of this that most people assume research and development of anything cutting edge in computer science and technology has stagnated in the country. One new initiative however, is looking to change this view. It aims to bring together the best minds driving computer science and related fields in South Africa, hoping to spearhead the growth of that field's body of knowledge. Read more about it here.

-The Hyperion Team

What are Strings

A **String** is the equivalent of a sentence in English. It is a list of letters, numerals, symbols and special characters which are put together to form one big **String** or sentence. The values that we can store in a string are vast. An example of what strings can store is the surname, name, address, etc. of a person.

In Python, strings must be written within "quotation marks" in order for the computer to be able to read it.

The smallest possible string contains 0 characters and is called an **empty string**.

Examples

```
name = "Linda"

song = "The Bird Song",

licencePlate = "CTA 456 GP"
```

Strings are probably the most important variable in programming. They are used as a medium of communication between the computer and the user. The user is able to enter information as a string and the program is able to use the data to perform calculations and finally display the calculated answer to the user. It is the variable's job to store the information entered by the user and display information back to the user in an easy to read format.

```
Name = "John"
Joke = "Knock, knock, Who's there?"
```

You can use any name for your variable but the actual string you are assigning to the variable must be within “ “ (quotation marks).

Concatenation of Strings

Strings are able to be added to one another, this is called concatenation.

```
Name = "Peter"
Surname = "Parker"
FullName = Name + Surname
```

FullName will now store the value “PeterParker”

The + symbol simply joins the strings as is. If you want it to make your code more presentable you should put spaces between the words.

```
FullName = Name + " " + Surname
```

We now added a blank space in between the two strings, so FullName will now store the value “Peter Parker”

Numbers as Strings

We can even store numbers as strings. When we are storing a number (i.e. 9, 10, 231) as a string, we are essentially storing it as a word. The number will lose all its number defining characteristics. So the number will not be able to be used in any calculations. All you can do with it is read or display it.

In real life sometimes we don’t need numbers to actually do calculations we just need them for information purposes. For example, the house number you live in, let just say, 45 2nd Street JHB 2093. We won’t be performing any calculations with this address. What benefit would it be for us to find out the what the sum is of all the house numbers in an area? The only thing we need is to know what number the house is when visiting or delivering a package to it. The number just needs to be clearly visible. This is the same concept of storing a number as a string; all we want is to be able to take in a value and display it to the user.

Example

```
favNum = "9"
```

A note from Masood...

Hey again, Have you heard about a Raspberry Pi before? No, it's not a new dessert, let me explain.

WHAT IS A RASPBERRY PI?

The Raspberry Pi is a computer that is so small it can fit in your pocket. You are able to plug it into your TV and connect a keyboard to it and you will have a fully set up personal computer. Don't be so quick to judge it just because of its size. It can be used for electronics projects, and for many of the things that your desktop PC does, like spreadsheets, word processing, browsing the internet, and playing games. It also plays high-definition video.

Imagine all that power and it is only as big as a credit card. This is the newest outlet for where your programs can be run on. It is always useful to know what new developments in technology are happening so that you don't fall behind.

A Raspberry Pi is relatively cheap to buy here in South Africa. If you want to learn more about it you can visit: https://www.raspberrypi.org/

- **Masood Gool**, Online Trainer

We will now explore the various things you can do to Strings

A string is essentially a list of characters. For example the word “Hello” is made up of the characters H+e+l+l+o. We are able to use this to our advantage as we are able to access the exact character we need using indexing. Each character of a String (including blank spaces) is indexed by numbers starting from 0 for first character on the left.

```
greeting = "Hello"
print  greeting[0] + greeting[1] + greeting[2] + greeting[3] + greeting[4]
```

You can use len() to get the number of characters in a string or length of a string. The print statement below prints out 12, because "Hello world!" is 12 characters long, including punctuation and spaces.

```
print len("Hello World!")
```

You can also slice a string. Slicing in Python, extracts characters from a string based on a starting index and ending index. It enables you to extract more than one character or "chunk" of characters from a String. The print statement below will print out a piece of the string. It will start at position/index 3, and end at position/index 6.

```
print aString[3:7]
```

You can even put negative numbers inside the brackets . The characters are also indexed from right to left using negative numbers, where -1 is the rightmost index and so on. Using negative indices are an easy way of starting at the end of the string instead of the beginning. This way, -3 means "3rd character from the end".

You can print a String in reverse by using [::-1]. This is know as an extended slice. The syntax for an extended slice is [begin : end : step]. By not including a beginning and end and specifying a step of -1, the string will be reversed. You can find out more about extended slices here.

String Handling

There are various functions in Python that we can use to manipulate strings. Functions are used to save us from having to write monotonous code over and over. They have built in code that is stored in Python which is automatically performed when we need it to. Here are a few examples of useful String functions.

```
aString = "Hello World"
print aString.upper()      # prints out HELLO WORLD
print aString.lower()      # prints out hello world
```

These make a new string with all letters converted to uppercase and lowercase, respectively.

```
aSentence = "Welcome$to$the$world$of$programming"
print aSentence.replace("$" , " ")   # prints out Welcome to the world of programming
```

This will replace any occurrence of a string with a string of your choice.

```
strHelp = "******Please leave me alone******"
print strHelp.strip('*')  # prints out Please leave me alone
```

This will remove any unwanted string from the start and end of a string.

Most of the programming languages provide built-in functions to manipulate strings, i.e., you can concatenate strings, you can search from a string, you can extract substrings from a string, etc. We will cover string handling and more functions in a future task.

Instructions

Before you get started we strongly suggest you start using Notepad++ or IDLE to open all text files (.txt) and python files (.py). Do not use the normal Windows notepad as it will be much harder to read.

First read example.py, open it using Notepad++ (Right click the file and select 'Edit with Notepad++') or IDLE.

- example.py should help you understand some simple Python. Every task will have example code to help you get started. Make sure you read all of example.py and try your best to understand.

- You may run example.py to see the output. Feel free to write and run your own example code before doing the Task to become more comfortable with Python.

- You are not required to read the entirety of Additional Reading.pdf, it is purely for extra reference.

Compulsory Task 1

Follow these steps:

- Create a new Python file in this folder called "Strings.py"
- Declare the variable called hero = "Super Man"
- Print it out in the following way:
  - S^U^P^E^R M^A^N

Compulsory Task 2

Follow these steps:

- Create a new Python file in this folder called "Replace.py"
- Save the sentence: "The!quick!brown!fox!jumps!over!the!lazy!dog!" as a single string
- Now reprint this sentence as "The quick brown fox jumps over the lazy dog" using the **replace()** function to change every "!" exclamation mark with a blank space.
- Now reprint that sentence as "THE QUICK BROWN FOX JUMPS OVER THE LAZY DOG. using the **.upper()** function
- Print the sentence in reverse.

Things to look out for:

1. Make sure that you have installed and setup all programs correctly. You have setup **Dropbox** correctly if you are reading this, but **Python or Notepad++** may not be installed correctly.
2. If you are not using Windows, please ask your mentor for alternative instructions.

Give your thoughts..

★ RATE

Hyperion strives to provide internationally-excellent course content that helps you achieve your learning outcomes. Think the content of this task, or this course as a whole, can be improved or think we've done a good job?

Click here to share your thoughts anonymously.