Strings 2

Review

- Strings are stored character by character.
- Can access each character individually by using an index:

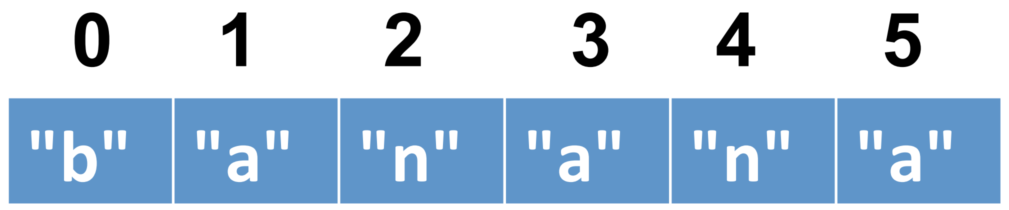
0	1	2	3	4	5	6	7
"C"	**"o"**	**"m"**	**"p"**	**"u"**	**"t"**	**"e"**	**"r"**

The basic string for loop

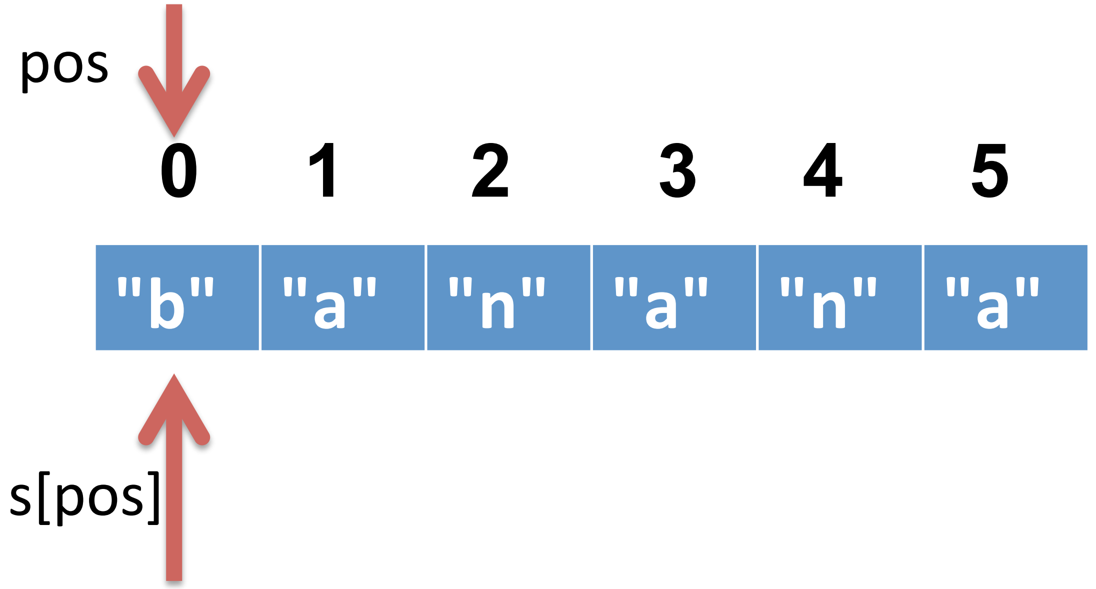
- Use this whenever you need to process a string one character at a time.

```
# assume s is a string variable
for pos in range(0, len(s)):
  # do something with s[pos]
```

```
s = "banana"
counter = 0
for pos in range(0, len(s)):
   if s[pos] == "a":
      counter = counter + 1
```

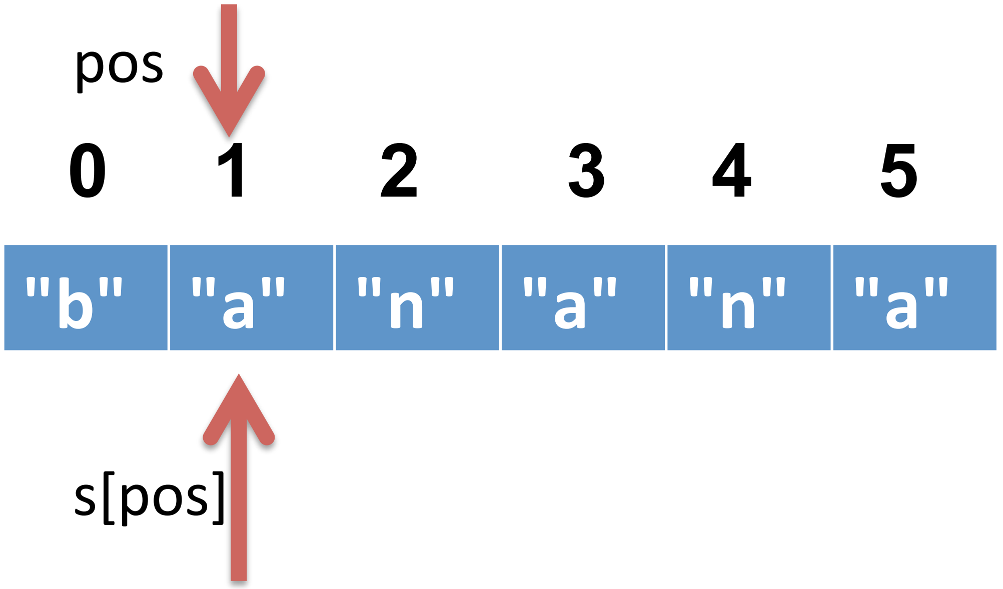


```
s = "banana"
counter = 0
for pos in range(0, len(s)):
   if s[pos] == "a":
      counter = counter + 1
```



1st iteration
pos: 0
s[pos]: "b"
counter: 0

```
s = "banana"
counter = 0
for pos in range(0, len(s)):
   if s[pos] == "a":
      counter = counter + 1
```



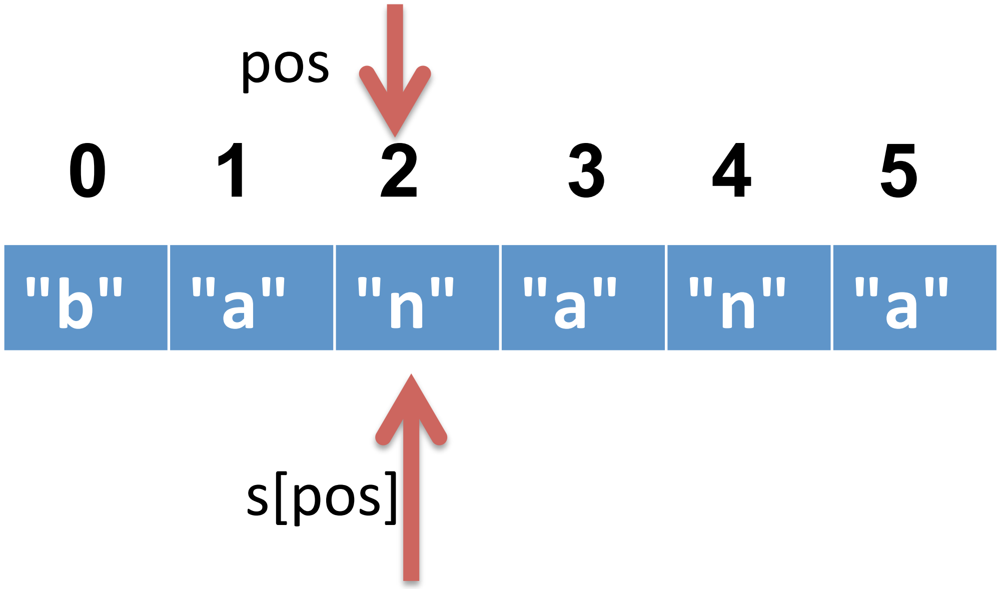
2nd iteration

pos: 1

s[pos]: "a"

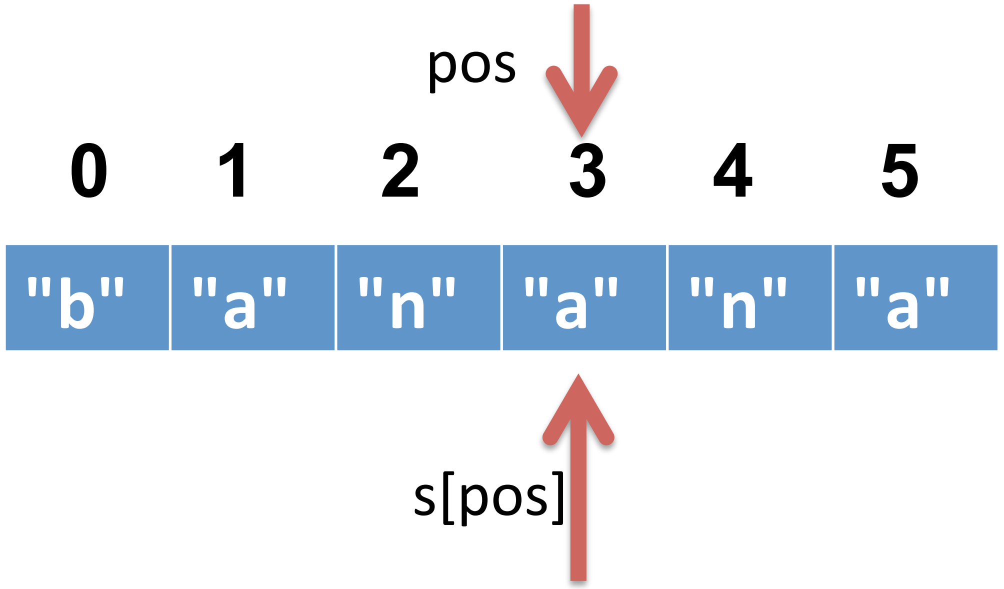
counter: 1

```
s = "banana"
counter = 0
for pos in range(0, len(s)):
   if s[pos] == "a":
      counter = counter + 1
```



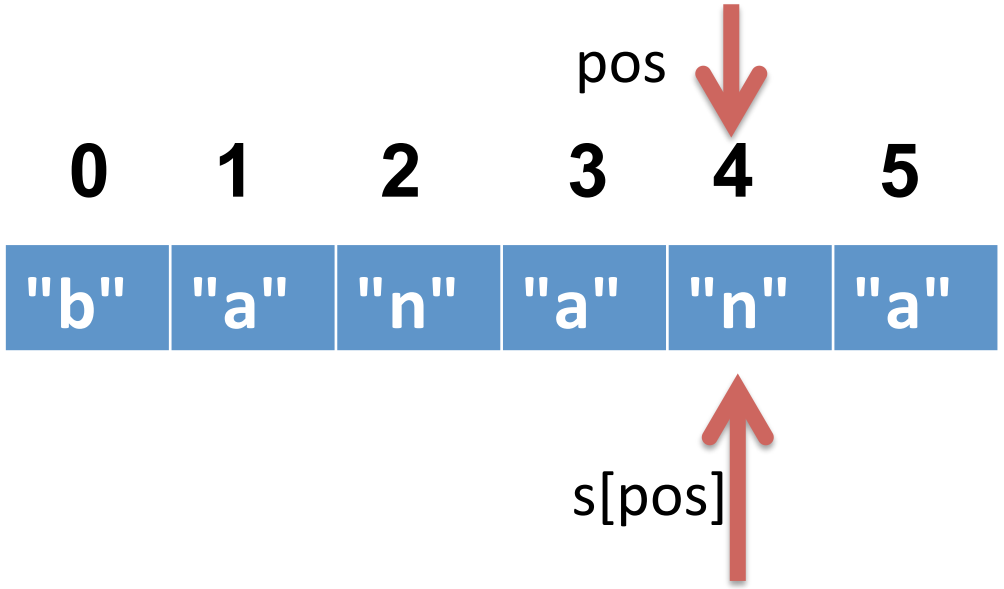
3rd iteration
pos: 2
s[pos]: "n"
counter: 1

```
s = "banana"
counter = 0
for pos in range(0, len(s)):
   if s[pos] == "a":
     counter = counter + 1
```



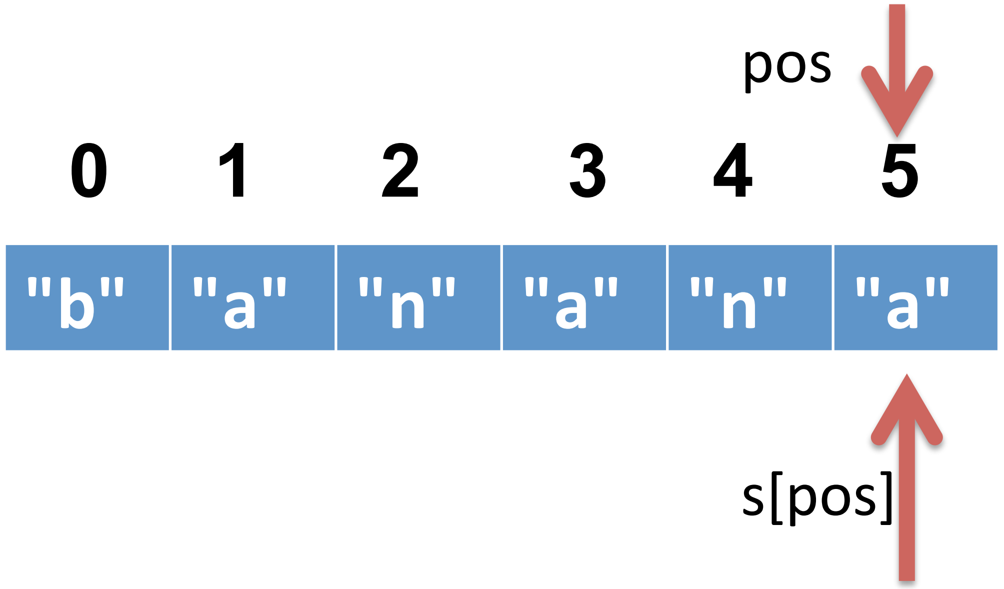
4th iteration
pos: 3
s[pos]: "a"
counter: 2

```
s = "banana"
counter = 0
for pos in range(0, len(s)):
   if s[pos] == "a":
      counter = counter + 1
```



5th iteration
pos: 4
s[pos]: "n"
counter: 2

```
s = "banana"
counter = 0
for pos in range(0, len(s)):
   if s[pos] == "a":
     counter = counter + 1
```



6th iteration
pos: 5
s[pos]: "a"
counter: 3

Algorithm -> Function

- Counting the number of a certain character in a string seems like a good candidate for a function.

```
def count_lowercase_a(s):
  counter = 0
  for pos in range(0, len(s)):
    if s[pos] == "a":
      counter = counter + 1
  return counter
```

```
def count_lowercase_a(s):
  counter = 0
  for pos in range(0, len(s)):
    if s[pos] == "a":
      counter = counter + 1
  return counter

def main():
  name = input("What is your name? ")
  freq = count_lowercase_a(name)
  print("Your name has", freq, "A's in it.")
```

You try it.

- Step 1: Change the count function so it takes a second argument called letter. The function should count the number of time that letter occurs in the string (instead of only A's).
- Step 2: Change the main program so that the user can type in their name and a letter and the program prints the frequency of that letter in their name.

For loop is not a magic bullet

- Not all string problems are solved with for loops.

```
def get_first_char(string):
  initial = string[0]
  return initial
```

- Two ways to use square brackets.
- 1 number inside -> gives you one character of a string.
  - s[0] gives you "C" # assume s = "Computer"
- 2 numbers inside -> gives you a substring or string slice.

s[a:b] gives you a substring starting from index a and ending at index b-1.

0	1	2	3	4	5	6	7
"C"	"o"	"m"	"p"	"u"	"t"	"e"	"r"

s[0:1] -> "C" just like s[0]

s[0:2] -> "Co"

s[0:7] -> "Compute"

s[3:6] -> "put"

s[0:8] -> "Computer"

Indices don't have to be literal numbers

- Say we have this code:

```
s = input("type in a string: ")
x = int(len(s) / 2)
print(s[0:x])
```

What does this print?

More fun with indices

- Indices can also be negative.
- A negative index counts from the right side of the string, rather than the left.

```
s = "Computer"
print(s[-1])           # prints r
print(s[-3:len(s)])  # prints ter
print(s[1:-1])         # prints ompute
```

More fun with indices

- Slices don't need both left and right indices.
- Missing left -> use 0 [far left of string]
- Missing right -> use len(s) [far right of string]

```
s = "Computer"
print(s[1:])            # prints omputer
print(s[:5])            # prints Compu
print(s[-2:])           # prints er
```

- Write a function called total_seconds that takes one string argument. This argument will be a string of the form "M:SS" where M is a number of minutes (a single digit) and SS is a number of seconds (2 digits). This function should calculate the total number of seconds in this amount of time and return it as an integer.
- Write a main function that lets the user type in a time as a string and will call your total_seconds function.