Strings

Strings

In Python (and most languages), text is represented in a *string*: a sequence of characters.

A string value is specified by a pair of single or double quotations.

Example.

```
s = 'trinity'
```


Strings

Each letter of a string is indexed by an integer, starting from 0.

Example.

```
0 1 2 3 4 5 6
t r i n i t y
```

Strings

To access a string `t`'s `i`th letter, use `t[i]`.

Example.

```
>>> t = 'trinity'
>>> print t[0]
t
>>> print t[3]
n
```


Strings

Strings are *immutable*. Unlike variables, letters inside a string cannot be modified.

Example.

```
>>> t = 'trinity'
>>> t[0] = 'T'
TypeError: object does not...
```


`len`

The function `len` gives the length of (or the number of letters in) a string.

Example.

```
>>> t = 'trinity'
>>> len(t)
7
```

The last letter is indexed by `len(t)-1`.

Slicing strings

Strings can be *sliced* by specifying the first and last indices: `s[i:j]` is the substring of `s` starting from `i` through `j-1`.

Example.

```
>>> t = 'trinity'
>>> print t[0:4]
trin
>>> print t[2:6]
init
```


Membership

With `in`, we can test whether or not a string is a substring of another:

Example.

```
>>> 'i' in 'trinity'
True
>>> 'k' in 'trinity'
False
```


Membership

With `in`, we can test whether or not a string is a substring of another:

Example.

```
>>> 'init' in 'trinity'
True
>>> 'int' in 'trinity'
False
```

for loop

Example.

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

This function prints all letters that appear in two given words.

for loop

Example.

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

```
>>> in_both('trinity', 'find')
i n i
```


for loop

Example.

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

```
>>> in_both('find', 'trinity')
i n
```

```
def find_first_vowel_index(word):
  "Find the index of the first vowel in the given word."
  i = 0
  while i < len(word):
    if word[i] in 'aeiou':
      return i
    i = i + 1
  return -1

def pig_latin(word):
  "Convert the given word to its Pig Latin version."
  pigword = ""
  i = find_first_vowel_index(word)
  if i == 0:           # word begins with a vowel
    pigword = word + 'yay'
  else:
    pigword = word[i:] + word[0:i] + 'ay'
  return pigword

print pig_latin('story')
print pig_latin('alpha')
print pig_latin('trinity')
print pig_latin('college')
```