

Python String

- In Python, strings are arrays of bytes representing Unicode characters.
- Python does not have a character data type; a single character is simply a string with a length of 1.
- Square brackets can be used to access elements of the string.
- Strings in Python can be created using single quotes or double quotes or even triple quotes.

Python Program to Access characters of String

```
String1 = "Python Programming"
print(String1) # displays Python Programming

# Printing First character
print(String1[0]) # displays P

# Printing Last character
print(String1[-1]) # display g

# Printing 3rd to 17th character
print(String1[3:17]) # displays hon Programming

# Printing characters between 3rd and 2nd last character
print(String1[3:-2]) # # displays hon Programmi
```

In Python, Updating or deleting of characters from a String is not allowed.

```
# Python Program to Update character of a String
String1 = "Hello, I'm a Geek"
print("Initial String: ")
print(String1)

# Updating a character
# of the String
String1[2] = 'p'
print("\nUpdating character at 2nd Index: ")
print(String1)
```

```
Traceback (most recent call last):
  File "/home/360bb1830c83a918fc78aa8979195653.py", line 10, in
    String1[2] = 'p'
TypeError: 'str' object does not support item assignment
```

```
# Use list if needed to modify string
s = list("Hello")
s[2] = 'p'
print(s)
```

More Python string method examples

```
str = "python"

#capitalize first letter
print('str.capitalize() = ', str.capitalize())

#computes length of a string
print('len(str) = ', len(str))
```

Test-run

```
str.capitalize() = Python
len(str) = 6
```

```
# example string
string = "THIS SHOULD BE LOWERCASE!"
print(string.lower())
```

```
# string with numbers
# all alphabets should be lowercase
string = "Th!s Sh0uLd B3 L0w3rCas3!"
print(string.lower())
```

Test-run

```
this should be lowercase!
Th!s sh0uld b3 l0w3rCas3!
```

The String to Reverse

```
text = "Hello World"
print(text[::-1])
```

Create a slice that starts at the end of the string, and moves backwards.

In this particular example, the slice statement `[::-1]` is the same as `[11:0:-1]` which means start at position 11 (because "Hello "World" has 11 characters), end at position 0, move with the step -1, *negative* one, which means one step backwards.

Formatting of Strings

Python Program for Formatting of Strings

Default order

```
String1 = "{} {} {}".format('Geeks', 'For', 'Life')
print("Print String in default order: ")
print(String1)
```

Positional Formatting

```
String1 = "{1} {0} {2}".format('Geeks', 'For', 'Life')
print("\nPrint String in Positional order: ")
print(String1)
```

Keyword Formatting

```
String1 = "{1} {f} {g}".format(g = 'Geeks', f = 'For', l = 'Life')
print("\nPrint String in order of Keywords: ")
print(String1)
```

Formatting of Integers

```
String1 = "{0:b}".format(16)
print("\nBinary representation of 16 is ")
print(String1)
```

Formatting of Floats

```
String1 = "{0:e}".format(165.6458)
print("\nExponent representation of 165.6458 is ")
print(String1)
```

```
# Rounding off Integers
String1 = "{0:.2f}".format(1/6)
print("\none-sixth is : ")
print(String1)

# String alignment
String1 = "|{:<10}|{:^10}|{:>10}|".format('Geeks','for','Geeks')
print("\nLeft, center and right alignment with Formatting: ")
print(String1)
```

Output

Print String in default order:
Geeks For Life

Print String in Positional order:
For Geeks Life

Print String in order of Keywords:
Life For Geeks

Binary representation of 16 is
10000

Exponent representation of 165.6458 is
1.656458e+02

one-sixth is :
0.17

Left, center and right alignment with Formatting:
|Geeks | for | Geeks|

Example program using of Python math library

```
import math

p = math.pow(2,3)
print (p)

data = 21.6
print (math.ceil(data))
```

```
value = 3.0
print ("square root of 3 is", math.sqrt(value))

# returning the factorial of 5
b = 5
print ("The factorial of 5 is: ", math.factorial(b))
```

Test-run

```
8.0
22
square root of 3 is 1.7320508075688772
The factorial of 5 is: 120
```

String Function - Example Programs

<https://www.geeksforgeeks.org/python-strings/>

List of Functions in Python String Modules

Method	Description
Python String capitalize()	Converts first character to Capital Letter
Python String center()	Pads string with specified character
Python String casefold()	converts to casefolded strings
Python String count()	returns occurrences of substring in string
Python String endswith()	Checks if String Ends with the Specified Suffix
Python String expandtabs()	Replaces Tab character With Spaces
Python String encode()	returns encoded string of given string

Method	Description
Python String find()	Returns the index of first occurrence of substring
Python String format()	formats string into nicer output
Python String index()	Returns Index of Substring
Python String isalnum()	Checks Alphanumeric Character
Python String isalpha()	Checks if All Characters are Alphabets
Python String isdecimal()	Checks Decimal Characters
Python String isdigit()	Checks Digit Characters
Python String isidentifier()	Checks for Valid Identifier
Python String islower()	Checks if all Alphabets in a String are Lowercase
Python String isnumeric()	Checks Numeric Characters
Python String isprintable()	Checks Printable Character
Python String isspace()	Checks Whitespace Characters
Python String istitle()	Checks for Titlecased String
Python String isupper()	returns if all characters are uppercase characters
Python String join()	Returns a Concatenated String
Python String ljust()	returns left-justified string of given width
Python String rjust()	returns right-justified string of given width
Python String lower()	returns lowercased string
Python String upper()	returns uppercased string
Python String swapcase()	swap uppercase characters to lowercase; vice versa
Python String lstrip()	Removes Leading Characters

Method	Description
Python String <code>rstrip()</code>	Removes Trailing Characters
Python String <code>strip()</code>	Removes Both Leading and Trailing Characters
Python String <code>partition()</code>	Returns a Tuple
Python String <code>maketrans()</code>	returns a translation table
Python String <code>rpartition()</code>	Returns a Tuple
Python String <code>translate()</code>	returns mapped charactered string
Python String <code>replace()</code>	Replaces Substring Inside
Python String <code>rfind()</code>	Returns the Highest Index of Substring
Python String <code>rindex()</code>	Returns Highest Index of Substring
Python String <code>split()</code>	Splits String from Left
Python String <code>rsplit()</code>	Splits String From Right
Python String <code>splitlines()</code>	Splits String at Line Boundaries
Python String <code>startswith()</code>	Checks if String Starts with the Specified String
Python String <code>title()</code>	Returns a Title Cased String
Python String <code>zfill()</code>	Returns a Copy of The String Padded With Zeros
Python String <code>format_map()</code>	Formats the String Using Dictionary

List of Functions in Python Math Module

Function	Description
----------	-------------

<code>ceil(x)</code>	Returns the smallest integer greater than or equal to x.
<code>copysign(x, y)</code>	Returns x with the sign of y
<code>fabs(x)</code>	Returns the absolute value of x
<code>factorial(x)</code>	Returns the factorial of x
<code>floor(x)</code>	Returns the largest integer less than or equal to x
<code>fmod(x, y)</code>	Returns the remainder when x is divided by y
<code>frexp(x)</code>	Returns the mantissa and exponent of x as the pair (m, e)
<code>fsum(iterable)</code>	Returns an accurate floating point sum of values in the iterable
<code>isfinite(x)</code>	Returns True if x is neither an infinity nor a NaN (Not a Number)
<code>isinf(x)</code>	Returns True if x is a positive or negative infinity
<code>isnan(x)</code>	Returns True if x is a NaN
<code>ldexp(x, i)</code>	Returns $x * (2^{**i})$
<code>modf(x)</code>	Returns the fractional and integer parts of x
<code>trunc(x)</code>	Returns the truncated integer value of x
<code>exp(x)</code>	Returns e^{**x}
<code>expm1(x)</code>	Returns $e^{**x} - 1$
<code>log(x[, base])</code>	Returns the logarithm of x to the base (defaults to e)
<code>log1p(x)</code>	Returns the natural logarithm of 1+x
<code>log2(x)</code>	Returns the base-2 logarithm of x
<code>log10(x)</code>	Returns the base-10 logarithm of x
<code>pow(x, y)</code>	Returns x raised to the power y
<code>sqrt(x)</code>	Returns the square root of x

<code>acos(x)</code>	Returns the arc cosine of x
<code>asin(x)</code>	Returns the arc sine of x
<code>atan(x)</code>	Returns the arc tangent of x
<code>atan2(y, x)</code>	Returns $\text{atan}(y / x)$
<code>cos(x)</code>	Returns the cosine of x
<code>hypot(x, y)</code>	Returns the Euclidean norm, $\sqrt{x^2 + y^2}$
<code>sin(x)</code>	Returns the sine of x
<code>tan(x)</code>	Returns the tangent of x
<code>degrees(x)</code>	Converts angle x from radians to degrees
<code>radians(x)</code>	Converts angle x from degrees to radians
<code>acosh(x)</code>	Returns the inverse hyperbolic cosine of x
<code>asinh(x)</code>	Returns the inverse hyperbolic sine of x
<code>atanh(x)</code>	Returns the inverse hyperbolic tangent of x
<code>cosh(x)</code>	Returns the hyperbolic cosine of x
<code>sinh(x)</code>	Returns the hyperbolic cosine of x
<code>tanh(x)</code>	Returns the hyperbolic tangent of x
<code>erf(x)</code>	Returns the error function at x
<code>erfc(x)</code>	Returns the complementary error function at x
<code>gamma(x)</code>	Returns the Gamma function at x
<code>lgamma(x)</code>	Returns the natural logarithm of the absolute value of the Gamma function at x
<code>pi</code>	Mathematical constant, the ratio of circumference of a circle to it's diameter (3.14159...)
<code>e</code>	mathematical constant e (2.71828...)