

CSE 160 Midterm Cheat Sheet

```
# if/elif/else syntax
if condition1:
    # statements
elif condition2:
    # other statements
else:
    # more statements
```

```
# for Loop syntax
for i in sequence:
    # statements

# function definition syntax
def function_name(param1, param2, ...):
    # statements
```

Function	Description
<code>range([start,] stop [, step])</code>	Returns a sequence of numbers from start (inclusive) to stop (exclusive) incremented by step
<code>len(Lst)</code>	Returns the number of elements in Lst

Lists

Function	Description
<code>lst = []</code>	Creates an empty list
<code>lst[idx]</code>	Returns the element in Lst at index idx
<code>lst[start : end]</code>	Creates a sublist of Lst from index start to index end (exclusive)
<code>lst.append(elmt)</code>	Adds the element elmt to the end of Lst
<code>lst.index(elmt)</code>	Returns the index of the first occurrence of elmt in Lst
<code>lst.count(elmt)</code>	Returns the number of times elmt occurs in Lst
<code>lst.remove(elmt)</code>	Removes first occurrence of elmt from Lst
<code>lst.pop(idx)</code> <code>lst.pop()</code>	Removes and returns the element at index idx in Lst With no parameter, removes the last element in Lst
<code>lst.insert(idx, elmt)</code>	Inserts an element elmt in list at index idx

File I/O

Function	Description
<code>open(filepath)</code>	Opens the file with given filepath for reading
<code>my_file.close()</code>	Closes file my_file

Sets

Function	Description
<code>{<i>elmt</i>(s)}, set(<i>Lst</i>)</code>	Constructs a set of provided <i>elmt</i> (s), or of elements in <i>Lst</i>
<code><i>my_set</i>.add(<i>elmt</i>)</code>	Adds <i>elmt</i> to <i>my_set</i>
<code><i>my_set</i>.remove(<i>elmt</i>)</code>	Removes an element from <i>my_set</i> if present, otherwise error
<code><i>my_set</i>.discard(<i>elmt</i>)</code>	Removes an element from <i>my_set</i> (no errors thrown)
<code><i>my_set</i>.pop()</code>	Removes a random element from <i>my_set</i>

Set Operation	Description
<code>&</code>	Intersection, or logical AND
<code> </code>	Union, or logical OR
<code>^</code>	XOR
<code>-</code>	Difference

Dictionaries

Function	Description
<code><i>dict</i> = {}</code>	Creates a new, empty dictionary
<code><i>dict</i>[<i>key</i>]</code>	Returns the value associated with the given key in <i>dict</i>
<code><i>dict</i>.keys()</code>	Returns list of keys in <i>dict</i>
<code><i>dict</i>.values()</code>	Returns list of values in <i>dict</i>

Sorting

Function	Description
<code>sorted(<i>collection</i> [,key=<i>sort_key</i>, reverse=<i>bool_val</i>])</code>	Returns a sorted copy of <i>collection</i> , based on optional sort key (<i>key</i>) and optional order preference (<i>reverse</i>)
<code><i>collection</i>.sort([key=<i>sort_key</i>, reverse=<i>bool_val</i>])</code>	Sorts the given collection, based on optional sort key (<i>key</i>) and optional order preference (<i>reverse</i>), and returns None