# Problema 2 Extra.

**Autor: Gabriel Andres Castillo Rosales**
**RUT: 19.679.401-8**
**Docente: Matias Pastene Orellana**
**Curso: ING1312**
**Seccion: 3**

**Descripcion:** Modulo extra para P2 de tarea 1.
Simulacion de la escalera siendo inundada y el escape de los gatos usando POO, instanciacion, iterables y algunas librerias adicionales. Se ignora la condicion "solo puede haber un gato por escalon" para simplificar un poco los condicionales, pero dado el codigo usado, esta debería cumplirse de todas formas para la configuracion básica de la simulacion.
No está testeado para las configs isRandomized=True, isFair=True.
Se hace entrega del modulo listo para ejecutar desde el depurador.

Notas Adicionales:

- Se usa la convencion camelCase para nombrar las variables y funciones.
- Se usa la convencion PascalCase para nombrar objetos.
- Se usa declaracion de variables por compresion cuando se considera conveniente.
- El codigo tiene una mezcla de ingles y español, ya que aprendí a usar Python de forma autodidacta. (Youtube, StackExchange, PythonDocs y mucho Google)
- Las líneas marcadas con "#Sanity Check" cumplen la funcion de prevenir errores fortuitos.
- Todas las funciones y objetos tienen comentarios con sus respectivas descripciones.
- Algunas líneas tienen comentarios con informacion adicional relevante.

Dependencias:

- time
- random
- sys

# Funciones

- `getValidInput`(*prompt, dtype='str', gate=0*) :
  Obtiene un input valido dependiendo del tipo de dato y el valor del mismo.

  –

**Uso:**

```
user_var = getValidInput(user_prompt, dtype=user_dtype,
gate=user_gate)
```

–

**Args:**

* prompt (*str*): Texto que se mostrara en la consola para pedir un input
* dtype (*str*, optional): Tipo de dato deseado. Por defecto es **'str'**.
* gate (*int*, optional): Define un umbral inferior para los valores numericos.

–

**Raises:**

* ValueError: Valor inesperado
* TypeError: Tipo de dato inesperado

Returns:
**dtype**: Devuelve el input del usuario en el tipo de dato deseado y con las restricciones de valor deseadas

- `main()` :
Funcion de ejecucion primaria. Al llamarla se ejecuta el modulo.

# Objetos

- `Cat`(*name, pos, status='active'*) :

  Objeto que representa a un gato.

  – Uso: `var = Cat(user_name, user_pos, status='user_status')`
  Inicializa una instancia del objeto Cat.

  –

  **Args:**

  * name (*string*): Nombre del gato.
  * pos (*int*): Posicion en la escalera del gato.
  * status (*str*, optional): Estado del gato. Puede ser ('active', 'sunk', 'safe'). Por defecto es **'active'**.

  –

### Metodos:

- `setPosition`(*self, new*):
  - **Uso:** self.setPosition(new) : Cambia el valor de Cat.pos
  - **Args:** - new (*int*): Nueva posicion del gato en la escalera. Debe ser un entero positivo no nulo.
- `setStatus`(*self, new*)
  - **Uso:** self.setStatus(*new*) : Cambia el valor de Cat.status
  - **Args:**
    - new (*string*): Nuevo estado del gato. Puede ser ('active', 'sunk', 'safe')

---

- `Stair`(*stepsTotal, waterLevel=0, cats=None*) :

  Objeto que define la escalera y maneja la ejecucion de la simulacion. Solo definir 1 instancia.

  - Uso: `Stair`(*user_stepsTotal, waterLevel=user_waterLevel, cats=user_cats*)
    Inicializa una instancia del objeto Stair.
  - **Args:**
    - stepsTotal (*int*): Cantidad de escalones total.
    - waterLevel (*int*, optional): Posicion Inicial de la marea. Por defecto es **0**.
    - cats (*list(`<class> Cat`)*, optional): Lista que contiene instancias del objeto Cat.Por defecto es **None**.
  - **Metodos:**
    - `parseCats`(*self*) : Separa y cuenta las instancias de Cat en self.cats segun su estado.

      **Returns:** - array: [catGroups, counters] - Donde "catGroups" es una lista de 3 listas con instancias de Cat separadas por estado y "counters" la lista de los largos de estas.
    - `printStairState`(*self, catGroups, counts, isFinal=False*) : Imprime el estado actual de la simulacion.

      **Args:**
      - catGroups (*array-like*): Iterable que contiene las instancias de todos los gatos en la simulacion.
      - counts (*array-like*): Iterable que contiene la cantidad de elementos de cada entrada de catGroups.
      - isFinal (*bool*, optional): Determina si se debe imprimir la pantalla final. Por defecto es **False**.

* `update`(*self, moveCats=1, moveWater=1, dt=10, isFair=True, isRandomized=False*) : Actualiza la simulacion.

  **Args:**
  - moveCats (*int*, optional): Cantidad de escalones que recorren los gatos. Por defecto es **1**.
  - moveWater (*int*, optional): Cantidad de escalones que recorre la marea. Por defecto es **1**.
  - dt (*int*, optional): Intervalo de tiempo elapsado. Por defecto es **10**.
  - isFair (*bool*, optional): Condicional de movimiento justo. Por defecto es **True**.
  - isRandomized (*bool*, optional): Condicional de movimiento aleatorio. Por defecto es **False**.

  Returns:
  - bool: Determina si la simulacion continua. Para **False**, se detiene la simulacion