



EXPLORER

User Settings

percipio08\_Strings.py

OPEN EDITORS 1 UNSAVED

User Settings C:\Us...

percipio08\_Strings...

PYTHON

Automate-Boring-Stuff

my\_code

Percipio\_Python3-Cour...

01\_Start

02\_Data-Sequence T...

percipio04\_int\_typ...

percipio05\_float\_ty...

percipio06\_math\_f...

percipio07\_boolea...

percipio08\_Strings...

percipio09\_float\_ty...

percipio10\_bytes\_t...

percipio11\_bytearr...

percipio12\_list\_typ...

percipio13\_tuple\_t...

percipio14\_slice\_ty...

percipio14a\_list\_co...

03\_Collections-Mapp..

04\_Modules-Function...

05\_Classes

06\_Working-with-Files

07\_Comprehensions

percipio45\_list\_co...

percipio46\_nested...

percipio47\_zip\_fun...

percipio48\_set\_co...

percipio49\_diction...

08\_Iterables-and-Ge...

09\_Exceptions

Python Projects\_2014

CMD\_Python\_Set-Path...

Python\_Clear-Window-...

python\_exercises\_00.py

python\_exercises\_01.py

Python\_Tutorial\_Runni...

Python\_Tutorials.md

```
1 '''
2 percipio08_string_type.py
3 Percipio video: Data & Sequence Types; The str Type in Python
4 This code file shows demonstrations on how to work with strings
5 Strings of class str are created with quotes
6 '''
7 nl = '\n'
8 quote_1 = 'single quoted'
9 quote_2 = "double quoted"
10 print('quote_1, quote_2 =\n', quote_1 + ' or ' + quote_2 + '\nresulting in the same output')
11 print(nl)
12 why_1 = 'She said, "Hello!"'
13 why_2 = "It's mine!"
14 print('why_1, why_2 =')
15 print(why_1, why_2)
16 why_not_1 = "she said, \"Hello!\""
17 why_not_2 = 'It\'s mine!'
18 print('why_not_1, why_not_2 =')
19 print(why_not_1, why_not_2)
20 print('Above shows 2 different ways to result with the same string')
21 print(nl, 'Next')
22 # Special escape sequences exist
23 new_line = '\nline1\nline2\nline3'
24 print('Create new lines by typing \\n =', new_line)
25 print(nl, 'Next')
26 tab_char = 'col1\tcol2\tcol3\t'
27 print('Create tabbed spacing by typing \\t =', tab_char)
28 print(nl, 'Next')
29 backslash = 'Print 1 backslash by typing 2 backslashes: \\'
30 print(backslash)
31 print(nl, 'Next')
32 # Raw strings prevent escape sequence interpretation printing exactly what is typed
33 raw_new_line = r'line1\nline2\nline3\n'
34 print('Raw string printed exactly as typed by adding "r" in front of string =', raw_new_line)
35 raw_tab_char = r'col1\tcol2\tcol3\t'
36 print('Raw string tabbed =', raw_tab_char)
37 raw_backslash = r'the backslash: \\'
38 print('Raw string with backslashes =', raw_backslash)
39 print(nl, 'Next')
40 # Sequence objects can use several operators and functions:
```





EXPLORER

1 OPEN EDITORS 1 UNSAVED

- User Settings C:\Us...
- percipio08\_Strings...

PYTHON

- Automate-Boring-Stuff
- my\_code
- Percipio\_Python3-Cour...
- 01\_Start
- 02\_Data-Sequence T...
- percipio04\_int\_typ...
- percipio05\_float\_ty...
- percipio06\_math\_f...
- percipio07\_boolea...
- percipio08\_Strings...
- percipio09\_float\_ty...
- percipio10\_bytes\_t...
- percipio11\_bytearr...
- percipio12\_list\_typ...
- percipio13\_tuple\_t...
- percipio14\_slice\_ty...
- percipio14a\_list\_co...
- 03\_Collections-Mapp..
- 04\_Modules-Functio...
- 05\_Classes
- 06\_Working-with-Files
- 07\_Comprehensions
- percipio45\_list\_co...
- percipio46\_nested...
- percipio47\_zip\_fun...
- percipio48\_set\_co...
- percipio49\_diction...
- 08\_Iterables-and-Ge...
- 09\_Exceptions
- Python Projects\_2014
- CMD\_Python\_Set-Path...
- Python\_Clear-Window...
- python\_exercises\_00.py
- python\_exercises\_01.py
- Python\_Tutorial\_Runni...
- Python\_Tutorials.md

User Settings

percipio08\_Strings.py

```
40 # Sequence objects can use several operators and functions:
41 sub_text = 'double'
42 print('sub_text =', sub_text)
43 print('sub_text + sub_text =', sub_text + sub_text)
44 print("'_' * 20 will multiply any character by any number =", '_' * 20)
45 print(nl, 'Next')
46 print('len(sub_text) returns string length =', len(sub_text)) # Length function telling how long a
    string is
47 print('min(sub_text) returns lowest unicode table value =', min(sub_text)) # Min function returns
    lowest unicode table value
48 print('max(sub_text) returns highest unicode table value =', max(sub_text)) # Max function returns
    highest unicode table value
49 print(nl, 'Next')
50 # Boolean resulting "in" or "not in" functions testing if a string is within, or not, another string
51 print('Boolean resulting "in" or "not in" functions testing if a string is within, or not, another
    string')
52 print('sub_text not in quote_1 =', sub_text not in quote_1)
53 print('sub_text in quote_2 =', sub_text in quote_2)
54 print(nl, 'Next')
55 # String objects have many available methods like:
56 print('why_1.count("e") returns number of "e"s within string =', why_1.count('e')) # count number
    of 'e's within string
57 print('why_1.index("e") returns character position =', why_1.index('e')) # check the index or
    position within a string
58 print('why_1.index("e", 3, 18) returns character position within a positional range =', why_1.index
    ('e', 3, 18)) # search at a particular start/stop position
59 print('Caution: Index method returns an error if character is not found.\n\tFind method is safer
    returning a "-1" if not found')
60 print('why_1.find("X") finds a string within a string OR returns -1 =', why_1.find('X')) # Find a
    string within a string OR returns -1
61 print(nl, 'Next')
62 print('why_1.startswith("She") returns boolean determining if string starts with indicator =',
    why_1.startswith('She')) # Boolean test determining if string starts with
63 print('why_1.endswith("!\\\"") returns boolean determining if string ends with indicator =',
    why_1.endswith("!\\\"")) # Boolean test determining if string ends with
64 print(nl, 'Next')
65 print('why_1.upper() returns all upper case =', why_1.upper()) # changes case to upper
66 print('why_1.lower() returns all lower case =', why_1.lower()) # changes case to lower
67 print(nl, 'Next')
68 csv = 'a,b,c' # new string created for these examples
```





EXPLORER

1 OPEN EDITORS 1 UNSAVED

- User Settings C:\Us...
- percipio08\_Strings...

PYTHON

- Automate-Boring-Stuff
- my\_code
- Percipio\_Python3-Cour...
- 01\_Start
- 02\_Data-Sequence T...
- percipio04\_int\_typ...
- percipio05\_float\_ty...
- percipio06\_math\_f...
- percipio07\_boolea...
- percipio08\_Strings...
- percipio09\_float\_ty...
- percipio10\_bytes\_t...
- percipio11\_bytearr...
- percipio12\_list\_typ...
- percipio13\_tuple\_t...
- percipio14\_slice\_ty...
- percipio14a\_list\_co...
- 03\_Collections-Mapp..
- 04\_Modules-Function...
- 05\_Classes
- 06\_Working-with-Files
- 07\_Comprehensions
- percipio45\_list\_co...
- percipio46\_nested...
- percipio47\_zip\_fun...
- percipio48\_set\_co...
- percipio49\_diction...
- 08\_Iterables-and-Ge...
- 09\_Exceptions
- Python Projects\_2014
- CMD\_Python\_Set-Path...
- Python\_Clear-Window-..
- python\_exercises\_00.py
- python\_exercises\_01.py
- Python\_Tutorial\_Runni...

```
User Settings
percipio08_Strings.py

68 csv = 'a,b,c' # new string created for these examples
69 print('csv string =', csv)
70 print('csv.split(",") returns multiple strings split at indicated character =', csv.split(",") #
    returns sub-strings split at indicated character, not including that character; csv-string
    split-method on a comma
71 print('",".join(["a", "b", "c"]) returns strings joined together divided by indicated character =',
    ",".join(["a", "b", "c"])) # joins strings together using indicated character ","
72 print(nl, 'Next')
73 # Python has many inspection functions such as 'isthis' and 'isthat'
74 print('sub_text.isalpha() returns boolean if string is all alphabetic characters =',
    sub_text.isalpha()) # Boolean returning True if string is all alphabetic characters
75 print('sub_text.isdigit() returns boolean if string is all numeric characters =', sub_text.isdigit()
    ) # Boolean returning True if string is all numeric characters (0 to 9)
76 '''
77 RESULTS:
78 quote_1, quote_2 =
79 | single quoted or double quoted
80 | resulting in the same output
81
82
83 why_1, why_2 =
84 She said, "Hello!" It's mine!
85 why_not_1, why_not_2 =
86 she said, "Hello!" It's mine!
87 Above shows 2 different ways to result with the same string
88
89 | Next
90 | Create new lines by typing \n =
91 | line1
92 | line2
93 | line3
94
95 | Next
96 | Create tabbed spacing by typing \t = col1    col2    col3
97
98 | Next
99 | Print 1 backslash by typing 2 backslashes: \
100
101 | Next
```





EXPLORER

{ } User Settings

percipio08\_Strings.py •

OPEN EDITORS 1 UNSAVED

{ } User Settings C:\Us...

percipio08\_Strings...

PYTHON

Automate-Boring-Stuff

my\_code

Percipio\_Python3-Cour...

01\_Start

02\_Data-Sequence T...

percipio04\_int\_typ...

percipio05\_float\_ty...

percipio06\_math\_f...

percipio07\_boolea...

percipio08\_Strings...

percipio09\_float\_ty...

percipio10\_bytes\_t...

percipio11\_bytearr...

percipio12\_list\_typ...

percipio13\_tuple\_t...

percipio14\_slice\_ty...

percipio14a\_list\_co...

03\_Collections-Mapp..

04\_Modules-Function...

05\_Classes

06\_Working-with-Files

07\_Comprehensions

percipio45\_list\_co...

percipio46\_nested

101

Next

102

Raw string printed exactly as typed by adding "r" in front of string = line1\nline2\nline3\n

103

Raw string tabbed = col1\tcol2\tcol3\t

104

Raw string with backslashes = the backslash: \\

105

106

Next

107

sub\_text = double

108

sub\_text + sub\_text = doubledouble

109

'\_' \* 20 will multiply any character by any number = \_\_\_\_\_

110

111

Next

112

len(sub\_text) returns string length = 6

113

min(sub\_text) returns lowest unicode table value = b

114

max(sub\_text) returns highest unicode table value = u

115

116

Next

117

Boolean resulting "in" or "not in" functions testing if a string is within, or not, another string

118

sub\_not in quote\_1 = True

119

sub\_text in quote\_2) = True

120

121

Next

122

why\_1.count("e") returns number of "e"s within string = 2

123

why\_1.index("e") returns character position = 2

124

why\_1.index("e", 3, 18) returns character position within a positional range = 12

125

Caution: Index method returns an error if character is not found.

126

Find method is safer returning a "-1" if not found

127

why\_1.find("X") finds a string within a string OR returns -1 = -1





EXPLORER

{ } User Settings

percipio08\_Strings.py •



## OPEN EDITORS 1 UNSAVED

{ } User Settings C:\Us...

percipio08\_Strings....

## PYTHON

Automate-Boring-Stuff

my\_code

Percipio\_Python3-Cour...

01\_Start

02\_Data-Sequence T...

percipio04\_int\_typ...

percipio05\_float\_ty...

percipio06\_math\_f...

percipio07\_boolea...

percipio08\_Strings...

percipio09\_float\_ty...

percipio10\_bytes\_t...

percipio11\_bytearr...

percipio12\_list\_typ...

percipio13\_tuple\_t...

percipio14\_slice\_ty...

127  
128  
129  
130  
131  
132  
133  
134  
135  
136  
137  
138  
139  
140  
141  
142  
143  
144  
145`why_1.find("X")` finds a string within a string OR returns -1 = -1

Next

`why_1.startswith("She")` returns boolean determining if string starts with indicator = True`why_1.endswith("!\"")` returns boolean determining if string ends with indicator = True

Next

`why_1.upper()` returns all upper case = SHE SAID, "HELLO!"`why_1.lower()` returns all lower case = she said, "hello!"

Next

`csv string = a,b,c``csv.split(",")` returns multiple strings split at indicated character = ['a', 'b', 'c']`",".join(["a", "b", "c"])` returns strings joined together divided by indicated character = a,b,c

Next

`sub_text.isalpha()` returns boolean if string is all alphabetic characters = True`sub_text.isdigit()` returns boolean if string is all numeric characters = False

...

```
1 # Import the csv module
2 import csv
3
4 # Create a list of the rows in the csv file
5 rows = csv.reader(open('data.csv'))
6
7 # Print out the first row
8 print(rows[0])
9
10 # Print out the second row
11 print(rows[1])
12
13 # Print out the third row
14 print(rows[2])
15
16 # Print out the fourth row
17 print(rows[3])
18
19 # Print out the fifth row
20 print(rows[4])
21
22 # Print out the sixth row
23 print(rows[5])
24
25 # Print out the seventh row
26 print(rows[6])
27
28 # Print out the eighth row
29 print(rows[7])
30
31 # Print out the ninth row
32 print(rows[8])
33
34 # Print out the tenth row
35 print(rows[9])
36
37 # Print out the eleventh row
38 print(rows[10])
39
40 # Print out the twelfth row
41 print(rows[11])
42
43 # Print out the thirteenth row
44 print(rows[12])
45
46 # Print out the fourteenth row
47 print(rows[13])
48
49 # Print out the fifteenth row
50 print(rows[14])
51
52 # Print out the sixteenth row
53 print(rows[15])
54
55 # Print out the seventeenth row
56 print(rows[16])
57
58 # Print out the eighteenth row
59 print(rows[17])
60
61 # Print out the nineteenth row
62 print(rows[18])
63
64 # Print out the twentieth row
65 print(rows[19])
66
67 # Print out the twenty-first row
68 print(rows[20])
69
70 # Print out the twenty-second row
71 print(rows[21])
72
73 # Print out the twenty-third row
74 print(rows[22])
75
76 # Print out the twenty-fourth row
77 print(rows[23])
78
79 # Print out the twenty-fifth row
80 print(rows[24])
81
82 # Print out the twenty-sixth row
83 print(rows[25])
84
85 # Print out the twenty-seventh row
86 print(rows[26])
87
88 # Print out the twenty-eighth row
89 print(rows[27])
90
91 # Print out the twenty-ninth row
92 print(rows[28])
93
94 # Print out the thirtieth row
95 print(rows[29])
96
97 # Print out the thirty-first row
98 print(rows[30])
99
100 # Print out the thirty-second row
101 print(rows[31])
102
103 # Print out the thirty-third row
104 print(rows[32])
105
106 # Print out the thirty-fourth row
107 print(rows[33])
108
109 # Print out the thirty-fifth row
110 print(rows[34])
111
112 # Print out the thirty-sixth row
113 print(rows[35])
114
115 # Print out the thirty-seventh row
116 print(rows[36])
117
118 # Print out the thirty-eighth row
119 print(rows[37])
120
121 # Print out the thirty-ninth row
122 print(rows[38])
123
124 # Print out the fortieth row
125 print(rows[39])
126
127 # Print out the forty-first row
128 print(rows[40])
129
130 # Print out the forty-second row
131 print(rows[41])
132
133 # Print out the forty-third row
134 print(rows[42])
135
136 # Print out the forty-fourth row
137 print(rows[43])
138
139 # Print out the forty-fifth row
140 print(rows[44])
141
142 # Print out the forty-sixth row
143 print(rows[45])
144
145 # Print out the forty-seventh row
146 print(rows[46])
147
148 # Print out the forty-eighth row
149 print(rows[47])
150
151 # Print out the forty-ninth row
152 print(rows[48])
153
154 # Print out the fiftieth row
155 print(rows[49])
156
157 # Print out the fifty-first row
158 print(rows[50])
159
160 # Print out the fifty-second row
161 print(rows[51])
162
163 # Print out the fifty-third row
164 print(rows[52])
165
166 # Print out the fifty-fourth row
167 print(rows[53])
168
169 # Print out the fifty-fifth row
170 print(rows[54])
171
172 # Print out the fifty-sixth row
173 print(rows[55])
174
175 # Print out the fifty-seventh row
176 print(rows[56])
177
178 # Print out the fifty-eighth row
179 print(rows[57])
180
181 # Print out the fifty-ninth row
182 print(rows[58])
183
184 # Print out the sixtieth row
185 print(rows[59])
186
187 # Print out the sixty-first row
188 print(rows[60])
189
190 # Print out the sixty-second row
191 print(rows[61])
192
193 # Print out the sixty-third row
194 print(rows[62])
195
196 # Print out the sixty-fourth row
197 print(rows[63])
198
199 # Print out the sixty-fifth row
200 print(rows[64])
201
202 # Print out the sixty-sixth row
203 print(rows[65])
204
205 # Print out the sixty-seventh row
206 print(rows[66])
207
208 # Print out the sixty-eighth row
209 print(rows[67])
210
211 # Print out the sixty-ninth row
212 print(rows[68])
213
214 # Print out the seventieth row
215 print(rows[69])
216
217 # Print out the seventy-first row
218 print(rows[70])
219
220 # Print out the seventy-second row
221 print(rows[71])
222
223 # Print out the seventy-third row
224 print(rows[72])
225
226 # Print out the seventy-fourth row
227 print(rows[73])
228
229 # Print out the seventy-fifth row
230 print(rows[74])
231
232 # Print out the seventy-sixth row
233 print(rows[75])
234
235 # Print out the seventy-seventh row
236 print(rows[76])
237
238 # Print out the seventy-eighth row
239 print(rows[77])
240
241 # Print out the seventy-ninth row
242 print(rows[78])
243
244 # Print out the eightieth row
245 print(rows[79])
246
247 # Print out the eighty-first row
248 print(rows[80])
249
250 # Print out the eighty-second row
251 print(rows[81])
252
253 # Print out the eighty-third row
254 print(rows[82])
255
256 # Print out the eighty-fourth row
257 print(rows[83])
258
259 # Print out the eighty-fifth row
260 print(rows[84])
261
262 # Print out the eighty-sixth row
263 print(rows[85])
264
265 # Print out the eighty-seventh row
266 print(rows[86])
267
268 # Print out the eighty-eighth row
269 print(rows[87])
270
271 # Print out the eighty-ninth row
272 print(rows[88])
273
274 # Print out the ninetieth row
275 print(rows[89])
276
277 # Print out the ninety-first row
278 print(rows[90])
279
280 # Print out the ninety-second row
281 print(rows[91])
282
283 # Print out the ninety-third row
284 print(rows[92])
285
286 # Print out the ninety-fourth row
287 print(rows[93])
288
289 # Print out the ninety-fifth row
290 print(rows[94])
291
292 # Print out the ninety-sixth row
293 print(rows[95])
294
295 # Print out the ninety-seventh row
296 print(rows[96])
297
298 # Print out the ninety-eighth row
299 print(rows[97])
300
301 # Print out the ninety-ninth row
302 print(rows[98])
303
304 # Print out the hundredth row
305 print(rows[99])
```