Python Data Types: Strings

Computer Science ICS20

Creating Strings

Strings are another data type available in Python. You can think of them as a list of characters. Strings can be created by enclosing characters in either single quotes or double quotes; there is no difference in Python. For example, here are 2 string variables:

```
str1 = 'Hello World!'
str2 = "Python rules!"
```

Substrings

Even though strings are made up of characters, Python does not have a character data type. Instead, a single character is also a string of length one. Often it is useful to be able to access individual characters within a larger string. We can do this by using square brackets. For example:

```
str1 = 'Hello World!'
str2 = "Python rules!"

print("The 1st character of str1 is ", str1[0])
print("The 1st character of str2 is ", str2[0])
```

When the above code is executed the result is:

```
The 1st character of str1 is H
The 1st character of str2 is P
```

But the brackets can also be used to extract a substring as well. For example:

```
str1 = 'Hello World!'
str2 = "Python rules!"

print("str1[:3]: ", str1[:3])
print("str1[1:5]: ", str1[1:6])
print("str2[4:10] ", str2[4:10])
```

When the above code is executed the result is:

```
str[:3]: Hel
str1[1:7]: ello W
str2[4:10]: on rul
```

Changing Strings

Changing the contents of a string can be done by concatenating or appending other strings to the current string and saving it to another string variable or back to itself:

```
str1 = ‘Hello World!’
str2 = “Python rules!”
str3 = str2[:6] + ‘ ‘ + str1[6:]

print(“str3 = “ + str3)
str1 = str1[:5]
print(“str1 = “ + str1)
```

The above program would output

```
str3 = Python World!
str1 = Hello
```

Escape Characters

Escape characters in strings allow non-printable characters to be inserted within strings. These special characters allow certain actions to occur. Here are some commonly used escape characters:

Backslash Notation	Description
\b	backspace
\n	start a new line
\r	carriage return
\t	tab

```
str1 = ‘Hello World!’
str2 = “Python rules!”

str = str1[:5] + ‘\n’ + str1[6:]
print(str)
```

The above program would output

```
Hello
World!
```

String Operators

There are certain operators that take on a special meaning when applied to strings. Here are the most commonly used:

Operator	Description
+	Concatenation – joins strings on either side of the operator together

*	Repetition - Creates new strings, concatenating multiple copies of the same string

String Functions

There are many functions that can operator on strings. Here are a few of them:

Function Name	**Description**
len(string)	Returns the length of the string
isnumeric()	Returns true if a unicode string contains only numeric characters and false otherwise.
lower()	Converts all uppercase letters in string to lowercase.
upper()	Converts lowercase letters in string to uppercase.
lstrip()	Removes all leading whitespace in string.
rstrip()	Removes all trailing whitespace of string.
split()	Returns a list of all the words in the string, with whitespaces removed
find(str, beg=0 end=len(string))	Determine if str occurs in string or in a substring of string if starting index beg and ending index end are given returns index if found and -1 otherwise.
count(str, beg= 0,end=len(string))	Counts how many times str occurs in string or in a substring of string if starting index beg and ending index end are given.

Exercises

1. Write a Python program that converts a date in the form “dd/mm/yy” (or “MMDDYY”) into “8 March 1965".
2. Write a Python program that converts “hh:mm:ss” (24 hour clock) into more normal form such as “2:30 pm”.
3. Write a Python program that counts how many vowels are in a given string.
4. Write a Python program that counts how many digits are in a given string.
5. Write a Python program that counts the number of words in a given string.
6. Write a Python program that finds the average word length in a given string.
7. Write a Python program that finds the total number of occurrences of a given word in a given string.
8. Write a Python program that determines if a given string is a palindrome.
9. Write a Python program that validates product codes as follows:
    - the 1st part of the code can contain only capital letters and 6 digits in any order.
    - The 2nd part is all digits and is equal to the product of the first 6 digits taken in groups of two from the left.

Example: Product code = AX6BYU56UX6CV6BNT7NM 287430

This is a valid code because the 1st part of the code contains only capital letters and 6 digits and also because the 2nd part is all digits and is equal to the product of the first 6 digits taken in groups of two from the left. ie. 65 * 66 * 67 = 287430

10. Write a Python program that takes some text and turns it into pig latin. A pig latin word is made by taking the first letter and putting it on the end and adding an “a”: for example “pig” becomes “igpa”.