Class

Command	Structure	Example	Obs
Define Class	**class** [nomeClasse] **(** [class_inherits] **) :**	**class** Animal**(object) :**	
Define Objeto	[nomeObjeto] **=** [nomeClasse] **(** [variable1] **,** [variable2] **)**	zebra **=** Animal**(**'Jeffrey' **,** 21**)**	Instances = members of a class
__init__	**def __init__ (self ,** [arg1] **,** [arg2]**) :**	**def __init__ (self,** name**,** age**):**	first parameter (self) is used to refer to the object being created
	def __init__ (self , [variable1] **,** [variable2]**) :** **self.**[atributo1] **=** [variable1]	**def __init__ (self,** name**):** **self.**name **=** name	**Metodo construtor** para inicializar objeto
Methods **Access Method**	**def** [nomeMetodo] **(self ,** [variable1] **,** [variable2]**) :** [nomeObjeto] **.** [nomeMetodo] **( )**	**def** description**(self):** **print self.**name my_animal**.**description**()**	Methods = function in a class
Class Methods **@classmethod**	**@classmethod** **def** [nomeMetodo] **(cls,** [variable2] **) :** **cls.** [class_variable] **=** [variable2]	**@classmethod** **def** setAmount **(cls,** num**) :** **cls.**amount **=** num	Metodo utilizado para **alterar** o valor de uma **ClassVariable** para **todos** os **objetos**
Static Methods **@staticmethod**	**@staticmethod** **def** [nomeMetodo] **(** [variable1] **,** [variable2] **) :**	**@staticmethod** **def** idade **(**dataAtual**,** dataNascimento**) :**	Metodo que **não depende de nenhum objeto** da classe ou de nenhuma ClassVariable em especifico
Class Variables	**class** [nomeClasse] **(** [class_inherits] **) :** [class_variable] **=** [value]	**class** Animal**(** **object** **):** amount **=** 5	Class Variable = **atributo** que esta **disponível** para **todos** os **objetos** de uma classe.
Access Member Variables	[nomeClasse] **.** [class_variable] [nomeObjeto] **.** [class_variable]	Animal**.**amount my_animal**.**amount	Pode ser **acessado pela classe** ou **pelo objeto** da classe

Encapsulamento

Command	Structure	Example	Obs
Public	[atributo]		
	def [nomeMetodo] **(self ,** [variable1] **,** [variable2]**) :**		
Private	_[atributoPrivado]	**def __init__ (self ,** [variable1] **,** [variable2]**) :** **self.**_nome **=** [variable1]	
	__[atributoPrivado]	**def __init__ (self,** name**) :** **self.**__nome **=** name	
	def _[nomeMetodo] **(self ,** [variable1] **,** [variable2]**) :**		Metodo privado mas ainda **pode ser acessado publicamente** usando **_nomeMetodo**
	def __[nomeMetodo] **(self ,** [variable1] **,** [variable2]**) :**		Metodo privado só acessível internamente a classe
Getters **@property**	**@property** **def** [atributoPrivado] **(self):** **return self.**__[atributoPrivado] [nomeObjeto] **.** [atributoPrivado]	**@property** **def** nome**(self):** **return self.**__nome **print(**zebra**.**nome**)**	Permite **acessar valor** dos **atributos privados** utilizando o mesmo nome do atributo
Setter **@**[atributo]**.setter**	**@**[atributoPrivado]**.setter** **def** [atributoPrivado] **(self ,** [variable] **):** **self.**__[atributoPrivado] **=** [variable] [nomeObjeto] **.** [atributoPrivado] **=** [variable]	**@**nome**.setter** **def** nome **(self ,** value**):** **self.**__nome **=** value zebra**.**nome **=** 'Alvaro'	Permite **alterar valor** dos **atributos privados** utilizando o mesmo nome do atributo
Deleter **@**[atributo]**.deleter**	**@**[atributoPrivado]**.deleter** **def** [atributoPrivado] **(self):** **self.**__[atributoPrivado] **= None** **del** [nomeObjeto] **.** [atributoPrivado]	**@**nome**.deleter** **def** nome **(self):** **self.**__nome **= None** **del** zebra**.**nome	Permite **deletar valor** dos **atributos privados**

Herança

Command	Structure	Example	Obs
Inheritance	**class** [Derivated Class] **(** [Parent Class] **) :**	**class** Programador**(**Empregado**):**	
__init__	**def __init__ (self ,** [variable1] **,** [variable2] **,** [variable3] **) :** **super(). __init__(** [variable1] **,** [variable2] **)** **self.**[atributo3] **=** [variable3]	**def __init__ (self,** nome**,** sobrenome**,** linguagem**) :** **super(). __init__(**nome**,** sobrenome**)** **self.**linguagem **=** linguagem	Inicializa o objeto chamando o construtor da superclasse pros atributos da superclasse Completa com os atributos da subclasse
Super	**super (** [Derivated Class] **, self).** [method name] **(** [variable] **)**		Acessar atributos e métodos da superclasse
isinstance	**isinstance(** [nomeObjeto] **,** [nomeClasse] **)**		Retorna True/False se o [objeto] é uma instancia da [classe]
issubclass	**issubclass(** [nomeSubclasse] **,** [nomeSuperclasse] **)**		Retorna True/False se a [subclasse] é derivada da [Superclasse]

Metodos Especiais

Command	Structure	Example	Obs
	São sempre envolvidos por __ e __		
Representation Method **Access __repr__ method**	**def __repr__ (self) :** **return '** [declaracao_do_objeto] **'**	**def __repr__ (self):** **return '**Empregado('**{**nome**}**', '**{**sobrenome**}**') **'** **print(**nomeObjeto**)**	Exibe 'info' quando manda printar um objeto Usado para debugging, logging Objetivo de dar informações aos desenvolvedores
	repr([nomeObjeto] **)**		
Str Method **Access __str__ method**	**def __str__ (self) :** **return '** [info] **'**	**def __str__ (self):** **return '** **{**nomeCompleto**}** - **{**email**}** **'** **print(**nomeObjeto**)**	__str__ tem preferencia sobre o __repr__ Então se os 2 forem definidos e mandar printar o objeto, sera chamado o __str__ Exibe 'info' quando mandar printar um objeto Usado para exibir dados do objeto Objetivo de dar informações aos usuários
	str([nomeObjeto] **)**		
__add__	**def __add__ (self, other) :** **return self.**[atributoA] **+ other.**[atributoA]	**def __add__ (self, other) :** **return self.**salario **+ other.**salario **print(** objeto1 + objeto2 **)**	Metodo é chamado ao somar 2 objetos