NPTEL MOOC

PROGRAMMING, DATA STRUCTURES AND ALGORITHMS IN PYTHON

Week 5, Lecture 4

Madhavan Mukund, Chennai Mathematical Institute
http://www.cmi.ac.in/~madhavan

String processing

- Easy to read and write text files
- String processing functions make it easy to analyse and transform contents
  - Search and replace text
  - Export spreadsheet as text file (csv) and process columns
  - ...

Strip whitespace

- `s.rstrip()` removes trailing whitespace

```
for line in contents:
   s = line.rstrip()
```

- `s.lstrip()` removes leading whitespace
- `s.strip()` removes leading and trailing whitespace

Searching for text

```
s.find(pattern)
```

- Returns first position in `s` where `pattern` occurs, `-1` if no occurrence of `pattern`

```
s.find(pattern,start,end)
```

- Search for `pattern` in slice `s[start:end]`

```
s.index(pattern), s.index(pattern,l,r)
```

- Like `find`, but raise `ValueError` if `pattern` not found

Search and replace

```
s.replace(fromstr,tostr)
```

- Returns copy of s with each occurrence of fromstr replaced by tostr

```
s.replace(fromstr,tostr,n)
```

- Replace at most first n copies

- Note that s itself is unchanged — strings are immutable

Splitting a string

- Export spreadsheet as “comma separated value” text file

- Want to extract columns from a line of text

- Split the line into chunks between commas

```
columns = s.split(",")
```

  - Can split using any separator string

- Split into at most `n` chunks

```
columns = s.split(" : ", n)
```

Joining strings

- Recombine a list of strings using a separator

```
columns = s.split(",")
joinstring = ","
csvline = joinstring.join(columns)

date = "16"
month = "08"
year = "2016"
today = "-".join([date,month,year])
```

Converting case

- Convert lower case to upper case, …
- `s.capitalize()` — return new string with first letter uppercase, rest lower
- `s.lower()` — convert all uppercase to lowercase
- `s.upper()` — convert all lowercase to uppercase
- `s.title(), s.swapcase(), …`

Resizing strings

```
s.center(n)
```

- Returns string of length n with s centred, rest blank

```
s.center(n,"*")
```

- Fill the rest with * instead of blanks

```
s.ljust(n), s.ljust(n,"*"), s.rjust(n), …
```

- Similar, but left/right justify s in returned string

Other functions

- Check the nature of characters in a string

```
s.isalpha(), s.isnumeric(), …
```

- Many other functions
- Check the Python documentation