Strings

- Slicing
  \texttt{string[start:end:stride]}: Returns string starting at index \texttt{start}, up to but not including the character at index \texttt{end}, and jumping by \texttt{stride}. \texttt{start} defaults to 0, \texttt{end} defaults to length of string, \texttt{stride} defaults to 1.

- The following functions are built-in
  \texttt{len(string)}: Returns the number of characters in the string
  \texttt{int(string), float(string)}: Converts numeric string to int or float
  \texttt{str(object)}: Converts object, e.g., int or float to a string
  \texttt{sorted(string)}: Returns the characters of the string as a list in sorted order

- String object methods (invoked with the object\texttt{.method} syntax)
  \texttt{upper(), lower(), capitalize()}: Returns a new upper or lower-cased, or 1st letter upper-cased string
  \texttt{find(substring)}: Returns the first index that \texttt{substring} occurs at in the string or -1 if not found
  \texttt{find(substring, index)}: Same as above, but starts searching at index
  \texttt{replace(old, new)}: Returns a copy of the string with all occurrences of \texttt{old} substituted with \texttt{new}
  \texttt{startswith(prefix)}: Returns \texttt{True} if the string starts with \texttt{prefix}, \texttt{False} otherwise
  \texttt{endswith(suffix)}: Returns \texttt{True} if the string ends with \texttt{suffix}, \texttt{False} otherwise
  \texttt{strip()}: Returns a copy of the string with only the leading and trailing whitespace removed
  \texttt{split()}: Return a list of the words in the string using whitespace as the delimiter

- String operators
  \texttt{string1 + string2}: Returns a new string that is the concatenation of \texttt{string1} and \texttt{string2}
  \texttt{string * int}: Returns a new string that is \texttt{string} repeated \texttt{int} times
  \texttt{substr in string}: Returns \texttt{True} if \texttt{substr} is a substring of \texttt{string}, \texttt{False} otherwise

\textbf{for-loops on strings}

To loop over each character in string \texttt{s}:

\begin{verbatim}
for <element> in <string>:
  <statements>
\end{verbatim}

For example, if the variable \texttt{word} contains ‘abc’:

\begin{verbatim}
for letter in word:
  print(letter)
\end{verbatim}

produces output
\begin{verbatim}
a
b
c
\end{verbatim}