Módulo 1 - Capitulo 3

Unidad 17

Quiz 1

Defina una función llamada `my_greet` que imprima `"Bienvenido"` y llame esta función para que imprima el saludo 2 veces.

```
def my_greet(repeat):
    for _ in range(repeat):
        print("Bienvenido")

my_greet(2)
```

```
Bienvenido
Bienvenido
```

Quiz 2

Implemente la función `max2(m, n)` la cual tome dos parámetros llamados `m` y `n` que regrese el número mayor de estos dos valores. Por otro lado, implemente la función `min2(m, n)` que tome dos parámetros llamados `m` y `n` y regrese el número menos de estos dos valores. Luego, asigne el valor de 100 y 200 como argumentos para las dos funciones e imprima su resultado.

```
def max2(m, n):
    return m if m > n else n

def min2(m, n):
    return m if m < n else n

print('Número mayor:', max2(10, 20))
print('Número menor:', min2(10, 20))
```

```
Número mayor: 20
Número menor: 10
```

Quiz 3

Queremos cambiar el valor de la distancia de millas a kilómetros (Las millas son una unidad de medida que se usa principalmente en Estados Unidos, mientras que los kilómetros son la unidad estándar internacional). Para

ello implemente la función `mile2km(ml)` que toma valores de las millas y las convierte en kilómetros. Luego llame esta función para que imprima la conversión de las millas 1 a 5 kilómetros a través de la sentencia `for`. (Se define 1 milla con 1.61 km)

```
def mile2km(ml):
    for i in range(1, ml + 1):
        print(f'Mile {i} = Kilometer {i * 1.61}')

mile2km(5)
```

```
Mile 1 = Kilometer 1.61
Mile 2 = Kilometer 3.22
Mile 3 = Kilometer 4.83
Mile 4 = Kilometer 6.44
Mile 5 = Kilometer 8.05
```

Quiz 4

Implemente la función `cel2fah(cel)` que toma la temperatura en grados Celsius y que retorna la temperatura en grados Fahrenheit. Luego llame la función, a través de la sentencia `for`, que imprima las conversiones de temperatura desde 10 hasta 50 grados Celsius en pasos de 10 unidades. (Recuerde la fórmula de conversion `Fah = Cel * (9/5) + 32`)

```
def formula(cel):
    return cel * (9/5) + 32

def cel2fah(cel):
    for i in range(10, cel + 10, 10):
        print(f'Cel { i } = Fah { formula(i)}')

cel2fah(50)
```

```
Cel 10 = Fah 50.0
Cel 20 = Fah 68.0
Cel 30 = Fah 86.0
Cel 40 = Fah 104.0
Cel 50 = Fah 122.0
```

Quiz 5

Permita que un usuario ingrese 3 enteros a, b y c e imprima el valor promedio, el máximo y el mínimo de estos 3. Para ello, defina y llame las funciones `mean3(a, b, c)`, `max3(a, b, c)`, `min3(a, b, c)`

```
def mean3(a, b, c):
    return (a + b + c)/3

def max3(x):
    return max(x)

def min3(x):
    return min(x)

x = [int(x) for x in input('Enter three numbers: ').split()]

a = x[0]
b = x[1]
c = x[2]

print(f'The average value of {x} is {mean3(a, b, c)}')
print(f'The maximum value of {x} is {max3(x)}')
print(f'The minimum value of {x} is {min3(x)}')
```

```
The average value of [9, 2, 6] is 5.666666666666667
The maximum value of [9, 2, 6] is 9
The minimum value of [9, 2, 6] is 2
```

Unidad 18

Quiz 1

Tome un número `n` como entrada y encuentre su suma desde `1` hasta `n`. Escriba su función usando funciones recursivas.

```
n = 10

def sum2n(n):
    if n == 1:
        return 1
    return n + sum2n(n-1)

sum2n(n)
```

```
55
```

Quiz 2

Python tiene el operador ** el cual indica una potencia. Sin embargo, tomemos a x y n como entradas sin usar ese operador y use una función recursiva que retorne x a la potencia de n. Luego, calcule 2^{10} poniendo x como 2 y n como 10.

```
def powWithMulti(x, n):
    if n == 1:
        return x
    return x * powWithMulti(x, n - 1)


powWithMulti(2, 10)
```

```
1024
```

Quiz 3

El número natural e, también llamado el número de Eular o la constante de Napier, es un número irracional usado como la base de los logaritmos naturales y es definido con la siguiente formula:

$$ e = 1 + \frac{1}{1!} + \frac{1}{2!} + \frac{1}{3!} + ... + \frac{1}{n!} $$

donde el valor de Euler depende de la cantidad de términos n que se establezcan.

Desarrolle una función recursiva que permita encontrar el valor de `euler(20)` con 20 términos para encontrar los primeros 5 decimales.

Salida esperada para $n = 20$, es `euler(20) = 2.71828`

```
dic = {0:0, 1:1}

def factorial(n):
    if n in dic:
        return dic[n]
    dic[n] = n * factorial(n - 1)
    return dic[n]

def euler(n):
    if n == 0:
        return 1
    return 1 / factorial(n) + euler(n - 1)

print(f'euler(20) = {round(euler(20), 5)}')
```

```
euler(20) = 2.71828
```

Unidad 19

Quiz 1

Hay una lista de elementos enteros llamada `n_list = [1, 2, 3, 4, 5, 6, 7, 8, 9, 10]`. Retorne una lista llamada `even_list` la cual solo contiene los elementos pares de `n_list` usando las funciones `filter` y `lambda`.

Para este ejemplo, la respuesta esperada es: `even_list = [2, 4, 6, 8, 10]`

```
n_list = [1, 2, 3, 4, 5, 6, 7, 8, 9, 10]

even_list = []

for i in filter(lambda x: x % 2 == 0, n_list):
    even_list.append(i)

print(f'even_list = {even_list}')
```

```
even_list = [2, 4, 6, 8, 10]
```

Quiz 2

Hay una lista de elementos enteros llamada `n_list = [1, 2, 3, 4, 5, 6, 7, 8, 9, 10]`. Retorne una lista llamada `event_list` la cual solo contiene los elementos pares de `n_list` usando las funciones `filter` y `lambda`. Sin embargo, en este ejercicio no use el ciclo `for` y en vez de ello use la función `list`.

```
n_list = [1, 2, 3, 4, 5, 6, 7, 8, 9, 10]

even_list = list(filter(lambda x: x % 2 == 0, n_list))

print(f'even_list = {even_list}')
```

```
even_list = [2, 4, 6, 8, 10]
```

Quiz 3

Escriba una función `map` que convierta la variable `a_list`, que contiene el alfabeto en minúscula `['a', 'b', 'c', 'd']` y lo convierta en la variable `upper_a_list`, que contiene el alfabeto en mayúsculas `['A', 'B', 'C', 'D']`. También defina una función llamada `to_upper` que reciba las letras en minúsculas como parámetros y las retorne como letras en mayúsculas.

```
a_list = ['a', 'b', 'c', 'd']

def to_upper(letter: str) -> str:
    return letter.upper()

upper_a_list = list(map(to_upper, a_list))

print(f'upper_a_list = {upper_a_list}')
```

```
upper_a_list = ['A', 'B', 'C', 'D']
```

Quiz 4

Calcule la suma de enteros de 1 a 100 usando la función reduce y la expresión lambda dentro de ella. Use como entrada `range(1, 101)`.

Para este ejemplo, la respuesta esperada es `La suma de 1 a 100 es: 5050`

```
from functools import reduce

arr = range(1, 101)

sum = reduce(lambda x, y: x + y, arr)

print(f'La suma de 1 a 100 es: {sum}')
```

```
La suma de 1 a 100 es: 5050
```

Quiz 5

Tienes disponible la información de los puntajes de los exámenes de los estudiantes en inglés, matemáticas y ciencias que pueden ser expresados como una lista `[100, 90, 95]`. Si hay dos estuidantes, los puntajes se representarían como `[100, 90, 95, 90, 85, 93]`. Ahora, si un estudiante no realizó algún examen en particular se marca con el puntaje de 0. Por lo tanto, se requiere un código que imprima:

- De cuantos estudiantes se tienen los puntajes
- El número de estudiantes con puntajes válidos (Es decir, que no tienen 0 en alguna materia)
- El puntaje de los estudiantes con calificaciones válidas.

Ejemplo de una entrada:

```
scores = [100, 90, 95, 90, 80, 70, 0, 80, 90, 90, 0, 90, 100, 75, 20, 30, 50, 90]
```

Ejemplo de salida:

```
scores = [100, 90, 95, 90, 80, 70, 0, 80, 90, 90, 0, 90, 100, 75, 20, 30, 50, 90]

El número total de estudiantes es 6

El número total de estudiantes válidos es 4

[[100, 90, 95], [90, 80, 70], [100, 75, 20], [30, 50, 90]]
```

```
scores = [100, 90, 95, 90, 80, 70, 0, 80,
            90, 90, 0, 90, 100, 75, 20, 30, 50, 90]

# Contar el número total de estudiantes
n_students = len(scores) // 3


# Contar el número de estudiantes con puntajes válidos
# y crear una nueva lista de puntajes válidos
valid_scores = []
n_student_valid = 0

for i in range(n_students):
    start = i * 3
    end = start + 3
    scs = scores[start : end]
    if 0 not in scs:
        valid_scores.append(scs)
        n_student_valid += 1


print(f'El número total de estudiantes es {n_students}')
print(f"El número total de estudiantes válidos es {n_student_valid}")
print(valid_scores)
```

```
El número total de estudiantes es 6
El número total de estudiantes válidos es 4
[[100, 90, 95], [90, 80, 70], [100, 75, 20], [30, 50, 90]]
```

```
scores = [100, 90, 95, 90, 80, 70, 0, 80,
            90, 90, 0, 90, 100, 75, 20, 30, 50, 90]

# Contar el número de estudiantes
n_students = len(scores) // 3
```

```
# Filtrar y trasformar los puntajes de los estudiantes,
# tomando en cuenta solo los puntajes válidos
valid_scores = list(
    filter(
        lambda x: 0 not in x,
        map(
            lambda i: scores[i : i+3],
            range(0, len(scores), 3)
        )
    )
)

# Contar el número de estudiantes válidos
n_student_valid = len(valid_scores)


print(f'El número total de estudiantes es {n_students}')
print(f"El número total de estudiantes válidos es {n_student_valid}")
print(valid_scores)
```

```
El número total de estudiantes es 6
El número total de estudiantes válidos es 4
[[100, 90, 95], [90, 80, 70], [100, 75, 20], [30, 50, 90]]
```

```
scores = [100, 90, 95, 90, 80, 70, 0, 80,
            90, 90, 0, 90, 100, 75, 20, 30, 50, 90]

# Generar una lista de subgrupos de 3 puntajes consecutivos
sub_groups = [scores[i: i+3] for i in range(0, len(scores), 3)]

# Filtrar los subgrupos que no contienen un puntaje de 0
valid_scores = [sub_group for sub_group in sub_groups if 0 not in sub_group]

# Obtener el número total de estudiantes y el número de estudiantes válidos
n_students = len(scores) // 3
n_student_valid = len(valid_scores)


print(f'El número total de estudiantes es {n_students}')
print(f"El número total de estudiantes válidos es {n_student_valid}")
print(valid_scores)
```

```
El número total de estudiantes es 6
El número total de estudiantes válidos es 4
```

```
[[100, 90, 95], [90, 80, 70], [100, 75, 20], [30, 50, 90]]
```

Unidad 20

Quiz 1

Cree una función anidada llamando a esta función `greetings` y otra función llamada `say_hi` dentro de esa función. Llame la función `say_hi` dentro de la función `greetings` y luego, llame la función `greetings` e imprima `hello`.

Use la siguiente definición para la función `say_hi`:

```
def say_hi():
    print('hello')
```

```
def greetings():
    def say_hi():
        print('hello')
    return say_hi

greetings()()
```

```
hello
```

Quiz 2

Escriba la siguiente función `calc` y asigne `calc` a la variable `num`. Luego, ejecute `num(3)`, el resultado debe ser 14.

```
def calc():
    a = 3
    b = 5
    def mul_add(x):
        return a * x + b
    return mul_add
```

```
def calc():
    a = 3
    b = 5

    def mul_add(x):
        return a * x + b
```

```
    return mul_add

num = calc()
num(3)
```

```
14
```

Quiz 3

Reconstruya la función interna `mul_add` de la función anidada `calc` del problema anterior usando una expresión lambda y obtenga el mismo resultado.

```
def calc():
    a = 3
    b = 5

    return lambda x: a * x + b

num = calc()
num(3)
```

```
14
```

Quiz 4

Extraiga la lista resultante de la lista `lst` que tiene valores de 1 a 100. La lista resultante tiene los elementos de `lst` que son divisibles por 5 o 7. Para ello declare la función `func1(a)` y anide las funciones `func2` y `func3`. Luego llame las dos funciones desde la función `func1` e imprima los números que son divisibles por 5 o 7. Finalmente, alinee los valores resultantes usando la función pre-construida `sorted()`.

Funciones a anidar:

```
def func2():
    result1 = []
    for i in a:
        if i % 5 == 0:
            result1.append(i)
    return result1

def func3():
    result2 = []
    for i in a:
        if i % 7 == 0:
```

```
            result2.append(i)
    return result2
```

```
lst = list(range(1, 101))

def func1(a):
    result = []

    def func2():
        result1 = []
        for i in a:
            if i % 5 == 0:
                result1.append(i)
        return result1

    def func3():
        result2 = []
        for i in a:
            if i % 7 == 0:
                result2.append(i)
        return result2

    result.extend(func2())
    result.extend(func3())

    return sorted(result)

result = func1(lst)

print(f'lst: {lst}', end="\n\n")
print(f'result: {result}')
```

```
lst: [1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21,
22, 23, 24, 25, 26, 27, 28, 29, 30, 31, 32, 33, 34, 35, 36, 37, 38, 39, 40, 41,
42, 43, 44, 45, 46, 47, 48, 49, 50, 51, 52, 53, 54, 55, 56, 57, 58, 59, 60, 61,
62, 63, 64, 65, 66, 67, 68, 69, 70, 71, 72, 73, 74, 75, 76, 77, 78, 79, 80, 81,
82, 83, 84, 85, 86, 87, 88, 89, 90, 91, 92, 93, 94, 95, 96, 97, 98, 99, 100]

result: [5, 7, 10, 14, 15, 20, 21, 25, 28, 30, 35, 35, 40, 42, 45, 49, 50, 55, 56,
60, 63, 65, 70, 70, 75, 77, 80, 84, 85, 90, 91, 95, 98, 100]
```

```
lst = list(range(1, 101))

def func1(a):
    result = []
```

```
    def func2():
        return [x for x in a if x % 5 == 0]

    def func3():
        return [x for x in a if x % 7 == 0]

    result.extend(func2())
    result.extend(func3())

    return sorted(result)


result = func1(lst)

print(f'result: {result}')
```

```
result: [5, 7, 10, 14, 15, 20, 21, 25, 28, 30, 35, 35, 40, 42, 45, 49, 50, 55, 56,
60, 63, 65, 70, 70, 75, 77, 80, 84, 85, 90, 91, 95, 98, 100]
```

```
lst = range(1, 101)


def func1(a):
    result = []

    def func2():
        nonlocal result
        result = [x for x in a if x % 5 == 0 or x % 7 == 0]
        return sorted(result)

    return func2


result = func1(lst)()

print(f'result: {result}')
```

```
result: [5, 7, 10, 14, 15, 20, 21, 25, 28, 30, 35, 40, 42, 45, 49, 50, 55, 56, 60,
63, 65, 70, 75, 77, 80, 84, 85, 90, 91, 95, 98, 100]
```

Unidad 21

Quiz 1

Construya la clase `Dog` y sus objetos con sus funcionalidades descritas a continuación:

- Tiene un método llamado `def bark(self):` Este método imprime un ladrido.
- Genere una instancia llamada `my_dog` y refiera a `Dog` con un la instrucción `my_dog = Dog()`
- Imprima un ladrido con un método llamando `my_dog.bark()`

```
class Dog():
    def bark(self):
        print('woof woof')

my_dog = Dog()
my_dog.bark()
```

```
woof woof
```

Quiz 2

Defina la clase `Dog` con las funcionalidades descritas a continación y llama sus método e instancias:

- Esta clase `Dog` tiene un atributo llamado `name`.
- Tiene un método de inicialización llamado `def __ini__(self, name):`. Este método inicializa el nombre de la clase `Dog`.
- Tiene un método llamado `def bark(self)`. Este método imprime un ladrido.
- Genere una instancia `my_dog` que tiene el nombre de `Bingo` con el comando `my_dog = Dog('Bingo')`
- Imprima los siguientes ladridos con el método `my_dog.bark()`

```
class Dog():
    def __init__(self, name):
        self.name = name

    def bark(self):
        print(f'{self.name}: woof, woof')

my_dog = Dog(name='Bingo')
my_dog.bark()
```

```
Bingo: woof, woof
```

Quiz 3

Construya la clase `Student` que tiene las siguientes funcionalidades:

- Este estudiante toma quices para inglés, matemáticas, ciencias y los puntajes son dados como entradas. Genere instancias a través del uso de esta clase. Esta clase tiene los siguientes atributos y acciones.

attributes:

- `name`: nombre de estudiantes almacenado en tipo string
- `student_id`: id del estudiante como `s2020001` almacenado en enteros de 8 digitos.
- `eng_quiz`: puntaje del quiz de inglés del estudiante almacenado en formato de lista
- `math_quiz`: untaje del quiz de matemáticas del estudiante almacenado en formato de lista
- `science_quiz`: untaje del quiz de ciencias del estudiante almacenado en formato de lista

actions(methods):

- `__init__`: inicialice con el nombre del estudiante y el id.
- `__str__`: retorne el nombre del estudiante, el id, y los puntajes de los parciales como strings.
- `set_eng_quiz`: setea el quiz de inglés
- `set_math_quiz`: setea el quiz de matemáticas
- `set_science_quiz`: setea el quiz de ciencias
- `get_name`: retorna el nombre del estudiante
- `get_student_id`: retorna el id student_id
- `get_eng_quiz`: retorna el quiz de inglés
- `get_math_quiz`: retorna el quiz de matemáticas
- `get_science_quiz`: retorna el quiz de ciencias
- `get_total_score`: retorna el puntaje total de los quices
- `get_avg_score`: retorna el promedio de los quices

Respuesta esperada:

```
Enter the student's name: David Doe
Enter the student's ID: 20230217
Enter the student's English quiz score: 90
Enter the student's Mathematics quiz score: 95
Enter the student's Science quiz score: 100
Name: David Doe, ID: 20230217
English quiz score: 90, Mathematics quiz score: 95, Science quiz score: 100
Total: 285
Average: 95.0
```

```
class Student():
    def __init__(self, name: str, student_id: int, eng_quiz: int, math_quiz: int,
science_quiz: int):
        self.name = name
        self.student_id = student_id
        self.__eng_quiz = eng_quiz
        self.__math_quiz = math_quiz
        self.__science_quiz = science_quiz

    def set_eng_quiz(self, score):
```

```
        if score >= 0:
            self.__eng_quiz = score

    def get_eng_quiz(self):
        return self.__eng_quiz

    def set_math_quiz(self, score):
        if score >= 0:
            self.__math_quiz = score

    def get_math_quiz(self):
        return self.__math_quiz

    def set_science_quiz(self, score):
        if score >= 0:
            self.__science_quiz = score

    def get_science_quiz(self):
        return self.__science_quiz

    def get_total_score(self):
        return f'Total: {self.__eng_quiz + self.__math_quiz +
self.__science_quiz}'

    def get_avg_score(self):
        return f'Average: {(self.__eng_quiz + self.__math_quiz +
self.__science_quiz) / 3}'

    def __str__(self):
        return (f'Name: {self.name}, ID: {self.student_id} \nEnglish quiz score:
{self.get_eng_quiz()}, Mathematics quiz score: {self.get_math_quiz()}, Science
quiz score: {self.get_science_quiz()}')



name = input("Enter the student's name: ")
student_id = int(input("Enter the student's ID: "))
eng_quiz = int(input("Enter the student's English quiz score: "))
math_quiz = int(input("Enter the student's Mathematics quiz score: "))
science_quiz = int(input("Enter the student's Science quiz score: "))


student = Student(
    name,
    student_id,
    eng_quiz,
    math_quiz,
    science_quiz
)


print(student)
print(student.get_total_score())
print(student.get_avg_score())
```

```
Name: David Doe, ID: 20230217
English quiz score: 90, Mathematics quiz score: 95, Science quiz score: 100
Total: 285
Average: 95.0
```