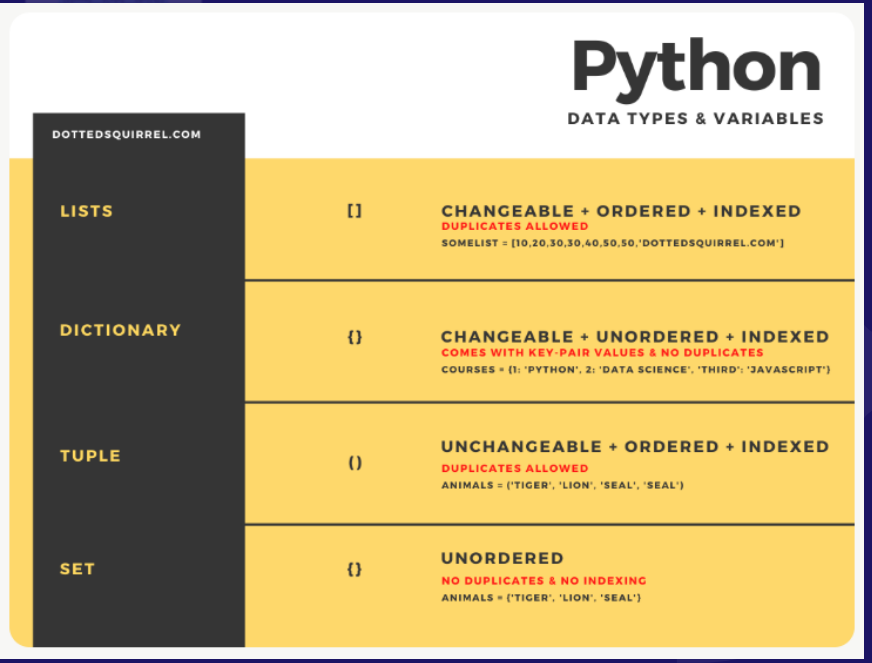
PROGRAMMING PYTHON

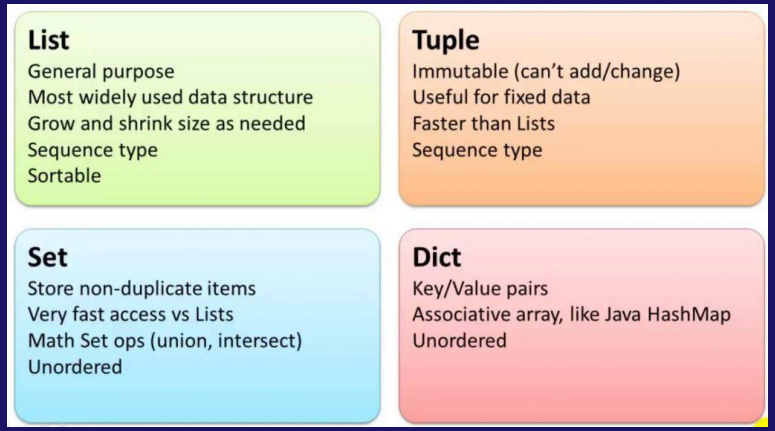
Lecture # 6 - Collectables

Code Camp | Alpha

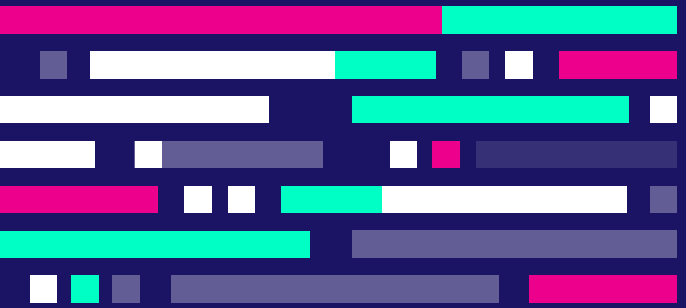


Collectables

- Collectables are used to store multiple values.
- There are collection types in Python:
  - List
  - Tuples
  - Set
  - Dictionary
- Each collectable serve a different purpose.



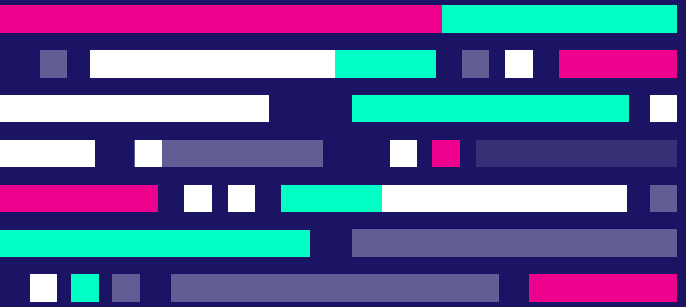
Code Camp | Alpha



Lists

- Lists are created using square brackets []
- List items are ordered, changeable, and duplicates are allowed
- **Ordered** = Values are indexed and can be accessed via indexes.
- **Changeable** = We can update values, there are methods available for this in lists
- **Duplicates** = Since each value is stores at a particular index, duplicates are ok
- Lists can store any data type.
- We can create list in two ways:
  - `mylist = ["cat", "dog", "sheep"]`
  - `mylist1 = list(("apple", "dog", "sheep"))`
- The second method is called list constructor method.
- The length of the list (like how many members it contains) can be checked using `len()` function.

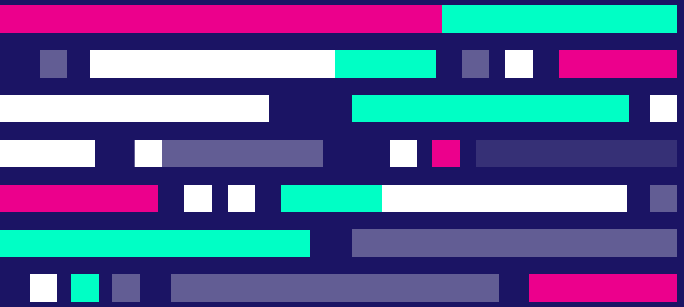
Code Camp | Alpha



Lists Items – Access

- We can access list items via indexes
  - Pet = list[1]
- Indexing follow all the rules of indexing like
  - Range
  - Negative indexing
  - Negative range
  - thislist = ["apple", "banana", "cherry", "orange", "kiwi", "melon", "mango"]
  - print(thislist[2:5])
- We can also check if an item is in the list using membership function
  - If apple in thislist:
    print("Exists")

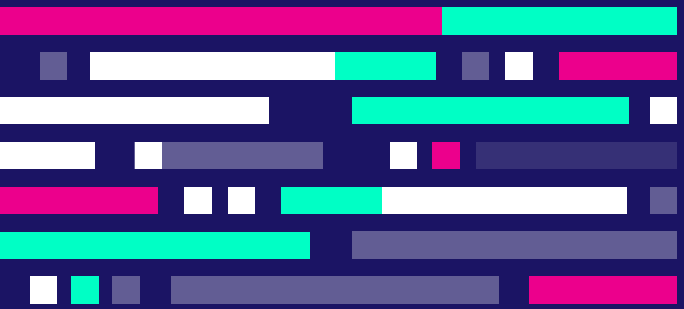
Code Camp | Alpha



Lists Items – Update

- We can change the value of a particular item using indexing
  - thislist = ["apple", "banana", "cherry"]
  - thislist[1] = "blackcurrant"
- Also we can a range of items if we select a range of indexes
  - thislist = ["apple", "banana", "cherry", "orange", "kiwi", "mango"]
  - thislist[1:3] = ["blackcurrant", "watermelon"]
  - Note: If you insert more items than you replace, the new items will be inserted where you specified, and the remaining items will move accordingly (This is an alternate to insertion)
  - Also, If you insert less items than you replace, the new items will be inserted where you specified, and the remaining items will move accordingly. (This is an alternate to deletion).

Code Camp | Alpha

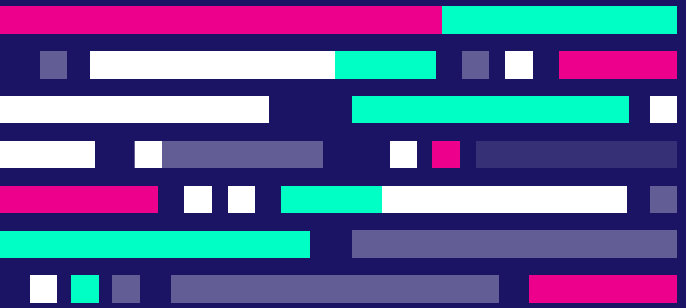


Lists Items - Add

- We can insert an item at a specified location by using insert()
  - thislist = ["apple", "banana", "cherry"]
  - thislist.insert(2, "watermelon")
  - print(thislist)

- We Use append() method to add an item to the list in the end.
  - thislist = ["apple", "banana", "cherry"]
  - thislist.append("orange")
  - print(thislist)
- We use extend() method to combine two or more lists. The appended items are added to the main list.
  - thislist = ["apple", "banana", "cherry"]
  - tropical = ["mango", "pineapple", "papaya"]
  - thislist.extend(tropical)
  - print(thislist)
- Note: Extend() methods can be used to combine any iterables.

Code Camp | Alpha



Lists Items - Remove

- We can use remove method to remove and specific item
  - thislist = ["apple", "banana", "cherry"]
  - thislist.remove("banana")
  - If there are more than one occurrences then only the first occurrence is deleted.
- We can use pop() to delete a specific index.
  - thislist = ["apple", "banana", "cherry"]
  - thislist.pop(1)
  - If index is not specified, pop() method removes the last index.
- The delete method del() also deletes a particular location or can also be used to delete an entire list.
- The clear() method is used to empty a list.
  - thislist = ["apple", "banana", "cherry"]
  - thislist.clear()

- NOTE: How methods are utilized. We use a name.method syntax.

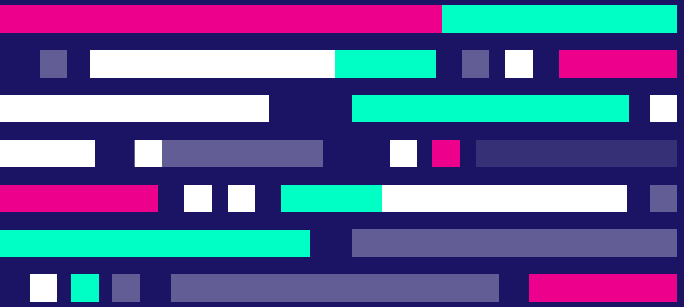
Code Camp | Alpha

Looping through lists

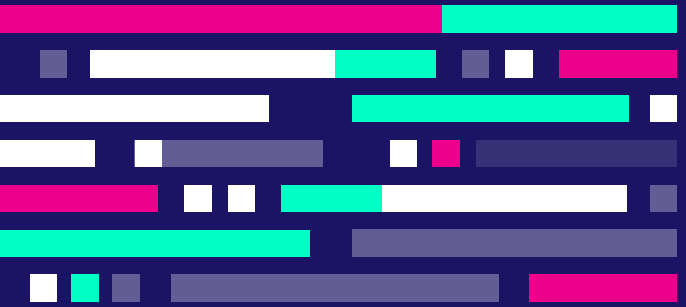
- We can use for and while to loop through entire lists.
- FOR LOOPS
  - thislist = ["apple", "banana", "cherry"]
  - for x in thislist:
  - print(x)
- We can also assign range in for loops if we want to iterate over a particular range.
  - thislist = ["apple", "banana", "cherry"]
  - for i in range(len(thislist)):
  - print(thislist[i])
- WHILE LOOP
  - thislist = ["apple", "banana", "cherry"]
  - i = 0
  - while i < len(thislist):
  - print(thislist[i])
  - i = i + 1

Code Camp | Alpha

List comprehensions – shortcuts



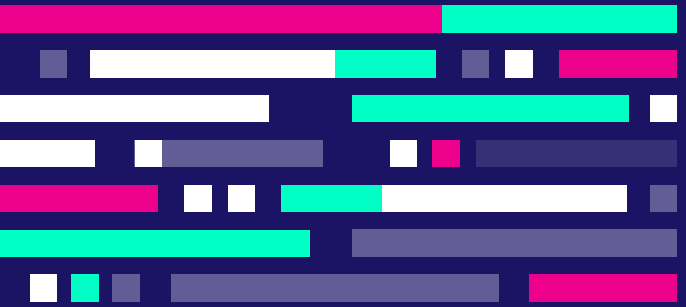
- Suppose we have a list of months as
  - Months = ["Jan","Feb","Mar","Apr","May","Jun","Jul","Aug","Sep","Oct","Nov",Dec"]
- Now if we want to separate a list of months starting with letter "J", conventionally we can use ifs in list but with list comprehensions can do in just a single line
  - Months_J = [x for x in Months if "J" in x]
  - print(Months_J)
- Syntax:
  - newlist = [expression for item in iterable if condition == True]
  - Here condition is like a filter that only accepts the items that valuate to True.
  - Conditions can be omitted.
  - Iterable is any object which can be iterated (like range, list etc.)
  - Expression is current item.
- Highly recommended: Check this list for more on this and practice: https://www.w3schools.com/python/python_lists_comprehension.asp



Lists items - Sorting

- We can use sort() to list items alphanumerically.
  - thislist = ["orange", "mango", "kiwi", "pineapple", "banana"]
  - thislist.sort()
  - NOTE: By default sorting is ascending but we can also reverse the order
  - thislist.sort(reverse = True)
  - Sort is case sensitive, it use sorts capital letters first.
  - But this method can be used to sort by using small letters
  - thislist.sort(key = str.lower)
- The reverse() method reverses the current sorting order of the elements.
  - thislist = ["banana", "Orange", "Kiwi", "cherry"]
  - thislist.reverse()

Code Camp | Alpha



Lists items - Copying / Joining

- We can also copy the contents of a list into another list
  - thislist = ["apple", "banana", "cherry"]
  - mylist = thislist.copy()
- Or with this method:
  - thislist = ["apple", "banana", "cherry"]
  - mylist = list(thislist)
- We can also join lists
  - With Concatenation
  - list1 = ["a", "b", "c"]
  - list2 = [1, 2, 3]
  - list3 = list1 + list2
  - With extend()
  - list1 = ["a", "b" , "c"]
  - list2 = [1, 2, 3]
  - list1.extend(list2)
  - print(list1)

Check

https://www.w3schools.com/python/python_lists_methods.asp