

Auxiliar 1: Clases y Arreglos en Python

Profesores: Jérémy Barbay, Nelson Baloian, Patricio Poblete
Auxiliares: F. Giovanni Sanguinetti, Auxiliar 2, Auxiliar 3, Auxiliar 4

Pregunta 1

El objetivo de esta pregunta es construir la clase Pokémon, para así recordar cómo implementar una clase en Python. Para esto, crearemos primero el constructor de dicha clase y luego definiremos métodos que le permitan a nuestros Pokémon efectuar operaciones.

- a) Para el constructor, se necesita que cada Pokémon tenga los siguientes atributos:
- Un string con su nombre.
 - Un entero con sus puntos de vida o HP.
 - Un entero que representa su número en la Pokédex.
- b) Por otra parte, todo Pokémon debe tener métodos que le permitan efectuar las siguientes operaciones:
- Atacar a cualquier Pokémon. El ataque que se produce al Pokémon objetivo es de la tercera parte, redondeada hacia arriba, del Pokémon atacante.
 - Recibir un ataque de un Pokémon enemigo.
 - Responder, en cualquier momento, a la pregunta de si sigue vivo o no.

Pregunta 2

En esta pregunta resolveremos varios ejercicios cortos y con dificultad relativamente sencilla que nos muestran cómo trabajar con los arreglos de NumPy.

- a) Crear un arreglo de **enteros** de tamaño 11, que sólo contenga ceros. Modificar el primer y el último ítem del arreglo para que contengan ambos un 5.
- b) Crear una matriz de 3x3 que contenga los números del 5 al 13 e imprimirla.

- c) Crear un arreglo que **contenga los números enteros** del 32 al 68. Imprimir el arreglo y sus dimensiones. Luego invertirlo utilizando una matriz con unos **enteros** como matriz auxiliar y volver a imprimirlo. Imprimir también sus dimensiones. Finalmente, convertir el arreglo en una matriz de 6x6 e imprimirla. Imprimir las dimensiones de la matriz.
- d) Crear un arreglo de Strings de tamaño 5 que soporte cadenas de letras de tamaño máximo 20. Rellenar el primer y el último elemento con cualquier palabra de máximo ese tamaño e imprimir el arreglo.
- e) Crear un arreglo de 15 **decimales** y llenarlos de tal forma que el primer elemento contenga un 3.30 y cada elemento siguiente sea mayor que el anterior en 0.3.
- f) Crear un arreglo que contenga tres Pokémon e imprima sus nombres.

Pregunta 3

En esta pregunta recordaremos lo que es la *recursión*, contenido que se vio en el curso de *Introducción a la Programación*. Imprimiremos todos los cursos que son requisitos para poder tomar *Mecánica*, curso que es de tercer semestre.

Primero, modelaremos la clase Curso. Un curso debe tener un String con su nombre, un String con su código de curso, un entero con su cantidad de UDs, un entero con el semestre al que corresponde y un arreglo que contenga los cursos que son requisitos para poder cursarlo. El único método que dicha clase debe tener es uno que imprima toda la información del curso en pantalla.

Luego crearemos instancias de los cursos con los ramos que son necesarios para cursar *FI2001 Mecánica*. Los requisitos de Mecánica son:

- *FI1002 Sistemas Newtonianos*
- *MA1002 Cálculo Diferencial e Integral*
- *MA1102 Álgebra Lineal*

Para poder cursar Sistemas Newtonianos se requiere haber cursado:

- *CC1001 Herramientas Computacionales*
- *FI1001 Introducción a la Física.*

Para poder cursar Álgebra Lineal y Cálculo Diferencial e Integral es necesario haber cursado:

- *MA1001 Introducción al Cálculo y*
- *MA1101 Introducción al Álgebra.*

Una vez creadas todas las instancias, imprimiremos los cursos que son requisitos para haber cursado Mecánica de forma recursiva, utilizando el método que creamos para ello. Asumiremos que los requisitos simultáneos son requisitos estrictos y no nos importará imprimir más de una vez un curso.