Creado por:

Isabel Maniega

Clases y Funciones

Nos permiten ordenar el código, como por ejemplo código repetido, etc

Ejemplo 1: 1 Clase - 2 funciones

In [1]:
```
class Clase1:

    def funcion1():
        print("Estamos en: Clase1 - Funcion1")

    def funcion2():
        print("Estamos en: Clase1 - Funcion2")
```

In [2]:
```
# funcion1() --> no funciona, no encuentra la funcion
```

In [3]:
```
# se define primero la clase + función
Clase1.funcion1()
```

```
Estamos en: Clase1 - Funcion1
```

In [4]:
```
Clase1.funcion2()
```

```
Estamos en: Clase1 - Funcion2
```

Ejemplo 1

In [5]:
```
class Clase1:

    def imprimir(y):
        print("Estamos en: ")
        print("Clase Python - funcion imprimir, argumento y: ", y)

    def funcion_suma():
        x = 1
        z = 3
        y = x + z

        print(f"La suma de {x} más {z} es {y}")
```

In [6]:
```
# con argumento
Clase1.imprimir(6)
```

```
Estamos en:
Clase Python - funcion imprimir, argumento y:  6
```

In [7]:
```
# sin argumento
Clase1.funcion_suma()
```

```
La suma de 1 más 3 es 4
```

Ejemplo 2

In [8]:
```
class Clase1:

    def funcion1():
        print("Estamos en: Clase1 - Funcion1")

    def funcion2():
        print("Estamos en: Clase1 - Funcion2")
        print("\n")
        funcion1() # No accede a la función se necesita declarar la clase
```

In [9]:
```
Clase1.funcion1()
```

```
Estamos en: Clase1 - Funcion1
```

In [10]:
```
Clase1.funcion2()
```

```
Estamos en: Clase1 - Funcion2
```

```
---------------------------------------------------------------------------
NameError                                 Traceback (most recent call last)
<ipython-input-10-5f71db1a008f> in <module>
----> 1 Clase1.funcion2()

<ipython-input-8-4efb4f4685cc> in funcion2()
      7         print("Estamos en: Clase1 - Funcion2")
      8         print("\n")
----> 9         funcion1() # No accede a la función se necesita declarar la clase

NameError: name 'funcion1' is not defined
```

In [11]:
```
class Clase1:

    def funcion1():
        print("Estamos en: Clase1 - Funcion1")

    def funcion2():
        print("Estamos en: Clase1 - Funcion2")
        print("\n")
        Clase1.funcion1()
```

In [12]:
```
Clase1.funcion2()
```

```
Estamos en: Clase1 - Funcion2


Estamos en: Clase1 - Funcion1
```

Ejemplo 3

In [13]:
```
class Clase1:

    def imprimir(y):
        print("Estamos en: ")
        print("Clase Python - funcion imprimir, argumento y: ", y)

    def funcion_suma():
        x = 1
        z = 3
        y = x + z

        Clase1.imprimir(y)
```

In [14]:
```
Clase1.funcion_suma()
```

```
Estamos en:
Clase Python - funcion imprimir, argumento y:  4
```

In [15]:
```
Clase1.imprimir(10)
```

```
Estamos en:
Clase Python - funcion imprimir, argumento y:  10
```

2 Clases con la misma función

- Una misma función puede estar en varias clases diferentes

Dado que tenemos que llamar previamente a la clase...

entonces:

- Sólo se ejecutará la funcion dentro de la clase

In [16]:
```
class Clase1:

    def funcion1():
        print("Estamos en: Clase1 - Funcion1")

class Clase2:

    def funcion1():
        print("Estamos en: Clase2 - Funcion1")
```

In [17]:
```
Clase1.funcion1()
```

```
Estamos en: Clase1 - Funcion1
```

In [18]:
```
Clase2.funcion1()
```

```
Estamos en: Clase2 - Funcion1
```

Ejemplo

In [19]:
```
class Frutas:
    def suma(x, y):
        suma = x + y
        print("la suma de frutas que tenemos es: ", suma)

class Verduras:
    def suma(x, y):
        suma = x + y
        print("la suma de verduras que tenemos es: ", suma)
```

In [20]:
```
Frutas.suma(3, 2)
```

```
la suma de frutas que tenemos es:  5
```

In [21]:
```
Verduras.suma(6, 2)
```

```
la suma de verduras que tenemos es:  8
```

1 clase que recibe otra clase

In [22]:
```
class Clase1:
    def funcion1():
        print("Estamos en: Clase1 - Funcion1")

class Clase2(Clase1):
    def funcion2():
        print("Estamos en: Clase2 - Funcion2")

    # def funcion1():
        # print("Estamos en: Clase1 - Funcion1")
```

In [23]:
```
Clase1.funcion1()
```

```
Estamos en: Clase1 - Funcion1
```

In [24]:
```
Clase2.funcion2()
```

```
Estamos en: Clase2 - Funcion2
```

In [25]:
```
# heredado con la clase --> HERENCIA
Clase2.funcion1()
```

```
Estamos en: Clase1 - Funcion1
```

Ejemplo práctico

In [26]:
```
class Frutas:
    def manzanas():
        suma_manzanas = 6
        return suma_manzanas

class Alimentos(Frutas):
    def peras():
        suma_peras = 7
        return suma_peras
    #def manzanas():
    #    suma_manzanas = 6
    #    return suma_manzanas
```

In [27]:
```
Frutas.manzanas()
```

Out[27]:
```
6
```

In [28]:
```
Alimentos.peras()
```

Out[28]:
```
7
```

In [29]:
```
Alimentos.manzanas()
```

Out[29]:
```
6
```

También podemos "llamar" variables

In [30]:
```
class Clase1:
    x = 5

class Clase2:
    def funcion():
        print("Estamos en la Clase2")
        print("\n")
        print("Ahora llamamos a la variable de la Clase1")
        print("Valor: ", Clase1.x)
        # print("Valor: ", x) # Necesario añadir la clase, sino error de variable No definida
```

In [31]:
```
Clase2.funcion()
```

```
Estamos en la Clase2


Ahora llamamos a la variable de la Clase1
Valor:  5
```

In [32]:
```
class Clase1:
    x = 5

class Clase2:
    def funcion():
        x = 6
        print("Estamos en la Clase2")
        print("\n")
        print("Ahora llamamos a la variable de la Clase1")
        print("Valor: ", Clase1.x)
        print("Valor: ", x) # Necesario añadir la clase, sino error de variable No definida
```

In [33]:
```
Clase2.funcion()
```

```
Estamos en la Clase2


Ahora llamamos a la variable de la Clase1
Valor:  5
Valor:  6
```

In [34]:
```
class Clase1:
    x = 6
    def funcion1(x):
        return x
```

In [35]:
```
Clase1.funcion1(Clase1.x)
```

Out[35]:
```
6
```

In [36]:
```
class Clase1:
    y = 6
    def funcion1(y):
        return y
```

In [37]:
```
Clase1.funcion1(y)
```

```
---------------------------------------------------------------------------
NameError                                 Traceback (most recent call last)
<ipython-input-37-452df15fe8ae> in <module>
----> 1 Clase1.funcion1(y)

NameError: name 'y' is not defined
```

In [38]:
```
x = 6
def funcion1(x):
    return x
```

In [39]:
```
funcion1(x)
```

Out[39]:
```
6
```

Creado por:

Isabel Maniega