



More About Strings

School of Information Studies
Syracuse University

Strings Defined

We can use either single or double quotes

```
>>>fruitstr = "banana"
```

Sequence of characters

```
>>>fruitstr[0]    Result – b
```

```
>>>len (fruitstr)  Result – 6
```

```
>>>fruitstr[3:5]   Result – ana
```

Iteration of Characters

We can use for loops on strings

```
>>>for char in fruitstr:
```

```
...     print (char)
```

```
...
```

```
b
```

```
a
```

```
n
```

Assign Items in a List

Assign items to a list

- Cannot assign characters to a string

```
>>>fruitlist = ['apple', 'banana', 'orange']
```

```
>>>fruitlist[0]
```

```
'apple'
```

```
>>>fruitlist[0] = 'pear'
```

```
>>>fruitlist
```

```
['pear', 'banana', 'orange']
```

```
>>>fruitstr[0] = 'd'      Would give an error
```

Using Boolean Operators

To test for an item in a list

```
>>> 'apple' in fruitlist
```

To test for a character in a string

```
>>> 'a' in fruitstr
```

```
>>> 'z' in fruitstr
```


More String Functions

To see a list of string functions

```
>>>dir (str)
```

Try out these functions

```
>>>mp = 'Monty Python'
```

```
>>>mp.lower ()          monty python
```

```
>>>mp.upper ()          MONTY PYTHON
```

```
>>>mp.strip ()           MontyPython
```

```
>>>mp.startswith ('M')  True
```

```
>>>mp.find ('P')         Python
```

Additional String Functions

```
>>> mp.isnumeric()           False
>>> mp.islower()             False
>>> mp.lower()                'monty python'
>>> mp.islower()             False
>>> mp                        'Monty Python'
>>> mp.isupper()             False
>>> mp.isprintable()         True
>>> mp.swapcase()             'mONTY pYTHON'
```





Formatting Strings

School of Information Studies
Syracuse University

FORMAT Function

Allows construction of larger strings

Format specification {}

Act as placeholders

Example using :d for integer decimal

```
>>>fruitlist = ('acai', 'banana', 'cherry')
```

```
>>>len (fruitlist)
```

```
3
```

```
>>>print ('There are {:d} fruits.'.format(len(fruitlist)))
```

```
'There are 3 fruits.'
```

Formatting for Strings

Examples using `{:s}` for strings

```
>>> adjective = 'great'
```

```
>>> 'These fruits are {:s}'.format(adjective)
```

```
'These fruits are great'
```

```
>>> for fruit in fruitlist:
```

```
>>>     print('My fruit is a {:s}'.format(fruit))
```

| Indexing the Format Argument

```
>>>'My fruits are {:s}, {:s}, and {:s}'.format(fruitlist[0],  
fruitlist[1], fruitlist[2])
```

```
'My fruits are acai, banana, and cherry'
```

Another possibility

```
>>>'My fruits are {0[0]}, {0[1]}, and  
{0[2]}'.format(fruitlist)
```

```
'My fruits are acai, banana, and cherry'
```


| Format for Floating Decimal

`{:.2f}` – format floating point number with two decimals

```
>>>num2 = 54.749
```

```
>>>'The estimate is {:.2f}.'format (num2)
```

```
'The estimate is 54.75.'
```

Statements and Comments

Statement—unit of execution

Comment—explains the code

- Use the ‘#’ to denote a comment

Program—sequence of statements

- Create a sequence of statements in notepad
- File extension .py



Formatting Strings

School of Information Studies
Syracuse University

FORMAT Function

Allows construction of larger strings

Format specification {}

Act as placeholders

Example using :d for integer decimal

```
>>>fruitlist = ('acai', 'banana', 'cherry')
```

```
>>>len (fruitlist)
```

```
3
```

```
>>>'There are {:d} fruits.'.format (len (fruitlist))
```

```
'There are 3 fruits.'
```

Formatting for Strings

Examples using `{:s}` for strings

```
>>> adjective = 'great'
```

```
>>> 'These fruits are {:s}'.format(adjective)
```

```
'These fruits are great'
```

```
>>> for fruit in fruitlist:
```

```
>>>     print('My fruit is a {:s}'.format(fruit))
```

| Indexing the Format Argument

```
>>>'My fruits are {:s}, {:s}, and {:s}'.format(fruitlist[0],  
fruitlist[1], fruitlist[2])
```

```
'My fruits are acai, banana, and cherry'
```

Another possibility

```
>>>'My fruits are {0[0]}, {0[1]}, and  
{0[2]}'.format(fruitlist)
```

```
'My fruits are acai, banana, and cherry'
```

| Format for Floating Decimal

`{:.2f}` – format floating point number with two decimals

```
>>>num2 = 54.749
```

```
>>>'The estimate is {:.2f}.'format(num2)
```

```
'The estimate is 54.75.'
```

Statements and Comments

Statement—unit of execution

Comment—explains the code

- Use the ‘#’ to denote a comment

Program—sequence of statements

- Create a sequence of statements in notepad
- File extension .py