PYTHON PROGRAMMING

Data Structures & Strings — Quick Reference Cheat Sheet

Dr. Amar Panchal | Career Credentials

DATA STRUCTURES IN PYTHON

LIST

Method	Description
`list.append(x)`	Add an item to the end of the list.
`list.extend(L)`	Extend list by appending all items in the given list.
`list.insert(i, x)`	Insert an item at a given position.
`list.remove(x)`	Remove the first item from the list whose value is x. Error if not found.
`list.pop([i])`	Remove the item at the given position and return it.
`list.clear()`	Remove all items from the list. Equivalent to del a[:].
`list.index(x)`	Return the index of the first item whose value is x. Error if not found.
`list.count(x)`	Return the number of times x appears in the list.
`list.sort()`	Sort the items of the list in place.
`list.reverse()`	Reverse the list in place.

TUPLE

Method	Description
`count(data)`	Count occurrences of data in the tuple.
`index(data)`	Return the index of the first occurrence of data.

SET

Method	Description
`add(data)`	Add one item to the set.
`update(set)`	Add all elements of the given set to the set.
`remove(data)`	Remove data from the set. Raises error if not found.
`discard(data)`	Remove data from the set without raising an error if not found.

DICTIONARY (dict)

Key Facts: Ordered (Python 3.7+), unordered in 3.6 and earlier. Duplicate keys are NOT allowed.

Method / Syntax	Description
`var.keys()`	Returns a list of all keys in the dictionary.
`var.values()`	Returns a list of all values in the dictionary.
`var.items()`	Returns each key-value pair as tuples in a list.
`var.get(key)`	Returns the value for the specified key.
`var.update({key: val})`	Updates the dictionary with the given key-value pair.
`var["key"] = value`	Add or update a key-value pair.
`var.pop(key)`	Removes the item with specified key and returns its value.
`var.popitem()`	Removes the last inserted key-value pair.
`var.clear()`	Removes all elements from the dictionary.
`var.copy()`	Returns a shallow copy of the dictionary.
`var.fromkeys()`	Returns a dictionary with specified keys and value.
`var.setdefault(key, val)`	Returns value of key; inserts key with value if key does not exist.
`del var[key]`	Deletes the specified key-value pair.
`if "key" in var:`	Check if a key exists in the dictionary.

Looping Over Dictionaries

Code	Output / Purpose
`for x in thisdict: print(x)`	Print all key names
`for x in thisdict: print(thisdict[x])`	Print all values by key
`for x in thisdict.values(): print(x)`	Print all values directly
`for x, y in thisdict.items(): print(x, y)`	Loop through both keys and values

STRINGS IN PYTHON

STRING OPERATORS

Operator	Description
`+`	Concatenation — joins two strings together.
`*`	Repetition — repeats the string a given number of times.
`[]`	Slice — fetches a character at the given index.
`[ : ]`	Range Slice — fetches characters from the given range.
`in`	Membership — returns True if a character exists in the string.
`not in`	Membership — returns True if a character does NOT exist in the string.

CASE CONVERSION METHODS

Method	Description
`capitalize()`	Capitalizes the first letter of the string.
`title()`	Converts string to title case — each word starts with uppercase.
`upper()`	Converts all characters to uppercase.
`lower()`	Converts all characters to lowercase.
`swapcase()`	Inverts the case of every character in the string.

SEARCH & COUNT METHODS

Method	Description
`count(str, beg, end)`	Counts how many times str occurs in the string (or substring if beg/end given).
`find(str, beg, end)`	Returns the index of the first occurrence of str, or -1 if not found.
`index(str, beg, end)`	Same as find() but raises an exception if str is not found.
`rfind(str, beg, end)`	Same as find() but searches backwards in the string.

Method	Description
`rindex(str, beg, end)`	Same as index() but searches backwards in the string.
`len(string)`	Returns the total length of the string.
`max(str)`	Returns the maximum alphabetical character from the string.
`min(str)`	Returns the minimum alphabetical character from the string.

ALIGNMENT & PADDING METHODS

Method	Description
`center(width, fillchar)`	Centers the string within width columns, padded with fillchar.
`ljust(width, fillchar])`	Left-justifies the string, padded with spaces (or fillchar) to total width columns.
`rjust(width, fillchar])`	Right-justifies the string, padded with spaces (or fillchar) to total width columns.

STRIP & WHITESPACE METHODS

Method	Description
`lstrip()`	Removes all leading whitespace from the string.
`rstrip()`	Removes all trailing whitespace from the string.
`strip([chars])`	Removes both leading and trailing whitespace (or specified chars).

REPLACE & SPLIT METHODS

Method	Description
`replace(old, new [, max])`	Replaces all occurrences of old with new (or at most max times if max given).
`split(str, num)`	Splits the string by delimiter str (space if not provided). Returns a list.
`splitlines(num)`	Splits the string at newlines and returns a list of lines.

BOOLEAN / VALIDATION METHODS

Method	Description
`isalnum()`	True if string has at least 1 character and all characters are alphanumeric.
`isalpha()`	True if string has at least 1 character and all characters are alphabetic.
`isdigit()`	True if string contains only digits.
`islower()`	True if all cased characters are lowercase.
`isupper()`	True if all cased characters are uppercase.
`isnumeric()`	True if a unicode string contains only numeric characters.
`isspace()`	True if string contains only whitespace characters.
`istitle()`	True if the string is properly title-cased.