Camino al Certamen 2: Resumen de Contenidos

Funciones

def *nombre(args):*	Declarar (crear) una función
return *valor*	Devolver un valor desde una funcion *El return es opcional en una función.*
nombre(valores)	Uso de una función

Módulos

Importar un módulo completo from *modulo* import * import *modulo*
Importar funciones específicas de un módulo from *modulo* import *funcion1, funcion2, …*

Listas

[*obj1, obj2, ...*] list(*obj_iterable*) [] (vacío)	Crear una lista
lista [*i*]	Obtener el elemento en la posición *i* de *lista*
lista [*i*]=*obj*	Modificar el elemento en la posición *i* de *lista*
lista .append(*obj*)	Agregar el elemento *obj* al final de *lista*
sum(*lista*)	Suma todos los elementos de *lista*
lista1 + *lista2*	Une *lista1* con *lista2*
lista * *n*	Repite *n* veces *lista*
obj (not) in *lista*	Conocer si *obj* (no) se encuentra en *lista*
lista [*i* :*j*]	Obtener desde el elemento *i* hasta el elemento *j-1* de *lista* , en una lista nueva
lista .count(*obj*)	Cuenta cuántas veces está *obj* en *lista*
lista .index(*obj*)	Entrega la posición de *obj* en *lista*
lista .remove(*obj*)	Elimina *obj* de *lista*
del *lista* [*i*]	Elimina el elemento en *i* de *lista*
lista .reverse()	Invierte *lista*
lista .sort()	Ordena *lista* de menor a mayor

Tuplas

(*obj1, obj2, ...*)	Crear una tupla
obj1, obj2, … = tupla *tupla* [*i*]	Desempaquetar una tupla
tupla1 == tupla2 *tupla1 < tupla2* *tupla1 > tupla2*	Comparación entre tuplas
tupla .count(*obj*)	Cuenta cuántas veces está *obj* en *tupla*
tupla .index(*obj*)	Entrega la posición de *obj* en *tupla*

Diccionarios

{*llave1* : *valor1, ...* }	Crear un diccionario
{} *(vacío)*	
dict() *(vacío)*	
dicc [*llave*]	Obtener el *valor* asociado a *llave*
dicc[llave]=*valor*	Asignar *valor* a una *llave* nueva o existente
del *dicc* [*llave*]	Eliminar *llave*
list(*dicc*)	Lista con las llaves de *dicc*
list(*dicc* .values())	Lista con los valores de *dicc*
len(*dicc*)	Cantidad de pares llave-valor en *dicc*
Iterar sobre las llaves	
for *llave* in *diccionario*	
Iterar sobre los valores	
for *valor* in *diccionario* .values()	
Iterar sobre las llaves y los valores simultáneamente	
for *llave* , *valor* in *diccionario* .items()	

Conjuntos

{*obj1, obj2, ...* }	Crear un conjunto
set(*obj_iterable*)	
set() *(vacío)*	
len(*conj*)	Cantidad de elementos de *conj*
obj (not) in *conj*	Conocer si *obj* (no) se encuentra en *conj*
conj .add(*obj*)	Agrega *obj* a *conj* , sólo si no se encuentra en el conjunto
conj1 & *conj2*	Intersección entre *conj1* y *conj2*
conj1 | *conj2*	Unión entre *conj1* y *conj2*
conj1 - *conj2*	Diferencia entre *conj1* y *conj2* (lo que está en *conj1* pero no en *conj2*)
conj1 ^ *conj2*	Diferencia Simétrica entre *conj1* y *conj2* *(elementos en conj1* o *conj2* pero no en ambos)
conj1 < *conj2*	Es *conj1* subconjunto de *conj2* ?
conj1 <= *conj2*	Es *conj1* subconjunto de (o igual a) *conj2* ?