

Programación Avanzada

IIC2233 2025-2

Cristian Ruz - Pablo Araneda - Francisca Ibarra - Tamara Vidal - Daniela Concha



Anuncios

21 de agosto de 2025



1. Tarea 1 se entrega (sin atraso) este domingo.
2. Hoy haremos la Actividad 2.
 - Recuerden firmar la lista y asistir a la sección en la que están inscritos, si quedan ausentes tendrán 0 puntos.

Introducción y tuplas

Introducción y tuplas

- ¿Qué es una estructura de datos secuencial?
- ¿Para qué usar tuplas?
- Desempaquetado
- ¿Para qué usar namedtuples?



Ejemplo de tuplas

```
nuevo_numero = (6, )
data_desactualizada = (5, 0, 4, 1, 3, 2)

# ¿Qué pasaría si hubiésemos tenido nuevo_numero = 6?
data = data_desactualizada + nuevo_numero

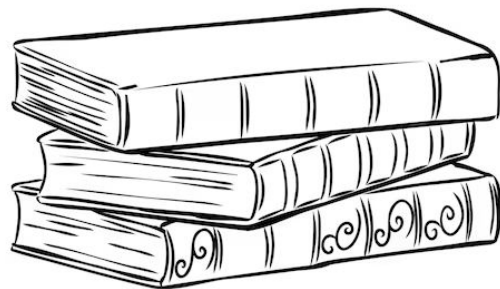
# ¿Por qué no usamos data[-1] en vez de data[-1:]?
data_ordenada = data[1::2] + data[-3::-2] + data[-1:]

minimo, *otros_numeros, maximo = data_ordenada
# ¿Es otros_numeros una tupla?
```

Stacks y colas

Stacks y colas

- ¿Cómo se representa un *stack* en python?
- ¿Cómo se representa una cola en python?
- ¿Cuáles son las diferencias entre *stack* y cola?



Ejemplo de stacks y colas

¿Cuál funciona de la forma deseada aquí?

```
from collections import deque

stack_reproduccion = ["nirvana feat. Tia"]
cola_reproduccion = deque(stack_reproduccion)

stack_reproduccion.append("Odo")
cola_reproduccion.append("Odo")

primera_cancion_stack = stack_reproduccion.pop()
primera_cancion_cola = cola_reproduccion.popleft()
```


Diccionarios

Diccionarios

- Entre llave y valor, ¿cuál tiene una restricción?
- ¿Para qué sirven?
- defaultdict
- ¿Son mutables?, ¿y *hasheables*?



Ejemplo de diccionarios

```
from collections import defaultdict

alumnos_cursos_nivel_miles = defaultdict(int)

cursos_dcc = { "IIC1103": ["Intro", 1216],
               "IIC2233": ["Avanzada", 528],
               "IIC1253": ["Discretas", 353],
               "IIC1005": ["Exploratorio", 139] }

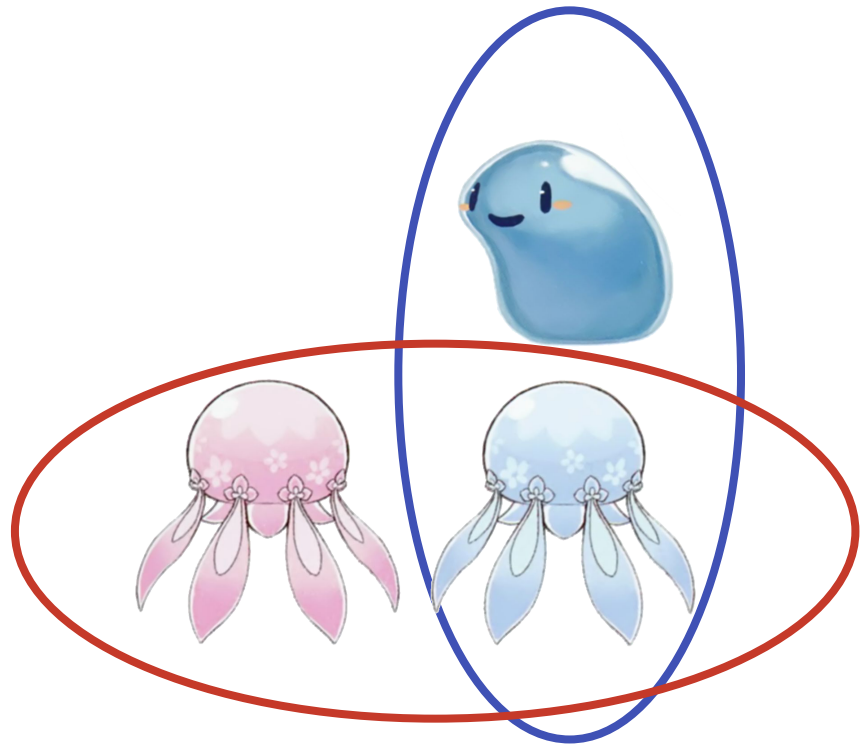
for sigla, (nombre, cantidad_alumnos) in cursos_dcc.items():
    alumnos_cursos_nivel_miles[sigla[3]] += cantidad_alumnos

print("Alumnos cursos IIC nivel 1000:", alumnos_cursos_nivel_miles['1'])
```

Sets

Sets

- ¿Para qué sirven?
- ¿Qué pueden contener?
- ¿Son mutables?, ¿y hasheables?



Ejemplo sets

¿A qué pregunta responde cada consulta?

```
profesores = {"Cristian", "Fran", "Pablo", "Dani", "Tama", "Tama"}

estudiantes_magister = set(("Pablo", "Dani"))
alejandras = {"Fran", "Dani"}
nombres_acortados = set(["Fran", "Tama", "Dani"])

consulta1 = alejandras | estudiantes_magister
consulta2 = alejandras ^ estudiantes_magister
consulta3 = profesores - nombres_acortados
consulta4 = estudiantes_magister & alejandras & nombres_acortados
consulta5 = alejandras.issubset(profesores)
```

args y kwargs

args y kwargs

- ¿Cuál es la diferencia entre argumento y parámetro?
- ¿Para qué sirven los args y kwargs?
- ¿Importa el orden en el que se usan?



Ejemplo args y kwargs

```
def elevar(numero, exponente):  
    return numero ** exponente  
  
def cuadrado_y_cubo(numero):  
    cubo = elevar(**{'exponente': 3, 'numero': numero})  
    return {'cuadrado': elevar(numero, 2), 'cubo': cubo}  
  
def es_sucesion_fibonacci(primerο, segundo, tercero=None, *otros):  
    if tercero is None:  
        return True  
    viene_fibonacci = es_sucesion_fibonacci(segundo, tercero, *otros)  
    return (primerο + segundo == tercero) and viene_fibonacci  
  
cuadrado_y_cubo(4)  
es_sucesion_fibonacci(0, 1, 1, 2, 3, 5, 8)
```

Resumen propiedades

Estructuras secuenciales

Estructura	Mutable	Hasheable	
Lista	✓	✗	Permiten agregar, eliminar, modificar elementos.
Tupla	✗	✓ *	Útiles para retornar múltiples valores y contener valores que no cambian.
<i>Named Tuple</i>	✗	✓ *	Tuplas donde se puede acceder a cada posición mediante un nombre.
Colas	✓	✗	Eficientes para insertar o retirar desde sus extremos.
<i>Stacks</i>	✓	✗	Inserción y eliminación son siempre en el tope del <i>stack</i> .

✓*: Depende de si su contenido es *Hasheable*

Acciones

Estructura	Insertar	Búsqueda por índice	Búsqueda por llave	Búsqueda por valor
Lista	✓	✓✓✓	✗	✓
Tupla	✗	✓✓✓	✗	✓
Diccionario	✓✓✓	✗	✓✓✓	✓
Set (Conjunto)	✓✓✓	✗	✓✓✓	✗

✓✓✓ Se puede y es eficiente.

✓ Se puede pero no siempre es eficiente.

✗ No se puede.

Veamos una pregunta de Evaluación Escrita

Tema: Estructuras de datos (Midterm 2023-2)

7. ¿Cuál(es) de la(s) siguiente(s) afirmación(es) es/son correctas respecto a las tuplas?
- I. Son estructuras de datos mutables.
 - II. Pueden desempaquetarse en variables independientes.
 - III. Se puede hacer slicing sobre una tupla.
 - IV. Se puede acceder a sus elementos a través de índices.
-
- A) Solo I
 - B) Solo III
 - C) II y III
 - D) II, III, IV
 - E) I, II, III y IV

Veamos una pregunta de Evaluación Escrita

Tema: Estructuras de datos (Midterm 2023-2)

7. ¿Cuál(es) de la(s) siguiente(s) afirmación(es) es/son correctas respecto a las tuplas?
- I. Son estructuras de datos mutables.
 - II. Pueden desempaquetarse en variables independientes.
 - III. Se puede hacer slicing sobre una tupla.
 - IV. Se puede acceder a sus elementos a través de índices.
-
- A) Solo I
 - B) Solo III
 - C) II y III
 - D) II, III, IV**
 - E) I, II, III y IV

Actividad

Comentarios AC2

- El “`if __name__ == '__main__':`” al final del archivo `main.py` es para que puedan probar el código. Igualmente, también pueden usar los test.
- Todas las funciones de la actividad son **independientes**, No necesitan completas las funciones anteriores, para hacer una que aparece más adelante en el enunciado.

NO GIT PUSH NO GAIN!

Programación Avanzada

IIC2233 2025-2

Cristian Ruz - Pablo Araneda - Francisca Ibarra - Tamara Vidal - Daniela Concha

