PYTHON COURSE

ENTRY LEVEL

Basics of programming in Python 3.10

This course will cover part of the arguments found in PCEP™ – Certified Entry-Level Python Programmer Certification

DATA COLLECTIONS: DICTIONARIES

DICTIONARIES

- data structure that **associate informations**
- Dictionaries are mutable
- ordered collection composed by **key-value pairs**
- **every key has a value stored in it, we can use the key to access the value**
- **Keys must be immutable types, values can be all types**

DICTIONARIES

Dictionaries are built using curly brakets

```python
user_data = {
    'name': "Mark",
    'surname': "Bennet",
    'age': 28,
}

print(user_data["name"]) # Prints "Mark"
```

key and values are divided by :
pairs are divided by commas

You can access dictionary values by their keys between square brackets

DICTIONARIES

you can handle errors by using .get() method of dictionaries to access elements

```python
user_data = {
    'name': "Mark",
    'surname': "Bennet",
    'age': 28,
}

print(user_data["title"]) # Gives error
print(user_data.get("title", "This key is not in the dictionary")) # Gives a print of second argument
```

trying to access a key that is not in the dictionary, will raise a KeyError, use .get() to handle this

DICTIONARIES

add a new key-value pair with item assignement

```
user_data = {
    'name': "Mark",
    'surname': "Bennet",
    'age': 28,
}

user_data['title'] = "Mr."
print(user_data['title']) # Prints 'Mr.'
```

DICTIONARIES

modify values accessing it by their keys

```
user_data = {
    'name': "Mark",
    'surname': "Bennet",
    'age': 28,
}

user_data['name'] = "John"
print(user_data['name']) # Prints 'John'
```

Duplicated keys is not allowed in dictionaries

DICTIONARIES

- delete values from dictionaries using *del* keyword

```
user_data = {
    'name': "Mark",
    'surname': "Bennet",
    'age': 28,
}

del user_data['name']
print(user_data['name']) # gives keyError
```

DICTIONARIES

delete elements using .*pop*() method consent to assign removed value to a variable

```
user_data = {
    'name': "Mark",
    'surname': "Bennet",
    'age': 28,
}

deleted = user_data.pop('name')

del user_data['title'] # gives KeyError

user_data.pop('title', 'Key not in the dictionary') # Prints 'Key not in the dictionary'
```

you can pass a second argument to *pop*() method to handle KeyErrors

ITERATE ON DICTIONARIES

iterate using .*keys*() method (returns a list containing all the keys to iterate on it)

```
user_data = {
    'name': "Mark",
    'surname': "Bennet",
    'age': 28,
}

for key in user_data.keys():
    print(user_data[key]) # Prints Mark, Bennet, 28
```

ITERATE ON DICTIONARIES

iterate using .values*() method (returns a list containing all the values to iterate on it)*

```
user_data = {
    'name': "Mark",
    'surname': "Bennet",
    'age': 28,
}

for value in user_data.values():
    print(value) # Prints Mark, Bennet, 28
```

ITERATE ON DICTIONARIES

iterate using .items*() method (returns a tuple of tuples(key, value) to unpack)*

```
user_data = {
    'name': "Mark",
    'surname': "Bennet",
    'age': 28,
}

for key, value in user_data.items():
    print(key, value, sep=": ") # name: Mark surname: Bennet age: 28
```

NESTING DICTIONARIES

you can nest dictionaries into lists

```
site_users = [
    {
    'name': "Mark",
    'surname': "Bennet",
    'age': 28,
    },
    {
    'name': "John",
    'surname': "Rockford",
    'age': 13,
    },
    {
    'name': "Danielle",
    'surname': "Bennet",
    'age': 45,
    },

]


for user in site_users:
    print("User data:")
    for key, value in user.items():
        print(key, value, sep = ": ")
```

QUESTION TIME

PYTHON

QUESTION 1:

What will be the output of this code:tion?

```
a = [1]
b = a
a[0] = 0
print(b)
```

- [0]
- 0
- [1]
- 1

PYTHON

QUESTION 2:

What will be the output of this code:tion?

```
a = [1]
b = a[:]
a[0] = 0
print(b)
```

- [0]
- 0
- [1]
- 1

PYTHON

QUESTION 3:

What will be the output of this code:tion?

```
lst = [1, 2, 3, 4]
lst = lst[-3:-2]
lst = lst[-1]
print(lst)
```

- 4
- 2
- 3
- 1

PYTHON

QUESTION 4:

How many elements will be in list2?

```
list1 = [False for i in range(1, 10)]
list2 = list1 [-1:1:-1]
```

- zero
- five
- seven
- three

EXERCISES