

Gonzalo Fernández
IWI-131 24/10/2017

IWI-131 24/10/2017

Temas de la Ayudantía de Hoy

- Conjuntos
- Trabajando con Conjuntos
- Diccionarios
- Funciones de los Diccionarios
- Trabajando con Diccionarios
- Ejercicios

Conjuntos

- Arreglos **no** ordenados donde los elementos no se pueden repetir, estos pueden ser de cualquier tipo
 - Se inicializan con la función **set**, ó, con **{}**
 - `a = set([1, 3, 3, 3, 3])` `# set([1, 3])`
 - `a = {1, 3, 3, 3}` `# set([1, 3])`
- Permiten realizar operaciones matemáticas de conjuntos, estas son:
 - Intersección: `a & b`
 - Unión: `a | b`
 - Resta: `a - b` `# Elementos de a que no estén en b`
 - Exclusión: `a ^ b` `# Elementos en a y b pero no ambos`
 - Subconjunto: `a < b` `# ¿Es a subconjunto de b?`

Trabajando con Conjuntos

- Agregar un elemento: **a.add(valor)**
- Remover un elemento: **a.remove(valor)**
- ¿ Existe v en a?: **v in a**
- Largo del conjunto: **len(a)**

```
a = set("ohayou gozaimasu progra-sama")  
set(['o', 'h', ' ', 'a', 'y', 'u', 'g', 'z', 'i', 'm', 's', 'p', 'r', '-'])
```

```
for i in a:  
    print i
```


Diccionarios

- Estructura de Python que nos permite almacenar datos (de cualquier tipo) y asignarles una llave única a estos
- La llave única puede ser un número, un string, ó, un flotante.
 - Se inicializan con la función **dict**, ó, con **{ llave : valor }**
 - `a = dict()` # esto sirve para clonar diccionarios
 - `a = {"llave1" : 1, 2: "l", 23:2}` # {'llave1': 1, 2: 'l', 23: 2}
 - `a = {}` # {}
 - `a[123] = "gomenasai"`
 - `a[123.1] = 3`
 - `a["123"] = [1,2,3]`
 - `a["dic1"] = { 5 : 1, 4 : "omae wa, mou shindeiru" }`

Funciones de Diccionarios

- `d.keys ()` ó `list(d)`:
 - Devuelve una **lista** con las llaves del diccionario
- `d.values ()`:
 - Devuelve una **lista** con los valores del diccionario
- `d.items ()`:
 - Devuelve una **lista de tuplas** con la llave y valor de cada elemento del diccionario

Funciones de Diccionarios

- **len(d):**
 - Devuelve la cantidad de elementos del diccionario.
- **del d[llave]:**
 - Elimina el elemento asociado a la llave ingresada.
- **llave in d:**
 - Devuelve **True** en caso de existir la **llave** ingresada en el diccionario **d**.

Trabajando con Diccionarios

- Recorrer un diccionario:

```
for i in diccionario:           # diccionario.keys() | list(diccionario)
    print diccionario[i]
```

```
for i in diccionario.values():
    print i
```

```
for llave, valor in diccionario.items():
    print "La llave", llave, "tiene el valor", valor
```


Trabajando con Diccionarios

```
Notas = [ ("Fernández", 100), ("Fernández", 100),  
          ("Fulanito", 28), ("ダークマトリックス", 0),  
          ("Fulanito", 0), ("ダークマトリックス": 0),  
          ("ダークマトリックス", 0), ("Kriz", 7) ]
```

```
a = {}
```

```
for nombre, nota in Notas:
```

```
    if nombre not in a:
```

```
        a[nombre] = {}
```

```
        a[nombre]["cantidad"] = 0
```

```
        a[nombre]["nota"] = 0
```

```
    a[nombre]["cantidad"] += 1
```

```
    a[nombre]["nota"] += nota
```

```
for nombre, datos in a.items():
```

```
    promedio = datos["nota"] / datos["cantidad"]
```

```
    a[nombre]["promedio"] = promedio
```

```
    print "El promedio de:", nombre, "es:", promedio
```

Ejercicios

- La Universidad Tropical Santa Filomena necesita que usted implemente el programa **maspera** que recibe una **lista de tuplas** con los nombres y notas de los alumnos y devuelva una lista filtrada con los nombres de alumnos con mayor probabilidad de reprobar sus ramos ordenada en base al promedio de sus notas.

```
Alumnos = [ ("Gonzalo Fernández", 0),  
             ("Mauricio Solar", 100)  
             ... ]
```

Ejercicios

- Usted se ha titulado de asesino serial y es muy metódico con sus víctimas, diseñe un algoritmo que reciba un diccionario con la información y vida diaria de sus víctimas que retorne una lista con los nombres de sus víctimas, ordenada sobre sus preferencias al asesinar.

Victimas = { "Esteban Sáez" : { "HorarioSalida" : 12.24
}