CHAPTER 10

Dictionaries

HASH

- Called dictionary in python
- Preserves the association between items

CREATING A DICTIONARY

- Strings use quotation marks
- Lists use square brackets (aka brackets)
- Dictionaries use curly brackets (aka braces)

To create an empty dictionary:

```
first_dictionary = { }
```

To create a dictionary with entries:

value

```
second_dictionary = {‘Mickey’ : 98.7, ‘Daisy’ : 86.2, ‘Donald’ : 70.0}
```

key

ADDING ELEMENTS

- Give the name of the dictionary, the key, and the value:

- Example:

```
second_dictionary[‘Scrooge] = 99.9
```

key value

ANOTHER EXAMPLE

- Two ways to create the same dictionary (with one entry):

Example 1 – create an empty dictionary and add an entry

example = { }

example[‘CSE 1284’] = 200

Example 2 – create the dictionary in one step

example = { ‘CSE 1284’ : 200 }

RETRIEVING A VALUE

- Use the dictionary name and key to retrieve the value

Using the example dictionary from the last slide:

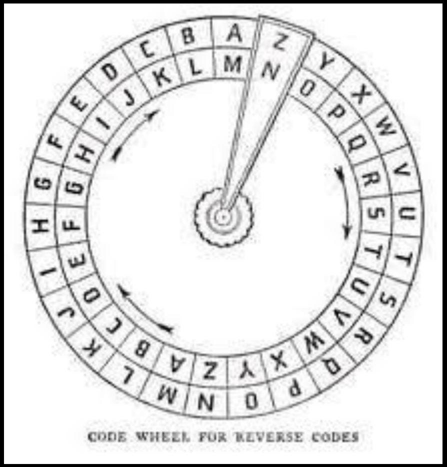
```
number_of_students = example[‘CSE 1284’]
print(number_of_students)
```

prints: 200

SUBSTITUTION CIPHER

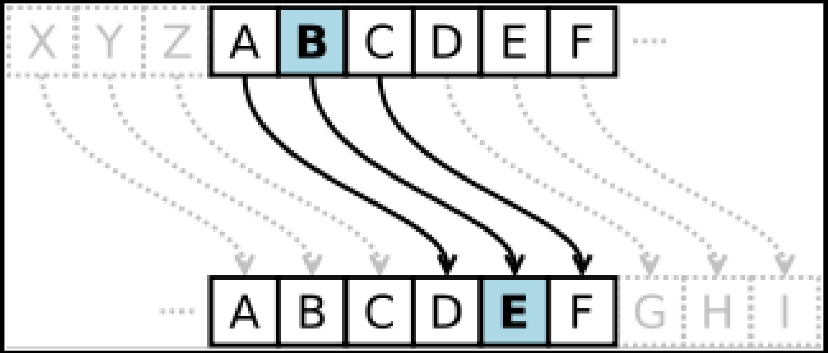
Before encryption
went binary

Encryption machine



CLASS EXERCISE

- Build the dictionary that can be used to encrypt a message from a file using a Caesar cipher.



CLASS EXERCISE

- Get a message from the user and encrypt it using your dictionary.
- Print the encrypted message.

HOW DO YOU UNENCRYPT?

OTHER NEAT THINGS

- Remove an entry from a dictionary: del dictionary_name[key]
- List the keys: dictionary_name.keys()
- Find number of items: len(dictionary_name)
- Clears contents: dictionary_name.clear()
- Returns all the keys in a dictionary and their associated values as a sequence of tuples: dictionary_name.items()
- Returns the value associated with a specified key and removes the dictionary entry: dictionary_name.pop(key)
- Returns all the values in the dictionary:
  dictionary_name.values()