Clase 21-10-2024

LINK: https://ellibrodepython.com/paso-por-valor-y-referencia

https://www.programiz.com/python-programming/online-compiler/

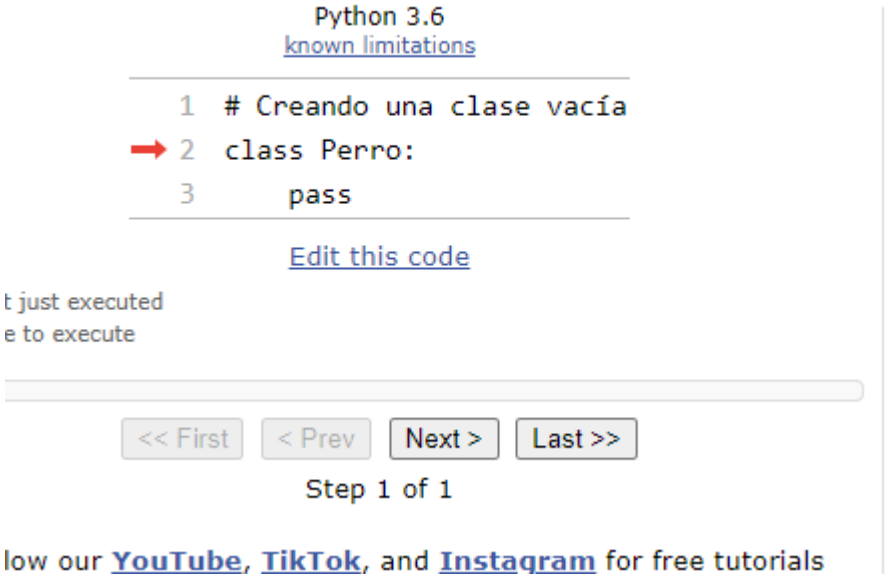
https://pythontutor.com/visualize.html#mode=edit

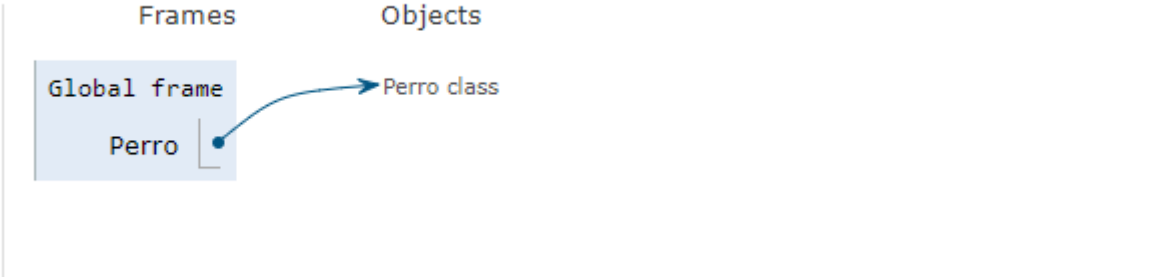
- Append es agregar

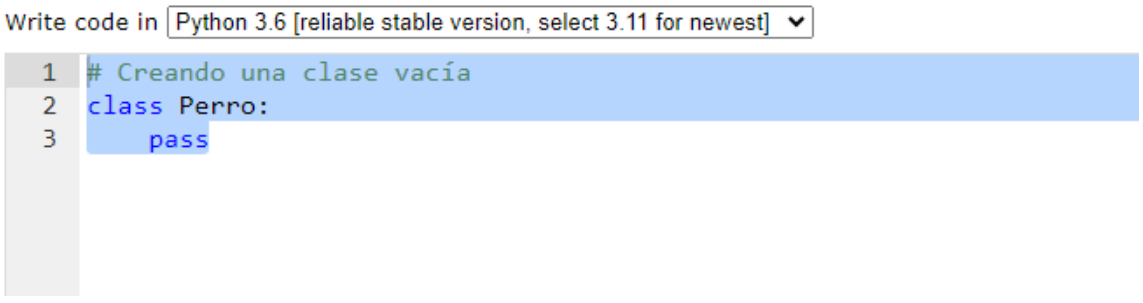
```
x = [10, 20, 30]
def funcion(entrada):
    entrada = []

funcion(x)
print(x)
# [10, 20, 30]
```

-
- If es si
- El programa se divide en cajas y en la página de pyton tutor lo refleja gráficamente.
- Luego de crear una clase, se comienza a dibujar
- Hay dos clases instancias y clase (es común para todos los objetos)
- Es importante la programación orientada a objeto.
- Init es para llamar y luego se comienza a clasificar y se nombran los atributos-

- Python tutor







self.en_marcha

main.py Share Run Output Clear

```python
# Creando una clase vacía
class Perro:
    pass
mi_perro = Perro()

class Perro:
    # El método __init__ es llamado al crear el objeto
    def __init__(self, nombre, raza):
        print(f"Creando perro {nombre}, {raza}")

        # Atributos de instancia
        self.nombre = nombre
        self.raza = raza

mi_perro = Perro("Toby", "Bulldog")
print(type(mi_perro))
# Creando perro Toby, Bulldog
# <class '__main__.Perro'>

print(mi_perro.nombre) # Toby
print(mi_perro.raza)   # Bulldog

class Perro:
    # Atributo de clase
    especie = 'mamífero'

    # El método __init__ es llamado al crear el objeto
    def __init__(self, nombre, raza):
        print(f"Creando perro {nombre}, {raza}")

        # Atributos de instancia
        self.nombre = nombre
        self.raza = raza
```

```
Creando perro Toby, Bulldog
<class '__main__.Perro'>
Toby
Bulldog

=== Code Execution Successful ===
```

- Self es porque la lista puede ser desordenada, es para tipear algo.

```python
#Definir la clase coche
class coche:
    #El metodo_init_ es el constructor, se llama al crear un objeto de la
    def _init_(self,marca,modelo,color):
        self.marca = marca #atributo de marca
        self.modelo = modelos #atributo modelo
        self.color = color #atributo color
        self.en_marcha = False #atributo para saber si el coche esta en ma

    #Metodo para arrancar el coche
    def arrancar(self):
        if not self.en_marcha:
            self.en_marcha = true
            print(f"El coche {self.marca} {self.modelo} ha arrancado.")

    #Metodo para detener el coche
    def detener(self):
        if not self.en_marcha:
            self.en_marcha = false
            print(f"El coche {self.marca} {self.modelo} se ha detenido.")
        else
            print(f"El coche {self.marca} {self.modelo} ya esta detenido."

    #Metodo para mostrar el estado del coche
    def mostrar_estado(self):
            print(f"Coche {self.marca} {self.modelo} color:{self.color}, e

    #Crear objetos (instancias) de la clase coche

    mi_coche = coche ("Toyota", "Corolla", "Rojo")
    tu_coche = coche ("Honda", "Civic", "Azul")

    #Mostrar el estado de los coches después de arrancar

    mi_coche.mostrar_estado()
    tu_coche.mostrar_estado()

    #Detener los coches

    mi_coche.detener()
    tu_coche.detener()

    #Mostrar el estado final de los coches

    mi_coche.mostrar_estado()
    tu_coche.mostrar_estado()
```

#Definir la clase coche
class coche:

```
  #El metodo_init_ es el constructor, se llama al crear un objeto de la clase
def _init_(self,marca,modelo,color):
    self.marca = marca #atributo de marca
    self.modelo = modelos #atributo modelo
    self.color = color #atributo color
    self.en_marcha = False #atributo para saber si el coche esta en marcha

  #Metodo para arrancar el coche
def arrancar(self):
    if not self.en_marcha:
      self.en_marcha = true
      print(f"El coche {self.marca} {self.modelo} ha arrancado.")

  #Metodo para detener el coche
def detener(self):
    if not self.en_marcha:
      self.en_marcha = false
      print(f"El coche {self.marca} {self.modelo} se ha detenido.")
    else:
      print(f"El coche {self.marca} {self.modelo} ya esta detenido.")

  #Metodo para mostrar el estado del coche
def mostrar_estado(self):
      print(f"Coche {self.marca} {self.modelo} color:{self.color},
estado{estado}")

  #Crear objetos (instancias) de la clase coche

  mi_coche = coche ("Toyota", "Corolla", "Rojo")
  tu_coche = coche ("Honda", "Civic", "Azul")

  #Mostrar el estado de los coches después de arrancar

  mi_coche.mostrar_estado()
  tu_coche.mostrar_estado()

  #Detener los coches
```

```
mi_coche.detener()
tu_coche.detener()

#Mostrar el estado final de los coches

mi_coche.mostrar_estado()
tu_coche.mostrar_estado()
```