Session 8 - Les dictionnaires - A.U. 23-24

November 29, 2023

[1]:
```
grp5 = {
    "name": "g5",
    "nb": 15,
    "list_etuds": ["isra", "hiba", "ahmed"],
    "math_exam": True,
}
len(grp5)  # --> 4
```

[1]:
```
4
```

[2]:
```
grp5["name"]  # --> 'g5'
```

[2]:
```
'g5'
```

[3]:
```
grp5["name"] = "G5"  # ecraser la valeur de la clé 'name' par 'G5' au lieu de␣
 ↪'g5'
# --> {'name': 'G5', 'nb': 15, 'list_etuds': ['isra', 'hiba', 'ahmed'],␣
 ↪'math_exam': True}
grp5
```

[3]:
```
{'name': 'G5',
 'nb': 15,
 'list_etuds': ['isra', 'hiba', 'ahmed'],
 'math_exam': True}
```

[4]:
```
grp5["moy"] = 17.5
grp5[1] = "val1"
grp5["1"] = "val"
grp5[True] = "bjr"
grp5[15.5] = "float"
grp5[(1, 2)] = "float"
grp5[range(1, 20)] = "float"
# --> {'name': 'G5', 'nb': 15, 'list_etuds': ['isra', 'hiba', 'ahmed'],␣
 ↪'math_exam': True, 'moy': 17.5, 1: 'val1', '1': 'val', True: 'bjr', 15.5:␣
 ↪'float', (1, 2): 'float', range(1, 20): 'float'}
grp5
```

[4]:
```
{'name': 'G5',
 'nb': 15,
 'list_etuds': ['isra', 'hiba', 'ahmed'],
 'math_exam': True,
 'moy': 17.5,
 1: 'bjr',
 '1': 'val',
 15.5: 'float',
 (1, 2): 'float',
 range(1, 20): 'float'}
```

[5]:
```
# --> ['isra', 'hiba', 'ahmed', 'oumaima']
grp5["list_etuds"].append("oumaima")
# --> {'name': 'G5', 'nb': 15, 'list_etuds': ['isra', 'hiba', 'ahmed',␣
 ↪'oumaima'], 'math_exam': True, 'moy': 17.5, 1: 'val1', '1': 'val', True:␣
 ↪'bjr', 15.5: 'float', (1, 2): 'float', range(1, 20): 'float'}
grp5
```

[5]:
```
{'name': 'G5',
 'nb': 15,
 'list_etuds': ['isra', 'hiba', 'ahmed', 'oumaima'],
 'math_exam': True,
 'moy': 17.5,
 1: 'bjr',
 '1': 'val',
 15.5: 'float',
 (1, 2): 'float',
 range(1, 20): 'float'}
```

[6]:
```
# Les clés doivent être immuables. C'est à dire que les clés ne doivent pas␣
 ↪être modifiables
# Ces instructions ne sont pas valides et generent des erreurs
grp5[[1, 2]] = "list"
grp5[{"a", 1}] = "abc"
grp5[{"a": 1}] = 10
```

[7]:
```
"name" in grp5  # 'name' in grp5.keys() #--->True
"khales" in grp5  # --->False
"G5" in grp5  # grp5['khales'] ---> KeyError: 'khales'
if "khales" in grp5:
    grp5["khales"]  # --->
grp5.get("name")  # if 'khales' in grp5:grp5['name']#--->G5
grp5.get("name", "cle mouch mawjoud")  # --->G5
grp5.get("namesssssssssss", "cle mouch mawjoud")  # --->cle mouch mawjoud
```

[7]:
```
'cle mouch mawjoud'
```

[8]:
```
# <dict>.pop(<key>[,default]) : supprime la clé <key> et retourne sa valeur. Si␣
 ↪la clé n'existe pas, retourne <default> ou lève une exception si <default>␣
 ↪n'est pas fourni.
grp5.pop("name")  # supprime la clé 'name' et retourne sa valeur --->'G5'
```

[8]:
```
'G5'
```

[9]:
```
str( grp5 )  # --->"{'nb': 15, 'list_etuds': ['isra', 'hiba', 'ahmed'],␣
 ↪'math_exam': True, 'moy': 17.5, 1: 'val1', '1': 'val', True: 'bjr', 15.5:␣
 ↪'float', (1, 2): 'float', range(1, 20): 'float'}"
list( grp5 )  # --->['nb', 'list_etuds', 'math_exam', 'moy', 1, '1', True, 15.
 ↪5, (1, 2), range(1, 20)]
tuple( grp5 )  # --->('nb', 'list_etuds', 'math_exam', 'moy', 1, '1', True, 15.
 ↪5, (1, 2), range(1, 20))
set( grp5 )  # --->{1, 15.5, '1', 'nb', 'moy', 'math_exam', (1, 2), True,␣
 ↪range(1, 20), 'list_etuds'}
```

[9]:
```
{(1, 2), 1, '1', 15.5, 'list_etuds', 'math_exam', 'moy', 'nb', range(1, 20)}
```

[10]:
```
# --->dict_keys(['nb', 'list_etuds', 'math_exam', 'moy', 1, '1', True, 15.5,␣
 ↪(1, 2), range(1, 20)])
grp5.keys()
```

[10]:
```
dict_keys(['nb', 'list_etuds', 'math_exam', 'moy', 1, '1', 15.5, (1, 2),
range(1, 20)])
```

[11]:
```
d = {
    "etud1": "isra",
    "etud2": "isra",
    "etud3": "hiba",
    "etud4": "isra",
    "etud5": "ahmed",
    "etud6": "isra",
}
list(d.values()).count("isra")  # --->4
```

[11]:
```
4
```

[12]:
```
# <dict>.items() : retourne une vue sur les couples (clé, valeur) du␣
 ↪dictionnaire.
# --->dict_items([('etud1', 'isra'), ('etud2', 'isra'), ('etud3', 'hiba'),␣
 ↪('etud4', 'isra'), ('etud5', 'ahmed'), ('etud6', 'isra')])
d.items()
```

[12]:
```
dict_items([('etud1', 'isra'), ('etud2', 'isra'), ('etud3', 'hiba'), ('etud4',
'isra'), ('etud5', 'ahmed'), ('etud6', 'isra')])
```

Parcourir un dictionnaire

[13]:
```
for k in grp5.keys():
    print(k)
```

```
nb
list_etuds
math_exam
moy
1
1
15.5
(1, 2)
range(1, 20)
```

[14]:
```
for k in grp5.values():
    print(k)
```

```
15
['isra', 'hiba', 'ahmed', 'oumaima']
True
17.5
bjr
val
float
float
float
```

[15]:
```
for k in grp5.items():
    print(k)
```

```
('nb', 15)
('list_etuds', ['isra', 'hiba', 'ahmed', 'oumaima'])
('math_exam', True)
('moy', 17.5)
(1, 'bjr')
('1', 'val')
(15.5, 'float')
((1, 2), 'float')
(range(1, 20), 'float')
```

[16]:
```
# for key,value in grp5.items():
for key, value in grp5.items():
    print(key, value)
```

```
nb 15
list_etuds ['isra', 'hiba', 'ahmed', 'oumaima']
math_exam True
moy 17.5
```

```
1 bjr
1 val
15.5 float
(1, 2) float
range(1, 20) float
```

[17]:
```python
# for hajja in grp5:
#     # print(hajja, 'est une autre cle')
#     print(f'"{hajja}" est une autre cle')

# for key in grp5:
# for key in list(grp5):
# for key in tuple(grp5):
# for key in set(grp5):
for key in grp5.keys():
    # print(key, 'est une autre cle')
    print(f'"{key}" est une autre cle. et sa valeur est {grp5[key]}')
```

```
"nb" est une autre cle. et sa valeur est 15
"list_etuds" est une autre cle. et sa valeur est ['isra', 'hiba', 'ahmed',
'oumaima']
"math_exam" est une autre cle. et sa valeur est True
"moy" est une autre cle. et sa valeur est 17.5
"1" est une autre cle. et sa valeur est bjr
"1" est une autre cle. et sa valeur est val
"15.5" est une autre cle. et sa valeur est float
"(1, 2)" est une autre cle. et sa valeur est float
"range(1, 20)" est une autre cle. et sa valeur est float
```

[18]:
```python
d1 = {}  # d1=dict() #d1=dict([]) #dictionnaire vide
d1["a"] = 10  # ajouter la cle 'a' avec la valeur 10
d1
```

[18]:
```
{'a': 10}
```

[19]:
```python
d3 = {
    "a": 1,
    "a": 10,
    "a": 100,
}  # --->{'a': 100} car les cles doivent etre uniques. Et la derniere valeur␣
 ↪est prise en compte
d3
```

[19]:
```
{'a': 100}
```

[20]:
```python
# <dict>.popitem() : supprime la dernière clé et retourne un tuple (clé,valeur)
grp5.popitem()  # --->((1, 2), 'float')
```

[20]:
```
(range(1, 20), 'float')
```

[21]:
```
grp5.pop()  # TypeError: pop expected at least 1 arguments, got 0
grp5.pop( "name" )  # --->KeyError: 'name' # car la cle 'name est supprimée␣
 ↪dans la ligne précédente
grp5.pop("name", "en cas mouch mawjoud")  # --->'en cas mouch mawjoud'
```

[22]:
```
# pour supprimer toutes les clés et leurs valeurs element par element, on peut␣
 ↪utiliser la boucle while
while len(grp5) > 0:
    grp5.popitem()
```

[2]:
```
d0 = {"a": 1, "b": 2, "c": 3}
d1 = {"c": 30, "e": 40, "f": 50}
d0.update(d1)
d0
```

[2]:
```
{'a': 1, 'b': 2, 'c': 30, 'e': 40, 'f': 50}
```