Difference btw class and object

* **Class**: A class is a blueprint or template for creating objects. It defines the properties (attributes) and behaviors (methods) that the objects created from the class will have. Essentially, a class describes what an object will be.
* **Object**: An object is an instance of a class. It is a concrete realization of the class, containing specific data and the ability to perform actions defined in the class.
* **Example:**

**Class Dog:**

**def\_\_init\_\_(self,name,age):**

**self.name =name**

**self.age =age**

**Def bark(self):**

**return “Woof”**

**my\_dog=Dog(“buddy”,3)**

**print(my\_dog.name)**

**print(my\_dog.bark())**

Constructor Method (\_\_init\_\_) vs

\_\_str\_\_() Function:

* **Constructor Method (\_\_init\_\_)**: This special method is called when an object is instantiated from a class. It initializes the object's attributes and sets up any necessary state. It’s where you define what happens when you create an instance of a class.
* **\_\_str\_****\_() Function**: This special method is called when you want to get a string representation of an object, typically when you use the print() function or str(). It helps provide a user-friendly description of the object.
* **Example:**

**Class Car:**

**def \_\_init\_\_(self,make,model):**

**self.make =make**

**self.model=model**

**def \_\_str\_\_(self):**

**return** **f\”{self.make} {self.model}”**

**my\_car =Car(“Toyota”,”Grandy”)**

**print(my\_car)**

**print(my\_car.make)**