Instituto Superior de Formación Técnica Nº 151

Carrera: Analista en Sistemas

X Año. Algoritmos y Estructuras de Datos X.

Trabajo Práctico Nº2	Unidad X2- Python
Modalidad: Semi -Presencial	**Estratégica Didáctica:** Trabajo individual.
Metodología de Desarrollo: Det. docente	**Metodología de Corrección:** Via Classroom.
Carácter de Trabajo: Obligatorio – Con Nota	**Fecha Entrega:** A confirmar por el Docente.

INTRODUCCION A PYTHON.

6. programación Funcional en Python

6.1 Que entiende por programación Funcional
6.2 Que herramientas provee Python para la programación Funcional
6.3 Para que sirve la función Map, ¿que facilita o permite?
6.4 ¿Que utilidad tiene la Función Filter?, como se realizaría en estructurado?
6.5 ¿Que permite la Función Reduce en Python?
6.6 Explicar el Siguiente Código de Ejemplo de programación Funcional

```
from functools import reduce
personas = [
    {'Nombre': 'Alicia', 'Edad': 22, 'Sexo': 'F'},
    {'Nombre': 'Bob', 'Edad': 25, 'Sexo': 'M'},
    {'Nombre': 'Charlie', 'Edad': 33, 'Sexo': 'M'},
    {'Nombre': 'Diana', 'Edad': 15, 'Sexo': 'F'},
    {'Nombre': 'Esteban', 'Edad': 30, 'Sexo': 'M'},
    {'Nombre': 'Federico', 'Edad': 44, 'Sexo': 'M'},
]
hombres = list(filter(lambda x: x['Sexo'] == 'M', personas))
suma_edades = reduce(lambda suma, p: suma + p['Edad'], hombres, 0)
media_edad = suma_edades/(len(hombres))
print(media_edad) # 33.0
```

6.7 De ser Posible comparar con C++

Marco Practico: https://pywombat.com/articles/programacion-funcional-python

6.1 Obtener el cuadrado de todos los elementos en la lista.
6.2 Obtengamos la cantidad de elementos mayores a 5 en una tupla.
6.3 Obtengamos la cantidad de elementos mayores a 5 en una tupla, usando Reduce

7. Excepciones en Python

Marco Teorico

7.1 Que es una Excepción, que significa “Lanza” una Excepción y “Capturarla”
7.2 Complementar con ejemplos de TypeError, KeyError, IndexError, NameError, RuntimeError, ZeroDivisionError
7.3 Que utilidad tiene la palabra Raise.

Marco Practico
https://www.freecodecamp.org/espanol/news/sentencias-try-y-except-de-python-como-menejar-excepciones-en-python/

7.1 Crear una Función que divia hasta cero, ej: dividir(27,0), verificar: ZeroDivisionError:
7.2 llamar a la función mas_10() con cualquier número, además: add_10("cinco") verificar TypeError:
7.3 Crear una Lista e Iterar mas allá del limite del Index, verídica: IndexError:
7.3 Crear un Diciconario en Python y buscar una clave Inexistente, verificar keyError:

8. Archivos en Python

Marco teórico

8.1 Explicar las Funciones Open() y Close() de Python.
8.2 Explicar los distintos “Modos” de Apertura.
8.3 Definir y dar Ejemplos de seek(), readline() y readlines()

Marco Practico **https://byte-mind.net/curso-python-trabajando-ficheros/#Problemas-propuestos**

8.1 Crear un programa que abra un fichero en modo lectura y escritura, si no existe lo creará, y añadir la frase “Estoy aprendiendo Python”
8.2 Crear un programa que abra el fichero editado anteriormente y muestre el estado del fichero, el modo en el que fue abierto, el nombre y la codificación de caracteres del mismo.
8.3 Realizar un programa que realice los ejercicios 1 y 2 utilizando la estructura with.

9. Módulos y Paquetes.

Marco teórico

9.1 Definir y relacionar los Conceptos de Módulos y Paquetes.
9.2 Que función tiene sys.path()
9.3 Por qué usaría “AS” para cambiar o poner alias a paquetes en Python.
9.4 ¿Como maneja Python las Excepciones al Importar Módulos?
9.5 Como maneja Python la inclusión Multiple de Módulos

Marco Practico https://pythonpanama.github.io/pythonpractico/9/

9.1: Hacer un paquete simple
9.2: Crear un directorio de aplicaciones
9.3: Scripts de nivel superior

10. Testing en Python.

10.1 ¿Que entiende x Testing?
10.2¿Que diferencia tenemos entre Test Manual y Automático?
10.3¿Que es un Assert en Test Automaticos?
10.4 Que es unittest, como se Implementa en Python?
10.5 Explicar el Siguiente Codigo:

```
# funciones.py

    def calcula_media(*args):
        return(sum(*args)/len(*args))

# tests.py

from funciones import calcula_media
import unittest
class TestCalculaMedia(unittest.TestCase):
  def test_1(self):
    resultado = calcula_media([10, 10, 10])
    self.assertEqual(resultado, 10)
  def test_2(self):
    resultado = calcula_media([5, 3, 4])
    self.assertEqual(resultado, 4)

if __name__ == '__main__':
  unittest.main()
```

10.6 Realizar una Breve descripcion de las siguientes funciones de UnitTes:

.assertEqual(a, b): Verifica la igualdad de ambos valores.
.assertTrue(x): Verifica que el valor es True.
.assertFalse(x): Verifica que el valor es False.
.assertIs(a, b): Verifica que ambas variables son la misma (ver operador is).

.assertIsNone(x): Verifica que el valor es None.
.assertIn(a, b): Verifica que a pertenece al iterable b (ver operador in).
.assertIsInstance(a, b): Verifica que a es una instancia de b
.assertRaises(x): Verifica que se lanza una excepción.

10.7 Que permite Usando setUp y tearDown en UnitTest?

Marco Practico

Avanzado Objetos unittest: https://pythonpanama.github.io/pythonpractico/8/

10.1 Explicar el Siguiente Código:

```
import unittest
class TestEjemplos(unittest.TestCase):
    def setUp(self):
        print("Entra setUp")
    def tearDown(self):
        print("Entra tearDown")
    def test_1(self):
        print("Test: test_1")
    def test_2(self):
        print("Test: test_2")

python -m unittest -v tests
```

La Cátedra