New Strings from Old

`s.startswith(t)`	True if the string s begins with the string t.
`s.endswith(t)`	True if the string s ends with the string t.

`s.find(t)`	Returns the lowest index at which substring t is found inside s.
`s.find(t, p)`	Same as above, but starts searching at position p.
`s.replace(t, t2)`	Returns a copy of s with all occurrences of t replaced by t2.

`s.upper()`	Returns a copy of s with all letters converted to uppercase.
`s.lower()`	Returns a copy of s with all letters converted to lowercase.

Counting the number of times something happens in a string

```
def some_counting_function(string):
  total = 0
  for pos in range(0, len(string)):
    if <test string[pos] for something>:
      total = total + 1
  return total
```

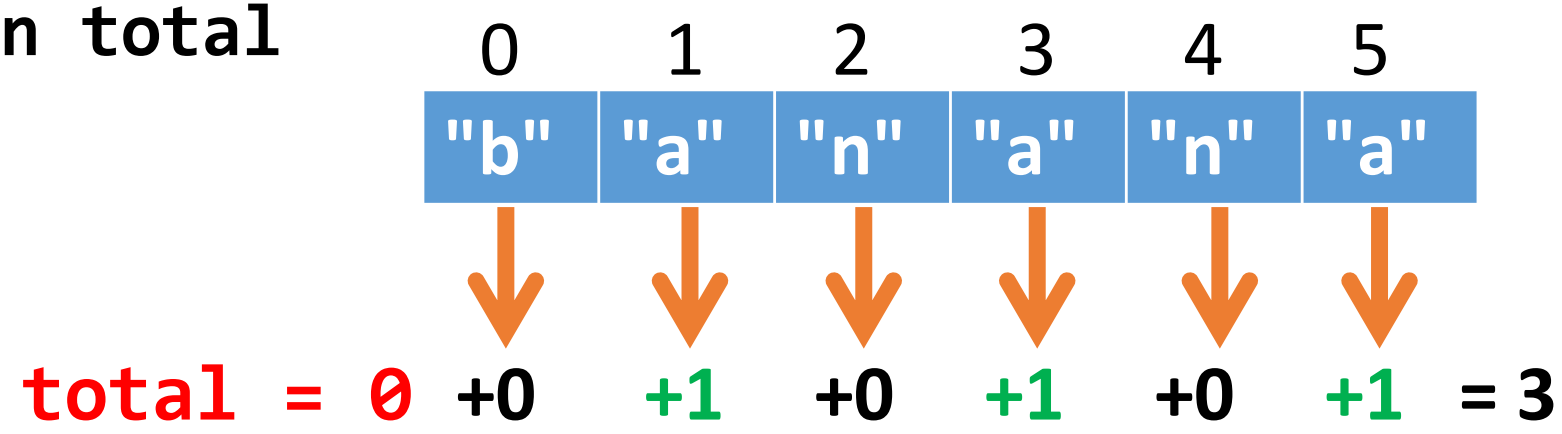
Counting the number of a's

```
def count_a(s):
    total = 0
    for pos in range(0, len(string)):
        if string[pos] == "a":
            total = total + 1
    else:           # implied
        total = total + 0
    return total
```

What do we add to `total` if the if statement is false?

Counting the number of a's

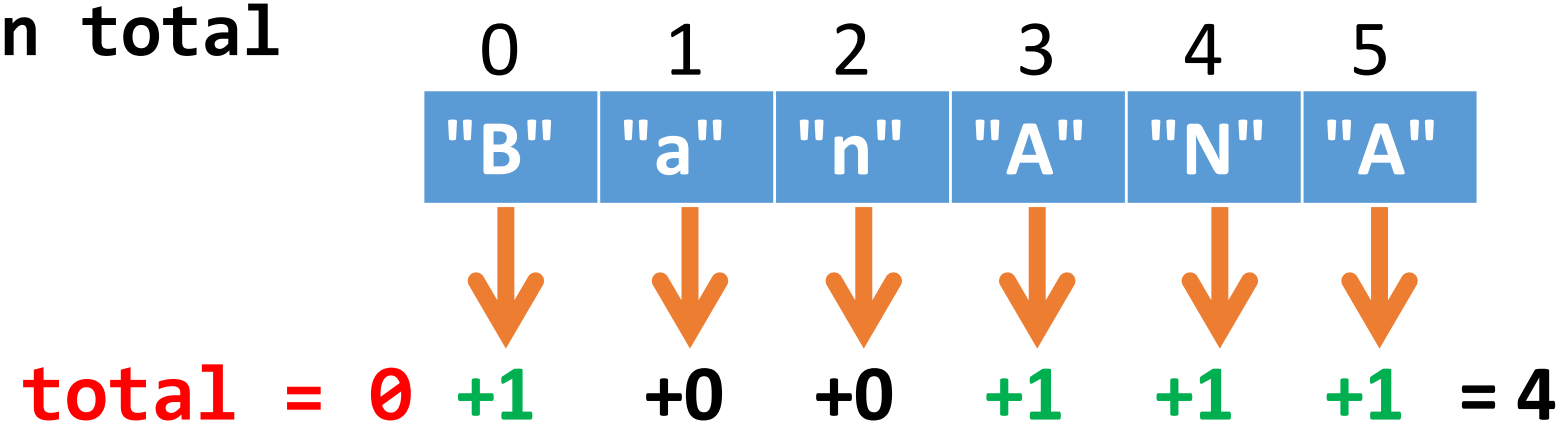
```
def count_a(string):
    total = 0
    for pos in range(0, len(string)):
        if string[pos] == "a":
            total = total + 1
    return total
```



```
def count_??(string):
  total = 0
  for pos in range(0, len(string)):
    if string[pos].isupper():
      total = total + 1
  return total
```

Counting the number of capital letters

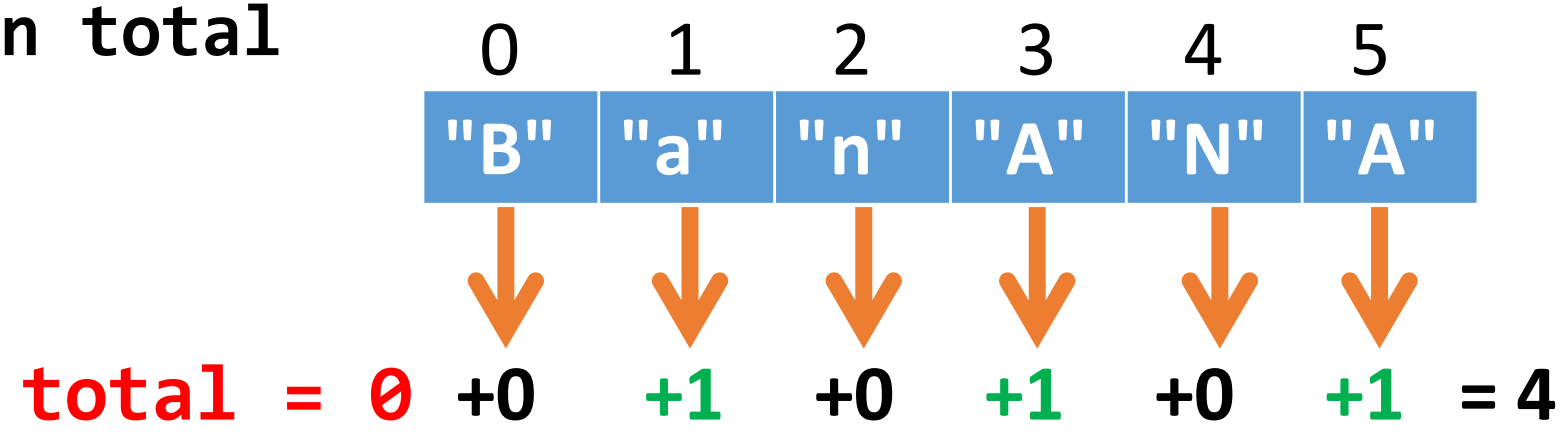
```
def count_caps(s):
    total = 0
    for pos in range(0, len(string)):
        if string[pos].isupper():
            total = total + 1
    return total
```



```
def count_??(string):
  total = 0
  for pos in range(0, len(string)):
    if string[pos] in "aeiouAEIOU":
      total = total + 1
  return total
```

Counting the number of vowels

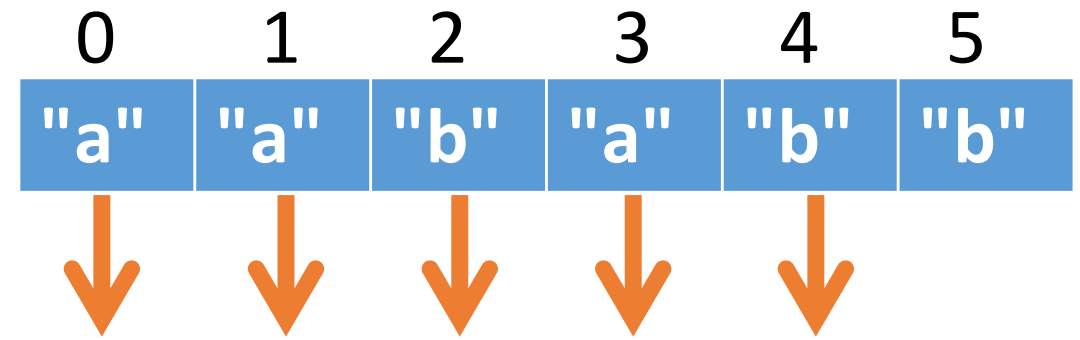
```
def count_vowels(s):
    total = 0
    for pos in range(0, len(string)):
        if string[pos] in "aeiouAEIOU":
            total = total + 1
    return total
```



```
def count_??(s):
  total = 0
  for pos in range(0, len(string)-1):
    if string[pos] == string[pos+1]:
      total = total + 1
  return total
```

Counting the number of duplicate letters

```
def count_dups(s):
    total = 0
    for pos in range(0, len(string)-1):
        if string[pos] == string[pos+1]:
            total = total + 1
    return total
```



total = 0 +1 +0 +0 +0 +1 = 2

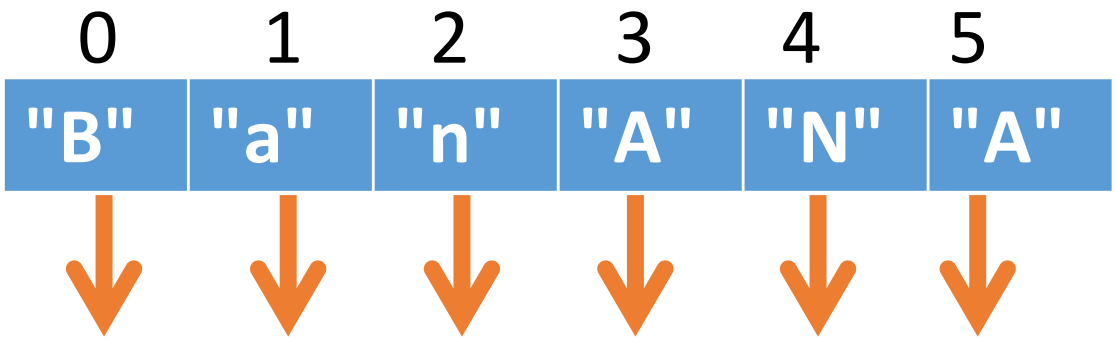
What if we turn this into a string concatenation problem?

```
def what_is_this_function(string):
  answer = ""
  for pos in range(0, len(string)):
    if string[pos].isupper():
      answer = answer + string[pos]
  return answer
```

```
def what_is_this_function(string):
    answer = ""
    for pos in range(0, len(string)):
        if string[pos].isupper():
            answer = answer + string[pos]
    else:           # implied
        answer = total + ""
    return answer
```

What do we add to `answer` if the if statement is false?

```
def filter_capitals(string):
    answer = ""
    for pos in range(0, len(string)):
        if string[pos].isupper():
            answer = answer + string[pos]
    return answer
```



answer = "" + "B" +"" +"" +"A"+"N"+"A"="BANA"

COUNT

```
def some_counting_function(string):
  total = 0
  for pos in range(0, len(string)):
    if <test string[pos] for something>:
      total = total + 1
  return total
```

FILTER

```
def some_filtering_function(string):
  answer = ""
  for pos in range(0, len(string)):
    if <test string[pos] for something>:
      answer = answer + string[pos]
  return answer
```