

Tarea Nro. 1 SB. - DECORADORES, COMPRESIÓN DE LISTAS

- Nombre y Apellido: Cristhia Michael Jimenez Jimenez
- Fecha: 21/dic/2020

Decoradores

Problema 1

Crear los decoradores para la función impresora de tal forma que se presente signos de * y % en medio del mensaje.

```
@strella
@porcentaje
def impresora(msg):
    print(msg)

*****

%%%%%%%%%%%%%%
Hello
%%%%%%%%%%%%%%
*****
```

Modifique el orden como se llaman los decoradores y explique lo que está sucediendo.

En Python, podemos pasar un número variable de argumentos a una función usando símbolos especiales

- args (argumentos sin palabras clave) ** kwargs (argumentos de palabras clave)

```
In [3]: def porcentaje(function):
        def funcion_interior(*args, **kwargs):
            print("%%%%%%%%%%%%%%")
            function(*args, **kwargs)
            print("%%%%%%%%%%%%%%")

        return funcion_interior

def stella(function):
    def funcion_interior(*args, **kwargs):
        print("*****")
        function(*args, **kwargs)
        print("*****")

    return funcion_interior

@strella
@porcentaje
def impresora(msg):
    print(msg)
```

```
def main():
    impresora("hello")

if __name__ == "__main__":
    main()

*****
%%%%%%%%%%%%%%
hello
%%%%%%%%%%%%%%
*****
```

In []:

Problema 2

Genere el decorador division_inteligente para la función divide de tal forma que trabaje como los resultados que se presentan a continuación:

```
@division_inteligente
def divide(a,b):
    return a/b

>>> divide(2,5)
Voy a dividir 2 y 5
0.4
>>> divide(2,0)
Voy a dividir 2 y 0
Pops! no puedo dividir
```

Lo encerramos en un bloque try catch para poder manejar la divicion entre cero

```
In [11]: def division_inteligente (function):
def funcion_interior(a,b):
    print(f"voy a dividir {a} y {b}")
    try:
        return function(a, b)
    except ZeroDivisionError as e:
        print(f"Pops! no puedo dividir para ({b})")
    return funcion_interior

@division_inteligente
def divide(a,b):
    return a/b

def main():
    print(divide(2, 0))

if __name__ == "__main__":
    main()
```

```
voy a dividir 2 y 0
Pops! no puedo dividir para (0)
None
```

Problema 3

Crear un decorador que permita mostrar un aviso de que una función será discontinuada en stderr cuando invoquemos por primera vez una función que ya no queremos. El objetivo es no modificar la función.

Compresión de listas

Problema 4

Genere la lista $[(1,1), (1,2), (1,3), (1,4), \dots, (1, n)]$, para cualquier n dado. Use esta lista para crear todas las fracciones $1.0 / k$, para k de 1 a n . Finalmente, use `round()` para redondear todas las fracciones a dos decimales.

Problema 5

Aproximar la función exponencial como $\sum_{k=0}^n \frac{x^k}{k!}$

Utilice compresión de listas para evaluar esta aproximación para x y n dados.

Problema 4

Escriba una lista de comprensión que resuelva la ecuación $y = x^2 + 1$. Su solución debe imprimir una lista de pares $[x, y]$; use el dominio $x \in [-5, 5]$ y el rango $y \in [0, 10]$.

Problema 5

Escriba una función llamada `mas_vocales(lista de palabras)` que tome una lista de cadenas y devuelva la cadena con más vocales en la lista. Puede suponer que las cadenas solo contienen letras minúsculas. Por ejemplo:

```
>>> mas_vocales(['Buddy', 'Oreo', 'Max', 'Toto'])
'Oreo'
>>> mas_vocales(['puppies', 'are', 'the', 'greatest'])
'puppies'
```

Utilice compresión de listas en la solución del problema.