Strings

Sequence Types

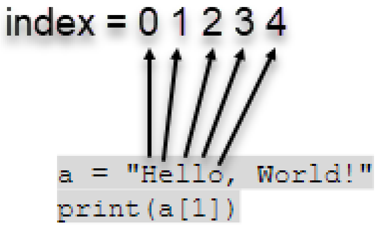
- Sequences allow you to store multiple values in an **organized and efficient way**.
- There are seven sequence types: strings, Unicode strings, lists, tuples, bytearrays, buffers, and xrange objects.
- *Nothing to worry looking at this long list, you will get to know it gradually.*

Python Strings

- Strings are sequence of characters.
- Let us see some examples of String: My name is Rahul, Rahul, Go home. All these are examples of String.
- In Python, Strings are called str.
- There is a specific way of defining String in Python – it is defined within single quote (') or double quotes (") or even triple quotes ("').

Accessing String Elements

- Square brackets can be used to access elements of the string.
- **Remember that the first character has index 0.**
- Index refers to position of a character in a string. In python index number starts from 0.
- Example: `a = "Hello, World!" print(a[1])`
- **Will give an output e. Can you understand why?**

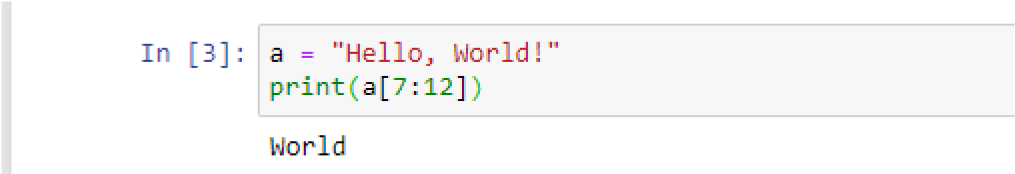


```
a = "Hello, World!"
print(a[1])
```

- Hope you got the answer to the previous question now!

String Slicing

- We can also call out a range of characters from the string using string slicing.
- Specify the start index and the end index, separated by a colon, to return a part of the string. Note that the character of the end index is not included.
- Suppose we want to print World from the string "Hello World". We can do so as below:



```
In [3]: a = "Hello, World!"
        print(a[7:12])

        World
```

Negative Indexing

- If we have a long string and we want to pinpoint an item towards the end, we can also count backwards from the end of the string, starting at the index number -1

- Printing 'r' from the string : `a = "Hello, World!"`

`print(a[-4])`

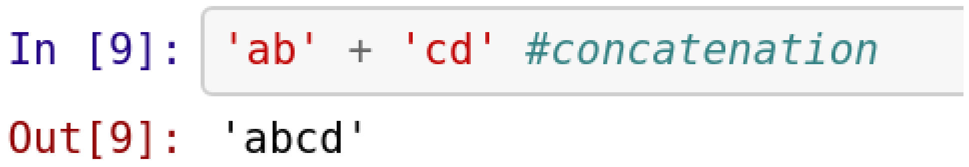
- Get the characters from position -5 to position -1, starting the count from the end of the string: `print(a[-5:-2])`

- Will give an output :orl

String Concatenation

- String concatenation means adding strings together.

- Use the + character to add a variable to another variable:
- Example:



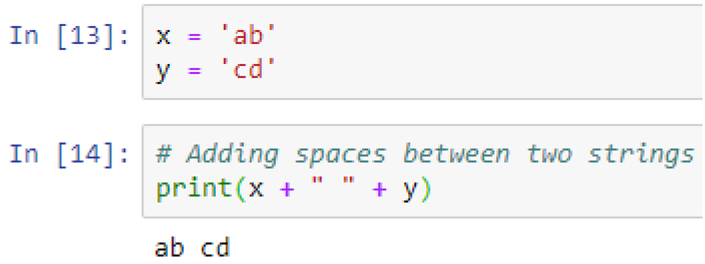
```
In [9]: 'ab' + 'cd' #concatenation

Out[9]: 'abcd'
```

- Another example: `x = "Python is " y = "awesome" z = x + y print(z)`

- **Output: Python is awesome**

String Concatenation: Add Space

- We can also add spaces between two strings



```
In [13]: x = 'ab'
         y = 'cd'

In [14]: # Adding spaces between two strings
         print(x + " " + y)

         ab cd
```

String Length

- To get the length of a string, use the len( ) function.

- Getting length of the string a :

```
a = "Hello, World!"
print(len(a))
```

- **Output: 13**

String Methods

- Python has a set of built-in methods that you can use on strings.
- Must learn: Learn about important string methods from the below cheatsheet: https://www.codecademy.com/learn/learn-python-3/modules/learn-python3-strings/cheatsheet
- Tip: If you are unable to follow, run the code and make out the difference.