Strings: the Basics

L435/L555
Dept. of Linguistics, Indiana University
Fall 2016

Strings: What we already know

- Strings are sequences: order is important
  - indexing, slicing
  - looping over characters in a string
  - concatenation, len(), etc.
- Strings are immutable: they do not change
  - no use of append, etc.
  - cannot change values via index re-assignment, etc.
- Strings can occur in boolean statements
  - in test
  - alphabetic checks (<, >, etc.)

String formatting

Basic placeholders

- Strings can have placeholders, the values placed in \texttt{.format()}: 
  \begin{verbatim}
  s="you're eating \{0\}".format("crazy\_cheese")
  \texttt{# s = "you're eating crazy cheese"}
  \end{verbatim}
- We can do that with variables, too:
  \begin{verbatim}
  location='paris'
  s="you\_would\_think \{0\}".format(location)
  \texttt{# s = "you would think I'm from paris"}
  \end{verbatim}
- And with more than one value:
  \begin{verbatim}
  lyric = "you\_know\_i\_get\_\{0\}\_get\_\{1\}"
  adjs = ("fly", "high")
  print(lyric.format(*adjs))
  \end{verbatim}

See: https://docs.python.org/3.4/library/string.html, sec. 6.1.3.2

String formatting

Alignment & width

Using a colon (:), we can do left (<), right (>), & center () alignment

\begin{verbatim}
s=[\{:<10\}\_know\_that\_i\_m\_gone\}.format("you")
\texttt{# s = "you know that I'm gone"}
\end{verbatim}

\begin{verbatim}
s=\{"les-b:)\_tell\_\{>10\}\_all\_why\}.format("you")
\texttt{# s = "I'm a tell you all why"}
\end{verbatim}

\begin{verbatim}
s=\{"who\_are\_you\_\{:20\}\}.format("dissing")
\texttt{# s = "who are you dissing"}
\end{verbatim}

\begin{verbatim}
s=\{"maybe\_i\_m\_\{:20\}\}.format("missing")
\texttt{# s = "maybe I'm missing"}
\end{verbatim}

In the last case, we use * as a fill character

Conversion types

\begin{verbatim}
s=\{string\_converted\_with\_str\}
r=\{string\_converted\_with\_repr\}
c=\{single\_character\}
d,n=\{decimal\_integer\}
f,F=\{floating\_point\_decimal\}
\end{verbatim}
Strings: the Basics

String basics

Review

Formatting

Conversion types

String methods

find

join & split

lower & upper

replace

strip

Example

Find where a string starts (cf. index() for lists)

```python
phrase="the reason that you're smilin'
phrase.find('son') # 7
phrase.find('smile') # -1
if phrase.find('you')>=0:
    print("me!")
```

- not: find does NOT return a Boolean value: if it does not find the substring, it returns -1

Changing case

1. Make a string all lowercase
   'SMILIN'.lower()
2. Make a string all uppercase
   'wildin'.upper()
3. Make all but the first letter of a string lowercase
   'LISTEN'.title()

replace

1. Replace smilin with frown in the phrase
   phrase.replace('smilin', 'frownin')
2. Replace e with o in the phrase
   phrase=phrase.replace('e', 'o')

strip

1. Strip off newline characters from end of the phrase
   phrase=phrase.strip('\\n')
2. Strip off any leading or trailing whitespace from the phrase, and convert to upper case
   phrase=phrase.strip().upper()
3. Strip off any leading or trailing whitespace from the haystack phrase, replace smilin with frownin and convert to upper case
   phrase=phrase.strip().replace('smilin', 'frownin').upper()