

La programmation orientée objet

Introduction et vocabulaire

Les objets et leurs attributs

- La POO est un style de programmation centré sur les « **objets** »
- Dans la vie courante... les « objets » peuvent être rangés en **classes**
- Exemples :



- Chaque objet a des **attributs**
- Il existe des **méthodes** d'interaction spécifiques avec les objets de chaque classe



Informations d'un concessionnaire



- Un concessionnaire dispose de voitures
- Elles ont différentes marques, couleurs, modèles etc.
- Il existe différentes façons de placer ces informations dans des variables



Mise en application



- **Proposer une façon de stocker dans des variables les attributs des différentes voitures** du concessionnaire.
- Attributs à stocker en mémoire : **marque, modèle, couleur, année**

C'est à vous !



Informations d'un concessionnaire

```
# voiture 1:  
marque1 = "Renault"  
modèle1 = "Mégane"  
couleur1 = "bleu"  
année1 = 2007
```

```
# voiture 2:  
marque2 = "Ferrari"  
modèle2 = "Testa Rossa"  
couleur2 = "rouge"  
année2 = 2013
```

```
# voiture 3:  
marque3 = "Ford"  
modèle3 = "Fiesta"  
couleur3 = "jaune"  
année3 = 1990
```

```
# on peut grouper les données de même nature :  
marques = ["Renault", "Ferrari", "Ford"]  
modèles = ["Mégane", "Testa Rossa", "Ford"]  
couleurs = ["bleu", "rouge", "jaune"]  
années = [2007, 2013, 1990]
```

```
# on peut grouper les données pour chaque voiture:  
# avec une liste :  
voiture1 = ["Renault", "Mégane", "bleu", 2007]  
etc.
```

```
# avec un dictionnaire:  
voiture1 = {"marque": "Renault", "modèle": "Mégane",  
            "couleur": "bleu", "année": 2007}  
etc.
```



Informations d'un concessionnaire

Un peu rébarbatif ?...



Informations d'un concessionnaire... avec classe

- On définit une **classe** qui regroupe les objets identiques
- Dans chaque classe on spécifie les **attributs** des objets regroupés
- On crée des **instances** de chaque classe
- Noter la **notation pointée**

```
class Mégane_2007:  
    marque = "Renault"  
    modèle = "Mégane"  
    couleur = "bleu"  
    année = 2007
```

```
class Testa_Rossa_2013:  
    marque = "Ferrari"  
    modèle = "Testa Rossa"  
    couleur = "rouge"  
    année = 2013
```

```
voiture1 = Mégane_2007()  
voiture2 = Testa_Rossa_2013()  
...  
voiture19 = Mégane()  
print(voiture1.marque) # "Renault"  
print(voiture2.marque) # "Ferrari"
```



Informations d'un concessionnaire... avec classe

- C'est mieux mais encore limité...
- Il serait intéressant de spécifier les attributs à la création d'un objet :

```
une_voiture = Voiture("Ford", "Fiesta", "orange", 2000)
```

- C'est possible en spécifiant un **constructeur**



Informations d'un concessionnaire... avec classe

```
class Voiture:
```

`self` représente l'instance en cours

```
def __init__(self, marque, modèle, couleur, année):  
    self.brand = marque  
    self.model = modèle  
    self.colour = couleur  
    self.year = année
```

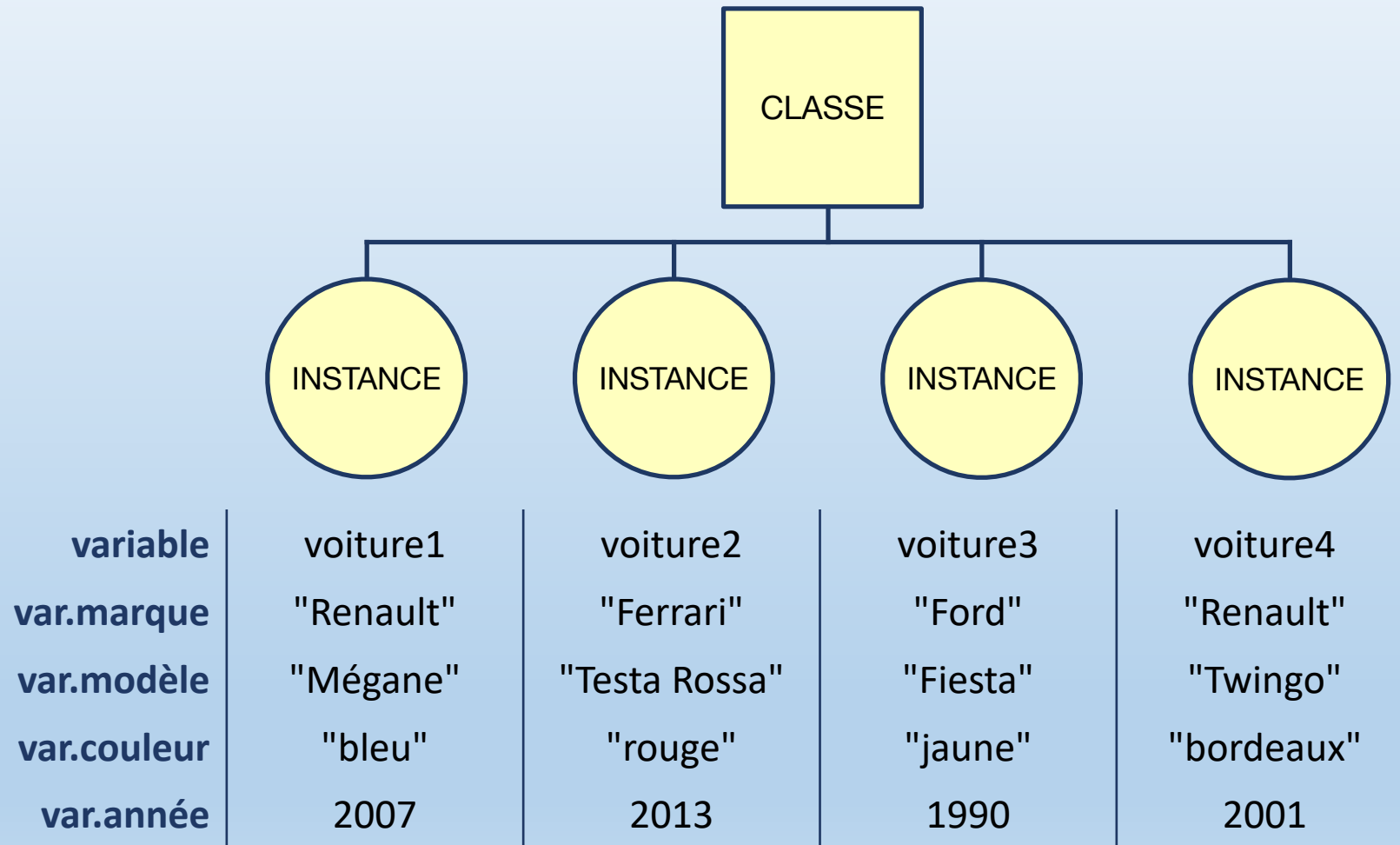
constructeur

```
une_voiture = Voiture("Peugeot", "308", "noir", 2019)
```

création d'une instance



Une classe – des instances



Mise en pratique - recenser des animaux

- À compléter...



Les méthodes sur les objets

- Il est possible de créer des fonctions applicables sur les instances que l'on crée
- Ses fonctions particulières sont appelées des méthodes
- On les définit dans le bloc de définition d'une classe, avec le mot clé **def** ; exemple :

```
class Voiture:
```

```
    def __init__(self, marque, modèle, couleur, année):  
        self.brand = marque  
        self.model = modèle  
        self.colour = couleur  
        self.year = année
```

```
    def presentation(self):  
        print(f"Voici une {self.brand} {self.model} datant de {self.year}.")
```

```
une_voiture = Voiture("Peugeot", "308", "noir", 2019)
```



Les méthodes sur les objets

```
class Voiture:
```

constructeur

```
def __init__(self, marque, modèle, couleur, année):  
    self.brand = marque  
    self.model = modèle  
    self.colour = couleur  
    self.year = année
```

méthode

```
def presentation(self):  
    print(f"Voici une {self.brand} {self.model} datant de {self.year}.")
```

```
une_voiture = Voiture("Peugeot", "308", "noir", 2019)
```

instanciation

```
une_voiture.presentation()
```

appeler la méthode sur l'instance



La POO – introduction et vocabulaire

