String Data Types in Hindi

Python - String Data Types

String ये Common Data Type है | ये data type सामान्यतः सभी Computer Languages में पाया जाता है |

String ये एक से ज्यादा charcaters का sequence होता है |

Example,

```
Hello Programmer ! I am a String
```

Python में String को single quotes(' ') या double quotes(" ") में लिखा जाता है |

Example,

```
'Enclosing String in single quotes'
"Enclosing String in double quotes"
```

Example for String

Source Code : Output :

```
Hello World
```

Python String Index

String की index '0' से शुरू होता है और आखिरी index '-1' होता है |

Source Code : Output :

```
H
d
```

Creating Substring from String

Python में string से substring को create करना हो तो square bracket([]) में colon(:) का इस्तेमाल किया जाता है |

Syntax for Creating Substring

सिर्फ starting index दिया जाता है तो startIndex से पूरा string display किया जाता है |

str[start_Index : end_Index(Optional default '-1')]

सिर्फ ending index दिया जाता है तो 0^{th} index से endIndex तक string display किया जाता है |

str[start_Index(Optional default '0') : end_Index]

Source Code : Output :

lo World
Hello Wo
lo Wo

String Concatenation

एक string को दुसरे string से जोड़ने के लिए '+' Operator का इस्तेमाल किया जाता है |

Example को देखकर फर्क देखिये |

Source Code :

```
str1 = "Hello World"
str2 = "Hello Friends"
str3 = str1,str2
print(str3) #Output:('Hello World', 'Hello Friends')
print(str1,str2) #Output:Hello World Hello Friends
print(str1+str2) #Output:Hello WorldHello Friends
```

Output :

('Hello World', 'Hello Friends')
Hello World Hello Friends
Hello WorldHello Friends

Python Escape Characters

Python में नीचे दिए हुए सभी Escape Sequnces है |

Escape Characters	Description	Example/Output
\v	Vertical Tab	

Use Triple single quotes(''' ''') OR double quotes(""" """)

triple single या double quote का इस्तेमाल सिर्फ multiline string और docstring के लिए किया जाता है |

Source Code : Output :

```
Hello
        World
Hello
        World
```

Working with Mixed Data Type String

सिर्फ string को ही concatenate किया जा सकता है |

Source Code : Output :

```
12
3
    print('1'+2)
TypeError: must be str, not int
```

Old Way String Formatting(C-Style)

Python String Formatting with Format Specifiers(Modulus Operator)

सामान्यतः विशेष रूप से C Programming और आदि में Formatting का इस्तेमाल किया जाता है | वहा पर printf() function का इस्तेमाल किया जाता है |

Python में भी print function में Format Specifiers का इस्तेमाल किया जाता है |

Format Specifier	Description
%c	character
%d	signed Integer
%e	lowercase exponential notation
%E	uppercase exponential notation
%f	floating point number

%g	%e and %f shorter
%G	%E and %f shorter
%i	signed Integer
%o	octal Integer
%s	String
%u	unsigned Integer
%x	lowercase hexadecimal Integer
%X	uppercase hexadecimal Integer

How to Use Format Specifiers(%c, %d, %s etc)

Format Specifier का इस्तेमाल print() function में किया जाता है | print function में left hand side में format string को और right hand side में tuple का इस्तेमाल किया जाता है |

Example,

```
print("String : %s" % ("Hello World")) #Output : String : Hello World
```

"String : %s" : Format String

%s : Format Specifier

% : Modulus Operator

("Hello World") : tuple have only one element

%c(character)

Single Character के लिए '%c' इस format specifier का इस्तेमाल किया जाता है |

%d(signed Integer)

यहाँ numeric value होती है लेकिन अपूर्णांकित हिस्सा नहीं होता है | अगर floating-point number होता है तो उसे Integer में convert किया जाता है |

```
a = 4.4
print("Float to Integer : %d" % (a))
#Output : Float to Integer : 4

b = 4
print("Integer : %d" % (b))
#Output : Integer : 4
```

%e(lowercase exponential notation)

Integer या Floating-point Number को exponential notation में convert किया जाता है |

%E(uppercase exponential notation)

Integer या Floating-point Number को exponential notation में convert किया जाता है |

%f(floating-point number)

Floating-point Number के लिए '%f' format specifier का इस्तेमाल किया जाता है | अगर integer number होता है तो उसे floating-point number में convert किया जाता है |

%g(%e and %f shorter)

'%g' का इस्तेमाल number को काफी छोटे हिस्से में convert करने के लिए किया जाता है | जरुरत पड़ने पर ये number को exponential number में भी convert करता है |

%G(%E and %f shorter)

'%G' का इस्तेमाल number को काफी छोटे हिस्से में convert करने के लिए किया जाता है | जरुरत पड़ने पर ये number को exponential number में भी convert करता है |

%i(signed Integer)

यहाँ numeric value होती है लेकिन अपूर्णांकित हिस्सा नहीं होता है | अगर floating-point number होता है तो उसे Integer में convert किया जाता है |

```
a = 4.4
print("Float to Integer : %i" % (a))
#Output : Float to Integer : 4

b = 4
print("Integer : %i" % (b))
#Output : Integer : 4
```

%o(octal Integer)

Integer Number में decimal value को octal Integer में convert किया जाता है |

```
a = -10
print("Decimal to Octal : %o" % (a))
#Output : Decimal to Octal : -12

b = 10
print("Decimal to Octal : %o" % (b))
#Output : Decimal to Octal : 12
```

%s(String)

String के लिए '%s' का इस्तेमाल किया जाता है | अगर numeric value दी जाती है तो उसे String में convert किया जाता है |

```
a = -10
print("String : %s" % (a))
#Output : String : -10

b = "Hello World"
print("String : %s" % (b))
#Output : String : Hello World
```

%x(lowercase hexadecimal Integer)

Hexadecimal number के लिए '%x' का इस्तेमाल किया जाता है | Integer Number को Hexadecimal Number में convert किया जाता है |

```
a = 15
print("Integer to Hexadecimal : %x" % (a))
#Output : Integer to Hexadecimal : f
```

%X(uppercase hexadecimal Integer)

Hexadecimal number के लिए '%X' का इस्तेमाल किया जाता है | Integer Number को Hexadecimal Number में convert किया जाता है |

```
a = 15
print("Integer to Hexadecimal : %X" % (a))
#Output : Integer to Hexadecimal : F
```

Format Specifier with Placeholder

जरुरत के हिसाब से Integer या Floating-point Number को display करना हो तो placdholder का इस्तेमाल किया जाता है |

Syntax for Placeholder

%[flag][length].[precision][type]

Parts of Placeholder

% : format specifier में % का होना अनिवार्य होता है |

flag : Optional. #, +, -, 0 और space( ) ये flags दिए जाते है |

length : Optional. यहाँ पर Number की length दी जाती है |

. : Optional. अगर floating-point number होता है तो decimal point को दिया जा सकता है |

precision : Optional. decimal point के बाद कितने numbers चाहिए उन numbers की सख्या यहाँ पर दी जाती है |

type : type को देना अनिवार्य होता है | for Example d, i, f, e, E etc.

Basic Example for Placeholder

Example के पहले Statement में 45.5 ये value दी गयी है और '%5d' ये format specifier दिया गया है | पहले value को float से integer में convert किया जायेगा और बाद में उस integer की length '5' की जायेगी | यहाँ पर integer 2 digit का ही है | उस length को '5' करने के लिए पहले 3 space precede किये जायेंगे |

Example के दूसरे statement में 45.5955 ये value दी गयी है और '%5.2f' ये format specifier दिया गया है | लेकिन यहाँ पर length से ज्यादा महत्व precision को दिया गया है |

Source Code :

Flags

String Format Specifier Flags

Flags	Description
#	%o, %x और %X के बीच में '#' flag का इस्तेमाल किया जाता है	
+	ये एक sign character है	अगर space precede होता है तो उसे '+' sign से replace किया जाता है	

-	ये left justification है | अगर space precede होता है तो उसे remove किया जाता है |
0	Number के left side से 0 के साथ pad किया जाता है |

Example for '#' Flag

Example में '#' flag का इस्तेमाल किया गया है \| o(octal) के साथ '#' को दिया जाता है तो '0o' precede लगाया जाता है \|

x(lowercase hexadecimal) के साथ '#' को दिया जाता है तो '0x' precede लगाया जाता है \|

X(uppercase hexadecimal) के साथ '#' को दिया जाता है तो '0X' precede लगाया जाता है \|

Source Code : Output :

```
0o55
0x2d
0X2D
```

Example for '+' Flag

Source Code : Output :

```
  +45
+45
```

Example for '-' Flag

Source Code : Output :

```
45
  45
```

Example for '0' Flag

Source Code : Output :

```
00045
  45
```

New Way String Formatting(Python-Style)

String Formatting with format() function

format() function से किसी भी प्रकार से String Formatting किया जाता है |

Python मे format() function; string formatting के लिए काफी उपयुक्त function है |

Syntax for format() function in Python

template.format(p0, p1, ..., pN, k0=v0, k1=v1, ..., kN=vN)

Parts of format() function

template : ये एक format string होता है, जिसमे एक से ज्यादा format codes(**{}**) होते है | Format codes ये output में replace होनेवाली जगह होती है |

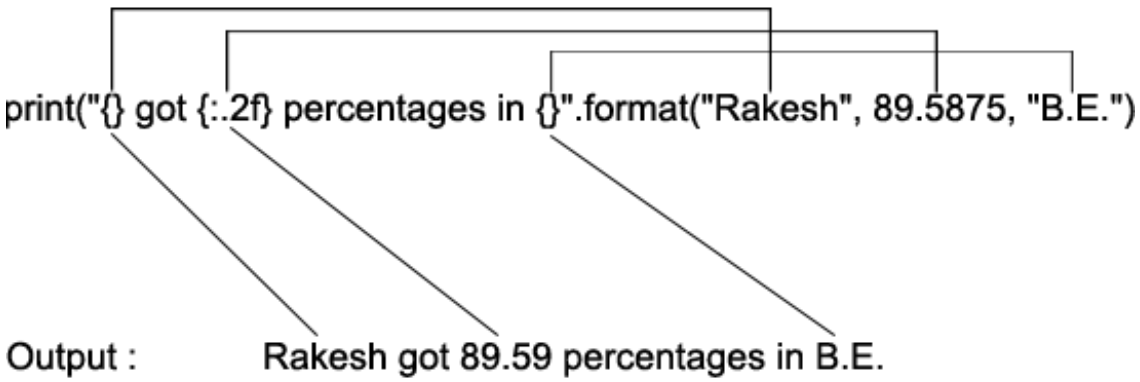
p0, p1, ..., pN : यहाँ पर ये positional parameters है | print() function में दिए गए placeholder({index}) की जगह format() पे दिए positional parameter से replace किया जाता है |

k0=v0, k1=v1, ..., kN=vN : यहाँ पर ये keyword parameters है | keyword parameter में key और value(key=value) इन दोनों की pairs होती है |print() function में दिए गए placeholder({key}) की जगह format() पे दिए keyword parameter से replace किया जाता है |

format() function ये formatted string को return करता है |

More About Positional Parameters

format() function को Example और उससे related example को समझे |



Source Code :

```
print("{} got {:.2f} percentages in {}".format("Rakesh", 89.5875, "B.E."))
```

Output :

```
Rakesh got 89.59 percentages in B.E.
```

format() function के parameters से ज्यादा placeholders({}) नहीं लिए जा सकते है | अगर लिए जाते है तो 'indexError' exception आ जाता है |

For Example,

```
print("{} got {:.2f} percentages in {}".format("Rakesh", 89.5875))
```

Output :

```
  print("{} got {:.2f} percentages in {}".format("Rakesh", 89.5875))
IndexError: tuple index out of range
```

format() function के parameters Placeholders({}) से ज्यादा लिए जा सकते है |

For Example,

```
print("{} got {:.2f} percentages".format("Rakesh", 89.5875, "B.E."))
```

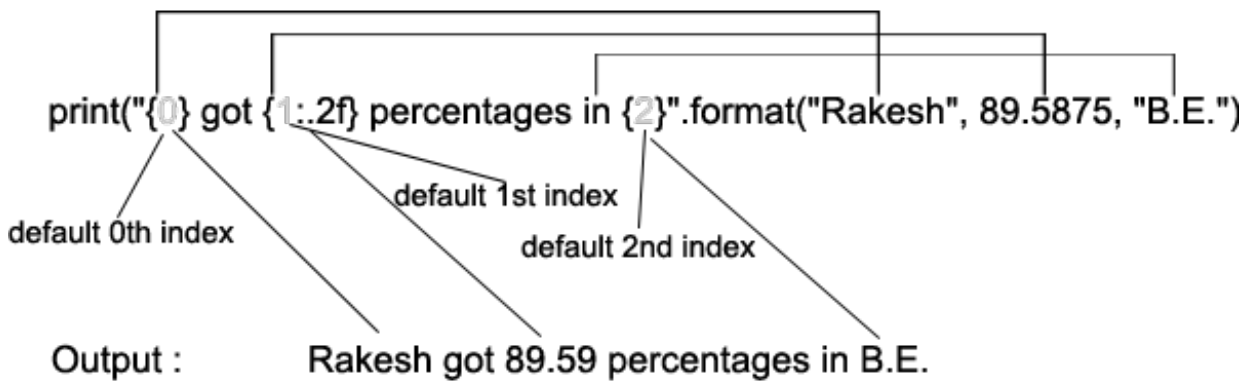
Output :

```
Rakesh got 89.59 percentages
```

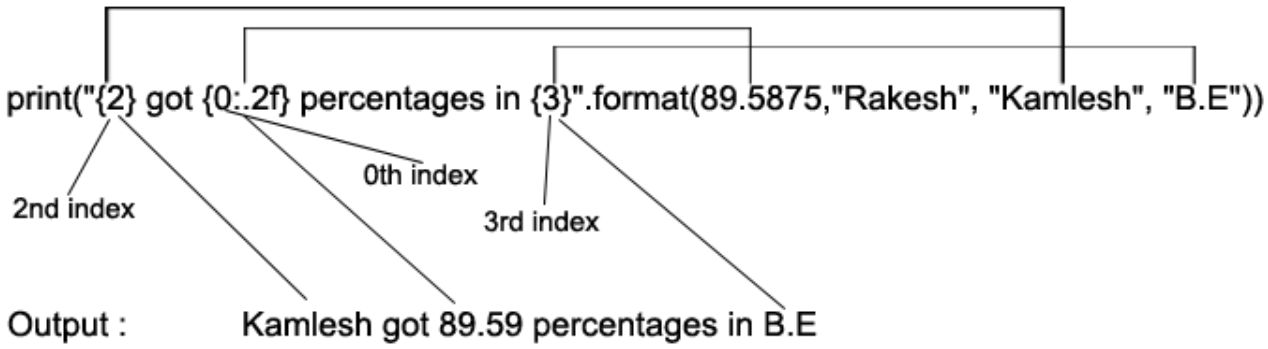
Know more About Placeholders and format() Function(Positional Parameters)

format() function को Example और उससे related example को समझे |

Placeholder में जब index दिया नहीं जाता है तो by default 0, 1, 2, 3 इस प्रकार से index interpreter द्वारा लिए जाते है | निचे image में देखके समझ आएगा |



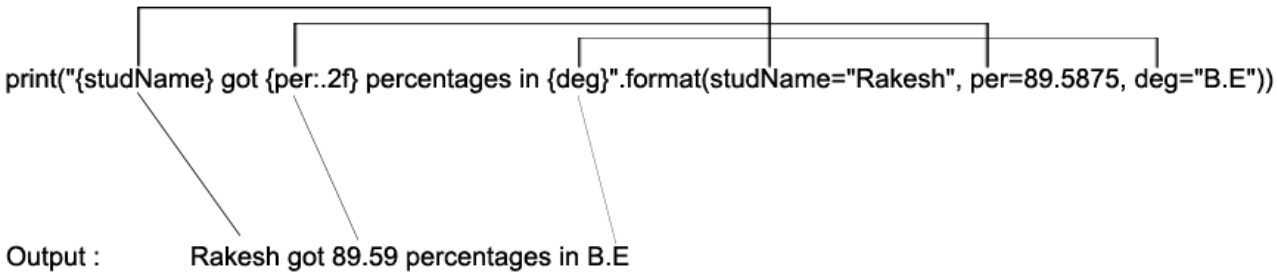
Programmer चाहे तो index को बदल भी सकता है |




```
print("{2} got {0:.2f} percentages in {3}".format(89.5875,"Rakesh", "Kamlesh", "B.E"))
#Output : Kamlesh got 89.59 percentages in B.E
```

More About Keyword Parameters

Keyword Parameters में key और value(key=value) की pairs होती है | Placeholders में key(index) के जरिये उनकी value को access किया जाता है |



Source Code :

```
print("{studName} got {per:.2f} percentages in {deg}".format(studName="Rakesh", per=89.5875, deg="B.E"))
```

Output :

Rakesh got 89.59 percentages in B.E

Using Format Specifier in Placeholders with format() Function

Format Specifier	Meaning
b	Binary
d	Integer
e	Exponential Notation(Lowercase)

E	Exponential Notation(Uppercase)
f	Floating-point (Lowercase inf, nan)
F	floating-point (Uppercase INF, NAN)
g	अगर number exponent(e) होने की बारी आती है तो decimal point के बाद सिर्फ 4 ही digit लिए जाते है | (like Lowercase 'e')
G	अगर number exponent(e) होने की बारी आती है तो decimal point के बाद सिर्फ 4 ही digit लिए जाते है | (Like Uppercase 'e')
o	Octal
s	String
x	hexadecimal(Lowercase)
X	Hexadecimal(Uppercase)

b(Binary)

Decimal को Binary में convert करने के लिए उपयोगी होता है |

For Example,

d(Integer)

Decimal को Binary में convert करने के लिए उपयोगी होता है |

For Example,

e(Lowercase Exponential Notation)

दिए गए Integer या Floating-point Number को lowercase exponential notation में convert करता है |

For Example,

E(Uppercase Exponential Notation)

दिए गए Integer या Floating-point Number को Uppercase exponential notation में convert करता है |

For Example,

f(floating-point Number)

Integer को Floating-point Number में convert कर सकता है | decimal point के बाद सिर्फ '6' digit ही लेता है |

For Example,

F(Floating-point Number)

Integer को Floating-point Number में convert कर सकता है | decimal point के बाद सिर्फ '6' digit ही लेता है |

For Example,

g(Like lowercase 'e')

For Example,

G(Like Uppercase 'E')

For Example,

o(Octal)

Decimal को octal में conver करने के लिए इस्तेमाल किया जाता है |

For Example,

s(String)

String के लिए इस्तेमाल किया जाता है |

For Example,

x(lowercase hexadecimal)

Decimal को Hexadecimal(lowercase) number में convert किया जाता है |

For Example,

X(Uppercase Hexadecimal)

Decimal को Hexadecimal(Uppercase) number में convert किया जाता है |

For Example,

String Functions in Python

String Function	Description
len()	string की length को return किया जाता है	
max()	String में से max character को return किया जाता है	
min()	String में से min character को return किया जाता है	

All String Functions in Python

String Method	Description
capitalize()	String के पहले word के पहले character को uppercase में convert किया जाता है	
center()	किसी विशिष्ट character से padded किये गए string को return करता है	
casefold()	normal string को casefold string में convert करता है	
count()	मुख्य string में से substring के occurrences; number में return किये जाते है	
endswith()	दिए गए suffix को string के end पर check करके boolean value return करता है	
expandtabs()	String में tab(s) की size को expand करके string की copy return की जाती है	

find()	substring को मुख्य string में ढूंढकर उसका पहला index return किया जाता है |
format()	इसका इस्तेमाल string formatting के लिए किया जाता है |
index()	दिए गए substring को मुख्य string में ढूंढकर उसका पहला index return किया जाता है |
isalnum()	अगर character या string alphanumeric या alphabetic या numeric होता है तो true return करता है अगर नहीं होते है तो false return होता है |
isalpha()	अगर character या string alphabetic होता है तो true return करता है अगर नहीं होते है तो false return होता है |
isdecimal()	अगर character या string decimal होता है तो true return करता है अगर नहीं होते है तो false return होता है |
isdigit()	अगर character या string digit होता है तो true return करता है अगर नहीं होते है तो false return होता है |
isidentifier()	दिए गए string या character एक valid identifier है या नहीं ये boolean value में return किया जाता है |
islower()	दिए गए string या character lowercase में है या नहीं ये check करके boolean value में return किया जाता है |
isnumeric()	दिए गए string या character numeric में है या नहीं ये check करके boolean value में return किया जाता है |
isprintable()	दिए गए string या character printable है या नहीं ये check करके boolean value में return किया जाता है |
isspace()	दिए गए string या character सिर्फ space है या नहीं ये check करके boolean value में return किया जाता है |
istitle()	दिए गया string; title string है या नहीं ये check करके boolean value में return किया जाता है |
isupper()	दिए गए string या character uppercase में है या नहीं ये check करके boolean value में return किया जाता है |
join()	दिए गए elements के sequence को किसी seperator से join करके string को return किया जाता है |
ljust()	string को left में justfy करके और width के हिसाब से दिए गए character से fill करके string को return किया जाता है |
rjust()	string को right में justfy करके और width के हिसाब से दिए गए character से fill करके string को return किया जाता है |
lower()	Uppercase के character या string को lowercase में convert किया जाता है |
upper()	lowercase के character या string को Uppercase में convert किया जाता है |
swapcase()	अगर string के characters; uppercase हो तो उसे lowercase में return करता है और lowercase हो तो उसे uppercase में return करता है |

lstrip()	left side(leading character) से दिए गए character को strip करके string की copy return की जाती है |
rstrip()	right side(trailing characters) से दिए गए character को strip करके string की copy return की जाती है |
strip()	left side(leading characters) और right side(trailing character) से दिए गए character को strip करके string की copy return की जाती है |
partition()	दिए गए String के लिए seperator का first occurrence को tuple के बीच में रख के tuple को return किया जाता है |
rpartition()	दिए गए String के लिए seperator का last occurrence को tuple के बीच में रख के tuple को return किया जाता है |
replace()	String के लिए दिए गए old substring के सभी occurrence नए substring से replace करके string की copy return की जाती है |
rfind()	दिए गए substring को मुख्य string में ढूंढकर उसका आखिरी index return किया जाता है |
rindex()	दिए गए substring को मुख्य string में ढूंढकर उसका आखिरी index return किया जाता है |
split()	दिए गए seperator से string को split करके list(sequence) में return किया जाता है |
rspilt()	दिए गए seperator से string को right side से split करके list(sequence) में return किया जाता है |
startswith()	दिए गए prefix को string के end पर check करके boolean value return करता है |
title()	Normal string को title string में convert करके return करता है |
zfill()	अगर string से ज्यादा length(width) दी जाती है तो left side से अतिरिक्त जगह पर '0' दिया जाता है और string की copy return की जाती है |
len()	string की length को return किया जाता है |
max()	String में से max character को return किया जाता है |
min()	String में से min character को return किया जाता है |