

---

# Class activities Week 2 Part 1.

#1. Covert a string to a list of dictionaries (20 min)

# Activity #1 part a (5 min)

- Assume that there is a string:

```
user_hobbies="" "user1*Jack*Swimming  
user2*John*Watching movies  
user3*Jane*Painting  
user4*Jake*Having nice dinner  
user5*John*Watching movies ""
```

- Write a function that takes `user_hobbies` as a parameter, breaks it into 5 pieces (each starting with 'user') and prints each piece, one per line of output.

# Sample Code – part a

```
user_hobbies="""user1*Jack*Swimming
user2*John*Watching movies
user3*Jane*Painting
user4*Jake*Having nice dinner
user5*John*Watching movies"""

def print_user_hobbies_list(user_hobbies):
    # break up the multi-line string to a list, one line per list element
    user_hobbies_lines = user_hobbies.split('\n')
    # for each item in the list, print it, with a line number
    i = 1
    for line in user_hobbies_lines:
        # print each line
        print "%d:"%(i),line
        i = i+1

print_user_hobbies_list(user_hobbies)
```

---

# Explanation

---

# Activity #1 – part b (10 min)

- Modify your function from Activity #1 part a.
  - Instead of printing each line, convert each line to a dictionary and print the dictionary.
  - Each line has an 'id', 'name', and 'hobby' separated by the '\*' character
  - Convert each line of `val1*val2*val3` to a dictionary like:
    - `{'id': 'val1', 'name': 'val2', 'hobby': 'val3'}`

# Sample Code – part b

```
user_hobbies="""user1*Jack*Swimming
user2*John*Watching movies
user3*Jane*Painting
user4*Jake*Having nice dinner
user5*John*Watching movies"""

def user_hobbies_dict(user_hobbies):
    # break up the multi-line string to a list, one line per list element
    user_hobbies_lines = user_hobbies.split('\n')
    # for each item in the list, print it, with a line number
    i = 1
    for line in user_hobbies_lines:
        # break each line into a list of individual field values
        fields = line.split('*')
        # allocate a new, empty dictionary
        hobby_dict = dict()
        # now assign each field value to a key in the dictionary
        hobby_dict['id'] = fields[0]
        hobby_dict['name'] = fields[1]
        hobby_dict['hobby'] = fields[2]
        # now print each dictionary
        print "%d:"%(i), hobby_dict
        i = i+1

user_hobbies_dict(user_hobbies)
```

---

# Explanation

---

---

Stop

---



# Class activities Week 2 Part 2.

- #2. Read a record from a file and search records (15 min)
- #3. Append records and save a result as a file (10 min)
- #4. Read a file, edit records, and save a result as a file (10 min)

## Activity #2 (15 min)

- Read a file named *user\_hobbies.txt* that formatted as follow:

```
user1*Jack*Swimming
user2*John*Watching movie
...
```

- Then generate a list named *user\_hobbies* that the components are formatted as a dictionary. The format of each dictionary should be the same with activity #1.

```
i.e., user_hobbies= [
    {'id':'user1', 'name':'Jack', 'hobby':'Swimming'},
    {'id':'user2', 'name':'John', 'hobby':'Watching movies'},
    ...
]
```

- Finally, print a list of user ids whose hobby is "Watching movies".

# Sample Code

```
# use the readFile() procedure to get the file as a list of lines
# two small modifications to create_user_hobbies_list
# (1) a parameter to get the list of lines
# (2) remove the lines that converted the str_user_hobbies to list
#
def create_user_hobbies_list(user_hobbies_lines=None):
    # allocate a new, empty list
    user_hobbies_list = list()
    # for each element/item/line in the list
    for line in user_hobbies_lines:
        # break each line into a list of fields
        line_field = line.split('*')
        # allocate a new, empty dictionary
        hobby_dict = dict()
        # now assign each field to the key in the dictionary
        hobby_dict['id'] = line_field[0]
        hobby_dict['name'] = line_field[1]
        hobby_dict['hobby'] = line_field[2]
        # add the dictionary to a growing list of dictionaries
        user_hobbies_list.append(hobby_dict)
    # return the result
    return user_hobbies_list
```

# Sample Code

```
# Now, just need a way to search
#
def search_hobbies(user_hobbies_list=None,find_hobby=None):
    # create a new, empty list to hold what we find
    result_list = list()
    # for each of the items/records in the user_hobbies_list
    for record in user_hobbies_list:
        # if the 'hobby' field in the record is equal
        # to the item we are trying to find
        if( record['hobby']==find_hobby ):
            # add the whole record to our result list
            result_list.append(record)
    # return the whole list of results, could be empty
    return result_list

# put it together
user_hobbies_lines = readFile('user_hobbies.txt')
user_hobbies_list = create_user_hobbies_list(user_hobbies_lines)
result_hobbies_list = search_hobbies(user_hobbies_list,"Watching movies")
print result_hobbies_list
```

## Activity #3 (10 min)

- Write the code that appends the following list named *user\_hobbies\_more* to *user\_hobbies* list, and write the components in *user\_hobbies* in a file named *user\_hobbies\_updated.txt*. Use '\*' for a delimiter.

```
user_hobbies_more= [  
    {'id':'user12', 'name':"Jeff", 'hobby':"Fishing"},  
    {'id':'user13', 'name':"Jevin", 'hobby':"Boxing"},  
    {'id':'user14', 'name':"Joffery", 'hobby':"Painting"},  
    {'id':'user15', 'name':"Jock", 'hobby':"Watching movie"}  
]
```

# Sample Code

```
# need to assume we have user_hobbies_more in a variable
# Creating a procedure that appends to a list is easy
#
def append_user_hobbies(user_hobbies_list=None,more_hobbies_list=None):
    for record in more_hobbies_list:
        user_hobbies_list.append(record)
    return user_hobbies_list

# need a procedure that converts to a list of lines
def convert_user_hobbies(user_hobbies_list=None):
    user_hobbies_lines = list()
    for record in user_hobbies_list:
        user_hobbies_string = "%s*%s*%s"%(record['id'],record['name'],record['hobby'])
        user_hobbies_lines.append(user_hobbies_string)
    return user_hobbies_lines

# we can lift, reuse the example code to write files
# now put it together ...

new_user_hobbies_list = append_user_hobbies(user_hobbies_list,user_hobbies_more)

user_hobbies_output_strings = convert_user_hobbies(new_user_hobbies_list)

writeFile("user_hobbies_updated.txt",user_hobbies_output_strings)
```

---

## Activity #4 (10 min)

- Now read the *user\_hobbies\_updated.txt* you've created, remove users' records whose hobby is 'Having dinner', and save the updated records to *user\_hobbies\_updated2.txt* again. Use '\*' for a delimiter.