Classes

class MyClassName:
myClassFunction(self):

Constructor

def **init**(self):

Getters, Setters and Deleters

Example:

```
def get_age(self):
    return self._age

def set_age(self, new_age):
  if isinstance(new_age, int):
    self._age = new_age
  else:
    raise TypeError

def delete_age(self):
  del self._age
```

Inheritance

Single Inheritance

```
class ParentClass:
  #class methods/properties...

class ChildClass(ParentClass):
  #class methods/properties...
```

Multiple Inheritance

```
class GrandParentClass():
  #class methods/properties...

class ParentClass(GrandParentClass):
  #class methods/properties...

class ChildClass(ParentClass):
  #class methods/properties...
```

OR

```
class YellowClass():
  #class methods/properties...

class FruitClass():
  #class methods/properties...

class BananaClass(YellowClass,FruitClass):
  #class methods/properties...
```

Note: super() would call YellowClass rather than FruitClass, as it's listed first
But you could say FruitClass.method(self) in order to do so

Super

Inside you childClass you can call

```
super().method(attributes)
```

In order to call the parent classes method

Related to:

[[Python]] #Classes

Dunder Methods / Overloading

common examples:

- **\init**
- **\repr**
- **\add**

simple define the method in the class, and it'll overload the parent method