Simplifying AI and Machine-Learning with Watson Studio
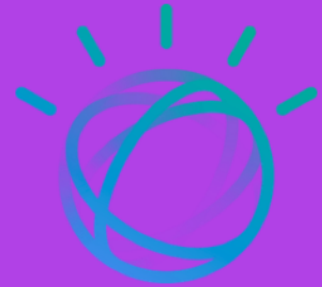
- Get your free account and use the Lite plan forever
- No credit card and no autorenewals

Click Here

(https://cocl.us/PYTHON101EN_portal_add_top)

(https://cognitiveclass.ai/)

String Operations

Welcome! This notebook will teach you about the string operations in the Python Programming Language. By the end of this notebook, you'll know the basics string operations in Python, including indexing, escape sequences and operations.

Table of Contents

What are Strings?

The following example shows a string contained within 2 quotation marks:

In [ ]:

```
# Use quotation marks for defining string

"Michael Jackson"
```

We can also use single quotation marks:

In [ ]:

```
# Use single quotation marks for defining string

'Michael Jackson'
```

A string can be a combination of spaces and digits:

In [ ]:

```
# Digitals and spaces in string

'1 2 3 4 5 6 '
```

A string can also be a combination of special characters :

In [ ]:

```
# Special characters in string

'@#2_#]&*^%$'
```

We can print our string using the print statement:

In [1]:

```
# Print the string

print("hello!")
```

hello!

We can bind or assign a string to another variable:

In [2]:

```
# Assign string to variable

Name = "Michael Jackson"
Name
```

Out[2]:

'Michael Jackson'

Indexing

It is helpful to think of a string as an ordered sequence. Each element in the sequence can be accessed using an index represented by the array of numbers:

Name= "Michael Jackson"

M	i	c	h	a	e	l		J	a	c	k	s	o	n
0	1	2	3	4	5	6	7	8	9	10	11	12	13	14

The first index can be accessed as follows:

[Tip]: Because indexing starts at 0, it means the first index is on the index 0.

In [3]:

```
# Print the first element in the string
print(Name[0])
```

M

We can access index 6:

In [4]:

```
# Print the element on index 6 in the string
print(Name[6])
```

l

Moreover, we can access the 13th index:

In [5]:

```
# Print the element on the 13th index in the string
print(Name[13])
```

o

Negative Indexing

We can also use negative indexing with strings:

Name= "Michael Jackson"

M	i	c	h	a	e	l		J	a	c	k	s	o	n
-15	-14	-13	-12	-11	-10	-9	-8	-7	-6	-5	-4	-3	-2	-1

Negative index can help us to count the element from the end of the string.

The last element is given by the index -1:

In [6]:

```
# Print the last element in the string

print(Name[-1])
```

n

The first element can be obtained by index -15:

In [7]:

```
# Print the first element in the string

print(Name[-15])
```

M

We can find the number of characters in a string by using len , short for length:

In [8]:

```
# Find the length of string

len("Michael Jackson")
```
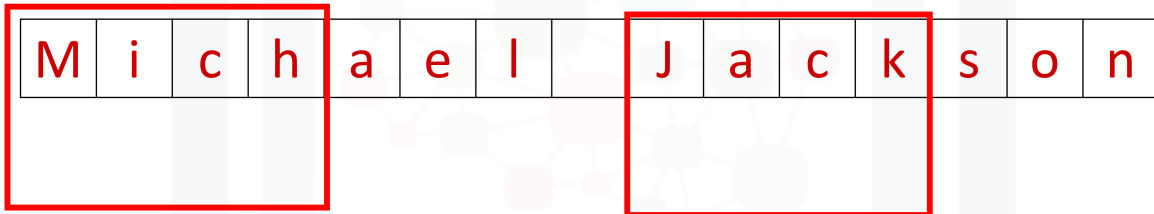
Out[8]:

15

Slicing

We can obtain multiple characters from a string using slicing, we can obtain the 0 to 4th and 8th to the 12th element:

Name= "Michael Jackson"

| M | i | c | h | a | e | l | | J | a | c | k | s | o | n |

Name[0:4] =Mich   Name[8:12] =Jack

[Tip]: When taking the slice, the first number means the index (start at 0), and the second number means the length from the index to the last element you want (start at 1)

In [9]:

```
# Take the slice on variable Name with only index 0 to index 3
Name[0:4]
```

Out[9]:

'Mich'

In [10]:

```
# Take the slice on variable Name with only index 8 to index 11

Name[8:12]
```

Out[10]:

'Jack'

Stride

We can also input a stride value as follows, with the '2' indicating that we are selecting every second variable:

Name= "Michael Jackson"

M	i	c	h	a	e	l		J	a	c	k	s	o	n
0	1	2	3	4	5	6	7	8	9	10	11	12	13	14

Name[::2]: "McalJcsn"

In [11]:

```
# Get every second element. The elments on index 1, 3, 5 ...

Name[::2]
```

Out[11]:

'McalJcsn'

We can also incorporate slicing with the stride. In this case, we select the first five elements and then use the stride:

In [12]:

```
# Get every second element in the range from index 0 to index 4

Name[0:5:2]
```

Out[12]:

'Mca'

Concatenate Strings

We can concatenate or combine strings by using the addition symbols, and the result is a new string that is a combination of both:

In [13]:

```
# Concatenate two strings

Statement = Name + "is the best"
Statement
```

Out[13]:

'Michael Jacksonis the best'

To replicate values of a string we simply multiply the string by the number of times we would like to replicate it. In this case, the number is three. The result is a new string, and this new string consists of three copies of the original string:

In [14]:

```
# Print the string for 3 times

3 * "Michael Jackson"
```

Out[14]:

'Michael JacksonMichael JacksonMichael Jackson'

You can create a new string by setting it to the original variable. Concatenated with a new string, the result is a new string that changes from Michael Jackson to "Michael Jackson is the best".

In [15]:

```
# Concatenate strings

Name = "Michael Jackson"
Name = Name + " is the best"
Name
```

Out[15]:

'Michael Jackson is the best'

Escape Sequences

Back slashes represent the beginning of escape sequences. Escape sequences represent strings that may be difficult to input. For example, back slash "n" represents a new line. The output is given by a new line after the back slash "n" is encountered:

In [16]:

```python
# New line escape sequence

print(" Michael Jackson \n is the best" )
```

Michael Jackson
is the best

Similarly, back slash "t" represents a tab:

In [17]:

```python
# Tab escape sequence

print(" Michael Jackson \t is the best" )
```

Michael Jackson   is the best

If you want to place a back slash in your string, use a double back slash:

In [18]:

```python
# Include back slash in string

print(" Michael Jackson \\ is the best" )
```

Michael Jackson \ is the best

We can also place an "r" before the string to display the backslash:

In [19]:

```python
# r will tell python that string will be display as raw string

print(r" Michael Jackson \ is the best" )
```

Michael Jackson \ is the best

String Operations

There are many string operation methods in Python that can be used to manipulate the data. We are going to use some basic string operations on the data.

Let's try with the method `upper` ; this method converts lower case characters to upper case characters:

In [20]:

```
# Convert all the characters in string to upper case

A = "Thriller is the sixth studio album"
print("before upper:", A)
B = A.upper()
print("After upper:", B)
```

before upper: Thriller is the sixth studio album
After upper: THRILLER IS THE SIXTH STUDIO ALBUM

The method `replace` replaces a segment of the string, i.e. a substring with a new string. We input the part of the string we would like to change. The second argument is what we would like to exchange the segment with, and the result is a new string with the segment changed:

In [21]:

```
# Replace the old substring with the new target substring is the segment has been found in the string

A = "Michael Jackson is the best"
B = A.replace('Michael', 'Janet')
B
```

Out[21]:

'Janet Jackson is the best'

The method `find` finds a sub-string. The argument is the substring you would like to find, and the output is the first index of the sequence. We can find the sub-string `jack` or `el` .

Name= "Michael Jackson"

M	i	c	h	a	e	l		J	a	c	k	s	o	n
0	1	2	3	4	5	6	7	8	9	10	11	12	13	14

Name.find('el'):5   Name.find('Jack'):8

In [22]:

```
# Find the substring in the string. Only the index of the first elment of substring in string will be the output

Name = "Michael Jackson"
Name.find('el')
```

Out[22]:

5

In [23]:

```python
# Find the substring in the string.
Name.find('Jack')
```

Out[23]:

8

If the sub-string is not in the string then the output is a negative one. For example, the string 'Jasdfasdasdf' is not a substring:

In [24]:

```
# If cannot find the substring in the string
Name.find('Jasdfasdasdf')
```

Out[24]:

-1

Quiz on Strings

What is the value of the variable   A   after the following code is executed?

In [26]:

```python
# Write your code below and press Shift+Enter to execute
A = "1"
print(A)
```

1

Double-click **here** for the solution.

What is the value of the variable   B   after the following code is executed?

In [27]:

```
# Write your code below and press Shift+Enter to execute
B = "2"
B
```

Out[27]:

'2'

Double-click **here** for the solution.

What is the value of the variable   C   after the following code is executed?

In [28]:

```
# Write your code below and press Shift+Enter to execute

C = A + B
C
```

Out[28]:

'12'

Double-click **here** for the solution.

Consider the variable   D   use slicing to print out the first three elements:

In [30]:

```
# Write your code below and press Shift+Enter to execute
D = "ABCDEFG"
D[0:3]
```

Out[30]:

'ABC'

Double-click **here** for the solution.

Use a stride value of 2 to print out every second character of the string   E :

In [31]:

```python
# Write your code below and press Shift+Enter to execute

E = 'clocrkr1e1c1t'
E[::2]
```

Out[31]:

'correct'

Double-click **here** for the solution.

Print out a backslash:

In [35]:

```python
# Write your code below and press Shift+Enter to execute
print("\\")
```

\

Double-click **here** for the solution.

Convert the variable   F   to uppercase:

In [36]:

```python
# Write your code below and press Shift+Enter to execute

F = "You are wrong"
F.upper()
```

Out[36]:

'YOU ARE WRONG'

Double-click **here** for the solution.

Consider the variable G , and find the first index of the sub-string snow :

In [38]:

```
# Write your code below and press Shift+Enter to execute

G = "Mary had a little lamb Little lamb, little lamb Mary had a little lamb \
Its fleece was white as snow And everywhere that Mary went Mary went, Mary went \
Everywhere that Mary went The lamb was sure to go"
G.find("snow")
```

Out[38]:

95

Double-click **here** for the solution.

In the variable G , replace the sub-string Mary   with Bob :

In [41]:

```python
# Write your code below and press Shift+Enter to execute
G.replace("Mary","Bob")
G
```

Out[41]:

'Mary had a little lamb Little lamb, little lamb Mary had a little lamb Its fleece was white as snow And everywhere that Mary went Mary went, Mary went Everywhere that Mary went The lamb was sure to go'

Double-click **here** for the solution.

The last exercise!

Congratulations, you have completed your first lesson and hands-on lab in Python. However, there is one more thing you need to do. The Data Science community encourages sharing work. The best way to share and showcase your work is to share it on GitHub. By sharing your notebook on GitHub you are not only building your reputation with fellow data scientists, but you can also show it off when applying for a job. Even though this was your first piece of work, it is never too early to start building good habits. So, please read and follow this article (https://cognitiveclass.ai/blog/data-scientists-stand-out-by-sharing-your-notebooks/) to learn how to share your work.

Get IBM Watson Studio free of charge!

Get IBM Watson Studio free of charge!

Build and train AI & machine learning models, prepare and analyze data – all in a flexible, hybrid cloud environment. Get IBM Watson Studio Lite Plan free of charge.

Learn
Get started or get better with built-in learning.

Create
Use the best of open source tooling with IBM innovation.

Collaborate
Work smarter using community, work faster with your team.

Sign Up For a Free Trial

(https://cocl.us/PYTHON101EN_portal_add_bottom)

About the Authors:

Joseph Santarcangelo (https://www.linkedin.com/in/joseph-s-50398b136/) is a Data Scientist at IBM, and holds a PhD in Electrical Engineering. His research focused on using Machine Learning, Signal Processing, and Computer Vision to determine how videos impact human cognition. Joseph has been working for IBM since he completed his PhD.

Other contributors: Mavis Zhou (www.linkedin.com/in/jiahui-mavis-zhou-a4537814a)