

Mise en œuvre de la programmation objet



Le constructeur

➤ **Tous les attributs** doivent être initialisés dans le constructeur `__init__`

- Passés en paramètre

```
def __init__(self, par1, par2, ...):  
    self.att1_ = par1  
    self.att2_ = par2
```

- *ou Calculés en fonction des paramètres ou autres attributs*

```
def __init__(self, par1, par2, ...):  
    self.att1_ = par1  
    self.att2_ = par2  
    self.att3_ = f(par1, par2 ...)
```

- *ou Fixés*

```
def __init__(self, par1, par2, ...):  
    self.att1_ = par1  
    self.att2_ = par2  
    self.att3_ = expression en fonction de par1 et par2  
    self.att4_ = constante
```

Création d'une instance

➤ Création d'une instance d'une classe

- ➔ c'est le constructeur `__init__` qui est appelé
- ➔ Il faut donc que tous les paramètres attendus par le constructeur de la classe apparaissent lors de la création

classe MaClasse

def __init__(self, param1, param2, ... paramn):

....

Instance = MaClasse(arg1, arg2, ...argn)

➤ Cas des paramètres par défaut

- Les paramètres pouvant prendre une valeur par défaut
 - Doivent apparaître à la fin de la liste des paramètres
 - S'ils sont omis lors de la création de l'instance, ils doivent l'être dans l'ordre en partant du dernier

Héritage

➤ Création d'une sous-classe

➔ La classe parente doit apparaître au moins deux fois

*Class Fille(**Parente**):*

def __init__(self, par1, par2, ...):

***Parente**.__init__(self, arg1, arg2, ...)*

➔ Dans l'appel du constructeur `__init__` de la classe parente, **tous les paramètres nécessaires** doivent être fournis