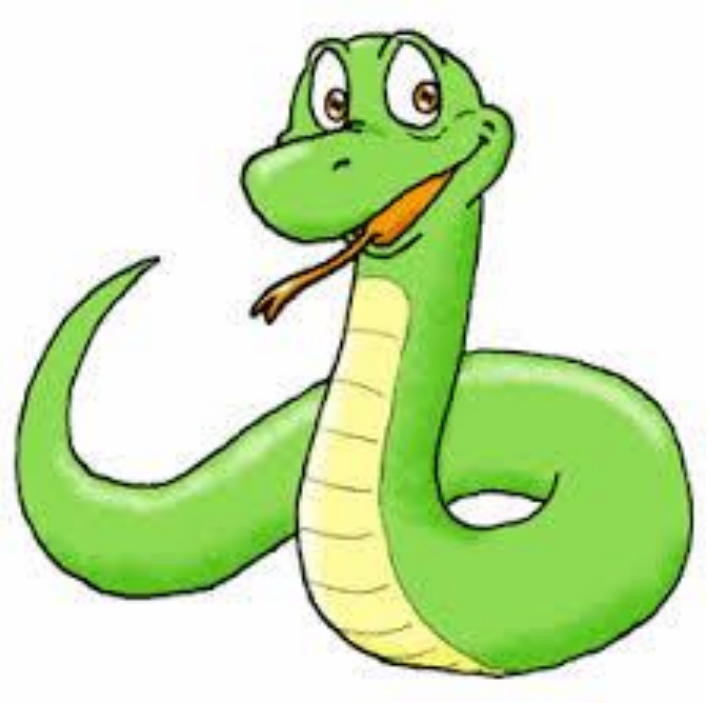
Application of GIS with Python

Chapter 4: Strings



https://www.python.org/
http://www.tutorialspoint.com/python/

Table of Content

- Concatenation, comparison
- Length of a string, string subscripts, positive and negative indices
- 'in' operator
- Strings are immutable
- Strings methods

Strings

- A string is a sequence of characters: compound data type
- Can access the characters one at a time with the bracket operator
- The expression in brackets is called an index

```
>>> fruit = 'banana'
>>> letter = fruit[1]
>>> print letter
a
```

- Strings "..." '...'
- Besides numbers, Python can also manipulate strings, which can be expressed in several ways.
- •Strings can be enclosed in single quotes or double quotes: " or '

```
>>> 'spam eggs'
‘spam eggs’
>>> 'doesn\'t'                  #\ as escape character
“doesn't”
>>> "doesn't"
“doesn't”
>>> '"Yes," he said.'
‘”Yes,” he said.’
```

String concatenation

- Strings can be concatenated (glued together) with the + operator, and repeated with *:

```
>>> word = 'Help' + 'A'
>>> word
'HelpA'
>>> '<' + word*5 + '>'
'<HelpAHelpAHelpAHelpAHelpA>'
```

Automatic concatenation

- Two string literals next to each other are automatically concatenated.
- This only works with two literals, not with arbitrary string expressions:

```
>>> 'str' 'ing' # <- This is ok
'string'
>>> 'str'.strip() + 'ing' # <- This is ok
'string'
>>> 'str'.strip() 'ing' # <- This is invalid
SyntaxError: invalid syntax
```

String comparison

- The comparison operators work on strings

```
>>>word=‘Pineapple’
>>> if word == 'banana':
        print 'All right, bananas.'
```

- Comparison for alphabetical order

```
>>> 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.'
```

- All the uppercase letters come before lower case.

```
>>> ord('A')
65
>>> chr(65)
'A'
>>>ord(‘a)
97
```

Your word, Pineapple, comes before banana.

Length of a string

- len built in function returns the number of characters in a string

```
>>>mycourse='Geomatics'
>>> print len(mycourse)
9
```

- Quiz: What is the last index of myCourse?

  len(myCourse) – 1

Remember there is a "zeroth index"!

String subscripts [...] [...:...]

- Strings can be subscripted (indexed)
- The first character of a string has subscript 0.
- A character is simply a string of size one.
- Substrings can be specified with the slice notation: two indices separated by a colon.

```
>>> word='HelpA'
>>> word[4]
'A'
>>> word[0:2]
'He'
>>> word[2:4]
'lp'
```

```
Check new string with combined content
>>> 'x' + word[1:]
'xelpA'
>>> 'Splat' + word[4]
'SplatA'
```

String Slices

- A character can be accessed by

```
>>> print myCourse[4]
```

- A slice is a segment of a string

```
>>> print myCourse[0:5]
>>> print myCourse[:5]
>>> print myCourse[3:7]
```

Operator [n:m]; from the n-th character to the m-th character (excluding m-th character)

Positive & Negative indices

- Indices may be positive numbers, to start counting from the left with the left edge of the first character numbered 0
- Indices may be negative numbers, to start counting from the right with the right edge of the last character numbered -1

```
+---+---+---+---+---+
| H | e | l | p | A |
+---+---+---+---+---+
  0   1   2   3   4
 -5  -4  -3  -2  -1
```

Negative indices

```
>>> word[-1]     # The last character; word=‘HelpA’
'A'
>>> word[-2]     # The last-but-one character
'p'
>>> word[-2:]    # The last two characters
'pA'
>>> word[:-2]    # Everything except the last two characters
'Hel'
```

Step

- There can be a third value in a slice, which indicates the step size:

  s[start : end : step]

- Default values:
  - start 0
  - end len(s)
  - step 1

Examples

```
>>> s = 'banana'
>>> s[1::2]
'aaa'
>>> s[::-1]
'ananab'
>>> s[-2:1:-1]
'nan'
```

How to generate

G
Ge
Geo
Geoi
Geoin
Geoinf
Geoinfo
Geoinfor
Geoinform
Geoinforma
Geoinformat
Geoinformati
Geoinformatic
Geoinformatics

Iterate over characters; for a case , while loop

```
>>>mycourse='Geoinformatics'
>>> character=''
>>> index=0
>>> while index < len(mycourse):
        character += mycourse[index]
        print character
        index = index + 1
```

- for loop

```
>>>char=''
>>> for chr in mycourse:
        char+=chr
        print (char)
```

‘in’ operator

- If we want to know whether a string contains a certain character but we're not interested in it's position then we can use the in operator: (Boolean expression)

```
>>> 'o' in 'hello world'
True
```

It even works for sub-strings:

```
>>> 'nana' in 'banana'
True
```

Strings are immutable

- A character in a string cannot be changed using [ ] operator on left side of assignment directly. As strings are immutable.
- Case is; object is string and the item to assign is character

```
>>> w='HelpA'
>>> w[2]='Q'
TypeError: object does not support item assignment
```

- Slices can help:

```
>>> w=w[:2]+'Q'+w[3:]
>>> w
'HeQpA'
```

String methods

- The additional functionalities for strings are called methods (like any other additional functionality of a Python object)
- A method is similar to a function—it takes arguments and returns a value but the syntax is different.

  Example, the method upper

  Instead of the function syntax **upper(word**), it uses the method syntax **word.upper( )**

String methods

```
>>> myCourse.count("i") # count occurrences
2
>>> myCourse.find("info") # returns first index
3
>>> myCourse.replace("informatics","graphy")
'Geography'
```

```
>>> 'Hello World'.upper()
'HELLO WORLD'

>>> 'Hello World'.lower()
'hello world'

>>> 'hello world'.title()
'Hello World'

>>> 'Hello World'.swapcase()
'hELLO wORLD'
```

Assignment 4:

1. What do you understand by Concatenation, comparison, Length of a string, string subscripts, positive and negative indices, 'in' operator, Strings methods in Python programming? WAP to demonstrate their use using python language.