

CSE 160 Final Exam 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 index of the first occurrence of eLmt in Lst , Error if eLmt is not 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 , Error if eLmt is not in 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>my_file = open(filepath)</code>	Opens the file with given filepath for reading, returns a file object
<code>my_file.close()</code>	Closes file my_file

```
# Process one line at a time:
for line_of_text in my_file:
    # process line_of_text
```

```
# Process entire file at once
all_data_as_a_big_string = my_file.read()
```

Sets

Function	Description
<code>{elmt(s)}, set(lst)</code>	Constructs a set of provided <i>elmt</i> (s), or of elements in <i>lst</i>
<code>my_set.add(elmt)</code>	Adds <i>elmt</i> to <i>my_set</i>
<code>my_set.remove(elmt)</code>	Removes an element from <i>my_set</i> if present, otherwise error
<code>my_set.discard(elmt)</code>	Removes an element from <i>my_set</i> (no errors thrown)
<code>my_set.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>my_dict = {}</code>	Creates a new, empty dictionary
<code>my_dict[key]</code>	Returns the value associated with the given key in <i>my_dict</i>
<code>my_dict.keys()</code>	Returns list of keys in <i>my_dict</i>
<code>my_dict.values()</code>	Returns list of values in <i>my_dict</i>

Sorting

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