CSC120
Sept. 29, 2017
Myrto Papadopoulou
Clarification for Lab3 (Piazza Question)

Q: “For the common_price_range part of the lab - would the function have to return the maximum amount that both people could afford individually, or the total amount that both can afford collectively?”

A: The returned string should reflect the most expensive category (a string) that both people can afford individually.

For example, if one person can spend 70 dollars and the other person can spend 20, then the returned string should be 'moderate'.
Strings

• Assume \( s = \text{“Hi all!”} \)

• How many characters does this string have?
  • 7
  • Space is a character. So is !

• String Indexing:
  • Accesses individual characters of the string. For example, \( s[1] \) accesses ‘i’

<table>
<thead>
<tr>
<th>Index 0</th>
<th>Index 1</th>
<th>Index 2</th>
<th>Index 3</th>
<th>Index 4</th>
<th>Index 5</th>
<th>Index 6</th>
</tr>
</thead>
<tbody>
<tr>
<td>H</td>
<td>i</td>
<td>a</td>
<td>!</td>
<td>!</td>
<td>!</td>
<td>!</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Index -7</th>
<th>Index -6</th>
<th>Index -5</th>
<th>Index -4</th>
<th>Index -3</th>
<th>Index -2</th>
<th>Index -1</th>
</tr>
</thead>
<tbody>
<tr>
<td>H</td>
<td>i</td>
<td>a</td>
<td>!</td>
<td>!</td>
<td>!</td>
<td>!</td>
</tr>
</tbody>
</table>
String Slicing

• Example: s[start_index:end_index]
• The slice includes all characters from start_index up to (but not including) the character at end_index.
Q:
“I tried running my last exercise e1.py by adding:

import doctest
doctest.testmod()

but it says:

Cannot read termcap database;
using dumb terminal settings.

Why is it saying that? And how do I resolve it?”

A:

It is safe to ignore this message, but here's what you can do to make it go away:

In Wing, select Edit -> Configure Python.

In the Environment option select: Add to inherited environment

and in the white box below write this: TERM=xterm-256color

Click Apply and then OK so your shell is restarted.
What is *pass* in E2? (on Piazza)

Answer: *pass* is a placeholder statement to indicate that you need to write code that is missing. Nothing happens when it is executed.

From the Python documentation here:

https://docs.python.org/3.2/reference/simple_stmts.html#the-pass-statement

*pass* "*is a null operation — when it is executed, nothing happens. It is useful as a placeholder when a statement is required syntactically, but no code needs to be executed, for example:*

```python
def f(arg): pass  # a function that does nothing (yet)
```