

Problem 1: string comparison

Hint: the following is True: "" < "0" < "9" < "A" < "Z" < "a" < "z"

Circle the expressions that are True:

"a" < "z"	"ax" < "ay"	"abc" < "abCd"
"a" < "Z"	"x2" < "x1"	"zero" < "999"
"x" < "x"	"abcX" < "abcY"	"10" < "999"
"0" < "x"	"abcX" < "aBcY"	"1000" < "999"
"1" < "0"	"abc" < "abcd"	"888888888888" < "9"

Problem 2: string functions

Functions: upper, lower, strip, rstrip, lstrip, format, startswith, endswith, find.

Expression:	Value (put in quotes):	Expression:	Value
"dog".upper()		"abcd".startswith("ab")	
"Dog".lower()		"abcd".endswith("bc")	
" paint ".strip()		"abcd".find("a")	
" paint ".rstrip()		"abcd".find("c")	
"val: {}".format(99)		"abcd".find("B")	
"{} {}".format("X", "Y")		"Python".find("th")	

Problem 3: sequence indexing

Assume **msg** is "Hello" and **x** is "num= 13". Some expressions cause an error.

Expression	Result	Expression	Result	Expression	Result
"abc"[0]		msg[4]		x[len(x) - 1]	
"abc"[2]		msg[5]		x[3]	
"abc"[-1]		msg[len(msg)]		x[1] + x[2]	

Problem 4: sequence slicing

Assume `msg` and `x` are as before, and `p` is `"= "`.

Expression	Result
<code>"abcde"[0:2]</code>	
<code>"abcde"[2:6]</code>	
<code>"abcde"[2:9]</code>	

Expression	Result
<code>msg[:2]</code>	
<code>msg[2:]</code>	
<code>msg[-2:]</code>	

Expression	Result
<code>msg[:msg.find('=')]</code>	
<code>msg[msg.find(' ')+1:]</code>	
<code>msg[msg.find(p)+len(p):]</code>	

Problem 5: for loop over sequence

What does the following code print?

```
msg = "301"
```

```
A = ""
```

```
B = ""
```

```
for character in msg:
    print(msg)
    A = A + character + "."
    B = character + B
```

What is in A afterwards?

What is in B afterwards?

Problem 6: for loop over range

What does this code print?

```
s = "PYTHON"
for i in range(len(s)):
    print(s[:i+1])
```