Lesson06

Les DICTIONNAIRES ET ENSEMBLES.

* DICTIONNAIRES.

1. **Définition**

Liste MUABLE d’éléments CONTENEUR (PAS *une séquence*

1. **Comparaisons**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Objets** | **Listes** | **Tableaux** | **Tuples** | **dictionnaires** |
| Définition | séquence d'éléments, dont les types peuvent être différents | Liste de listes | = liste **IMMUABLE** d’éléments. Ici, :   * séparés par une virgule * Entre parenthèse (*pas obligatoire*) | Liste MUABLE d’éléments  CONTENEUR (PAS *une séquence*) |
| Structure | * Séquence * Éléments divers * Précédés d’un **INDEX**. | * Séquence * Éléments divers * Précédés d’un INDEX | * Séquence * Éléments divers * Précédés d’un INDEX | * Conteneur * Éléments divers * Précédés d’une **CLEF**. |
| Type de **clef** ou d’**index**. |  |  |  | * String, entier ou tuple * De type immuable |
| Muabilité | Oui | Oui | Non | Oui |
| Accès aux élts | Via indice | Via indice | Via indice | Via clef |
|  |  |  |  |  |

1. **Dans les dictionnaires**

|  |  |  |  |
| --- | --- | --- | --- |
| **Nr** | **Objet** | **DICTIONNAIRES** | **Explication / comparaison (*avec liste*)** |
| 1 | Forme | dict = **{**clé 1 : valeur 1, clé 2 : valeur 2, ..., clé n : valeur n**}** | **CONTENEUR** (**PAS** *une ‘SEQUENCE*’) |
| 2 | Modification | dict**[**clé i**]** = valeur i | Màj de la valeur d’un élément d’une *nième* clef |
| 3 | Comparaison liste / dico | couleur = {0:'rouge', 1:'orange', 2:'jaune', 3:'vert', 4:'bleu', 5:'indigo', 6:'violet'} | couleur = ['rouge', 'orange', 'jaune', 'vert', 'bleu', 'indigo', 'violet'] |
| * Type | Type couleur   * <class 'dict'> | Type couleur   * <class 'list'> |
| 4 | Sélection | Couleur[2]   * Jaune | Couleur[2]   * Jaune |
| 5 | Exemple | couleur = {'red':'rouge', 'orange':'orange', 'yellow':'jaune', 'green':'vert', 'blue':'bleu', 'indigo':'indigo', 'violet':'violet'}  >>> couleur[green]   * Vert |  |
| 6 | Création | dct = {}  dct['Prénom'] = 'Gérard'  dct['Nom'] = 'Menfroid'  dct['Age'] = 30   * Dct * {'Prénom': 'Gérard', 'Age': 30, 'Nom': 'Menfroid'} |  |
| 7 | Modification | dct['Nom'] = 'Menchot' |  |
|  | Opérations | | |
| 8 | Longueur (= nb d’élts) | Len(dict)   * 3 |  |
| 9 | In | 'Prenom' in dict   * True   ‘autre’ in dict   * False | Opérateur « in » qui NE s’applique QUE sur les CLEFS |
| 10 | Boucle FOR | For i in dict  Print (i, dict[i])   * Prénom Gérard * Age 30 * Nom Menchot | Énumération des élts contenus via la boucle FOR … IN |
| 11 | Remplissage | client = {'1':['Mendfroid', 'Gerard', '0478 65 33 00'],  '2' :['Lergrave', 'Oussama', '0476 75 35 82'],  …} | Remplissage de tableau contenant ***une clef*** et ***plusieurs champs*** par enregistrement. |

1. **Méthodes sur les DICTIONNAIRES.**

|  |  |  |
| --- | --- | --- |
| **Méthodes** | **Résultat** | **Python** |
| dct.**get**(clé, valeur par défaut) | Retourne la valeur qui correspond à la clé. Retourne une valeur par défaut si la clé n'existe pas. | dct = {1:'un', 2:'deux', 3:'trois'}  dct.get(3, 'coucou')   * 'trois'   dct.get(4, 'coucou')   * 'coucou' |
| dct2 = dct1.**copy**() | Fait une vraie **copie** du dictionnaire | >>> dct1 = {1: 'un', 2: 'deux', 3: 'trois'}  >>> dct2 = dct1.copy()  >>> dct2   * {1: 'un', 2: 'deux', 3: 'trois'}   >>> dct1[2] = 'coucou'  >>> dct1   * {1: 'un', 2: 'coucou', 3: 'trois'}   >>> dct2   * {1: 'un', 2: 'deux', 3: 'trois'} |
| dct.**clear**() | Supprime tous les éléments du dictionnaire |  |
| dct.**pop**(clé) | Supprime la valeur désignée par sa clé, et retourne la valeur supprimée |  |
| dct.**update**(dct2) | Ajoute les couples clé – valeur d'un autre dictionnaire |  |
| dct.**keys**() | Retourne les clés du dictionnaire |  |
| dct.**values**() | Retourne les valeurs du dictionnaire |  |
| dct.**items**() | Retourne les clés et les valeurs du dictionnaire |  |

Les DICTIONNAIRES ET ENSEMBLES.

* ENSEMBLES.

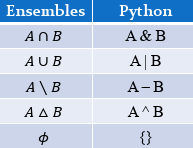
1. **Définition**

ENSEMBLES représentent des **COLLECTIONS** d’**OBJETS**.

1. **Comparaison**

|  |  |  |
| --- | --- | --- |
| **Objets** | **Ensembles** | **dictionnaires** |
| Définition | ENSEMBLES représentent des **COLLECTIONS** d’**OBJETS**. | Liste MUABLE d’éléments  CONTENEUR (PAS *une séquence*) |
| Structure | * Éléments d’**un set** sont des **clefs**. | * Conteneur * Éléments divers * Précédés d’une **CLEF**. |
| Syntaxe | E = {clé 1, clé 2, ..., clé n} | dct = {1:'un', 2:'deux', 3:'trois'} |
|  |  |  |
|  |  |  |
|  |  |  |

1. **Ensembles :**



1. **Exemples**

soit 𝒰 l'ensemble universel contenant les nombres d'un dé à jouer. 𝒰 = {1, 2, 3, 4, 5, 6}

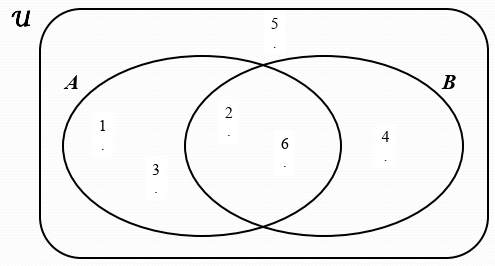
* + - A est l'ensemble des diviseurs de 6.

A = {1, 2, 3, 6}

* + - B est l'ensemble des nombres pairs du dé.

B = {2, 4, 6}

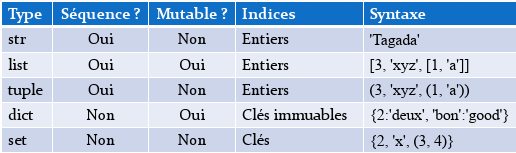
* + Ces ensembles :



RESULTATS :

|  |  |
| --- | --- |
| Éléments | Types |
| >>> A = {1, 2, 3, 6} | >>> type(A)  <class 'set'> |
| >>> B = {2, 4, 6} | >>> A & B  {2, 6} |
| >>> A | B | {1, 2, 3, 4, 6} |
| >>> A – B | {1, 3} |
| >>> B – A | {4} |
| >>> A ^ B | {1, 3, 4} |
| >>> A = {'a', 'b', 'c'} | >>> type (A)  <class 'set'> |
| >>> B = {'b', 'c', 'd', 'e'}  >>> A & B | {'b', 'c'} |
| >>> A | B | {'b', 'c', 'e', 'a', 'd'} |
| >>> A – B | {'a'} |
| >>> B – A | {'e', 'd'} |
| >>> A ^ B | {'e', 'a', 'd'} |

1. **Comparaison des conteneurs.**



**CONVERSIONS**