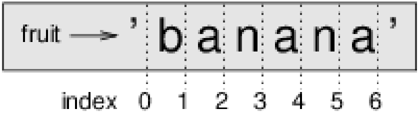
Think Python 2e, Chapter 8 Notes

Geoffrey Matthews

September 19, 2022

Strings are sequences



```
>>> fruit = ’banana’
>>> len(fruit)
6
>>> fruit[1]
’a’
>>> fruit[4]
’n’
>>> fruit[-1]
’a’
>>> fruit[1:3]
’an’
>>> fruit[2:]
’nana’
>>> fruit[:3]
’ban’
```

String traversal

```python
index = 0
while index < len(fruit):
    letter = fruit[index]
    print(letter)
    index = index + 1
```

```python
for letter in fruit:
    print(letter)
```

Strings are immutable

```
>>> greeting = 'Hello, world!'
>>> greeting[0] = 'J'
TypeError: 'str' object does not support item assignment
```

Strings are immutable

```
>>> greeting = ’Hello, world!’
>>> greeting[0] = ’J’
TypeError: ’str’ object does not support item
    assignment
```

You have to create a new string:

```
>>> greeting = ’Hello, world!’
>>> new_greeting = ’J’ + greeting[1:]
>>> new_greeting
’Jello, world!’
```

Searching

```python
def find(word, letter):
    index = 0
    while index < len(word):
        if word[index] == letter:
            return index
        index = index + 1
    return -1
```

Looping and counting

```python
word = 'banana'
count = 0
for letter in word:
    if letter == 'a':
        count = count + 1
print(count)
```

String methods

```
>>> word = ’banana’
>>> new_word = word.upper()
>>> new_word
’BANANA’
```

```
>>> word = ’banana’
>>> index = word.find(’a’)
>>> index
1
>>> word.find(’na’)
2
>>> word.find(’na’, 3)
4
>>> name = ’boba’
>>> name.find(’b’, 1, 2)
-1
```

The in operator

```
>>> ’nan’ in ’banana’
True
>>> ’seed’ in ’banana’
False
```

```
def in_both(word1, word2):
    for letter in word1:
        if letter in word2:
            print(letter)
```

```
>>> in_both(’apples’, ’oranges’)
a
e
s
```

String comparison

```python
if word == 'banana':
    print('All right, bananas.')
```

```python
if word < 'banana':
    print('Your word, '+word+', comes before banana.')
elif word > 'banana':
    print('Your word, '+word+', comes after banana.')
else:
    print('All right, bananas.')
```

```
Your word, Pineapple, comes before banana.
```

Capital letters come before lowercase.
Solution: convert all to lowercase before comparison.

Vocabulary

object: Something a variable can refer to. For now, you can use “object” and “value” interchangeably.

sequence: An ordered collection of values where each value is identified by an integer index.

item: One of the values in a sequence.

index: An integer value used to select an item in a sequence, such as a character in a string. In Python indices start from 0.

slice: A part of a string specified by a range of indices.

empty string: A string with no ch]aracters and length 0, represented by two quotation marks.

immutable: The property of a sequence whose items cannot be changed.

Vocabulary

traverse: To iterate through the items in a sequence, performing a similar operation on each.

search: A pattern of traversal that stops when it finds what it is looking for.

counter: A variable used to count something, usually initialized to zero and then incremented.

invocation: A statement that calls a method.

optional argument: A function or method argument that is not required.