

# PROGRAMMATION ORIENTÉE OBJET : SYNTHÈSE DE COURS

`class Dé:`

Création d'une classe « Dé ». Pas de parenthèses  
Par convention, les noms de classe commencent par une majuscule.

```
"""Crée un objet "dé" dont on précise le nombre de faces"""
```

Documentation qui s'affichera en tapant `help(Dé)`

```
def __init__(self, nbre_faces):  
    """nbre_faces : entier strictement positif"""
```

Méthode « Constructeur » qui sera appelée lors de l'instanciation (création) d'un objet de la classe Dé

```
    self.nombre_de_faces = nbre_faces  
    self.valeur = self.relancer(False)
```

Déclaration des « attributs » (variables) dans la méthode constructeur.

```
def relancer(self, pipé = False):
```

```
    """pipé : booléen"""
```

```
    if pipé:
```

```
        self.valeur = random.randint(self.nombre_de_faces // 2, self.nombre_de_faces)
```

```
    else:
```

```
        self.valeur = random.randint(1, self.nombre_de_faces)
```

```
    return self.valeur
```

Puis, déclaration des autres « méthodes » (fonctions)

```
def __repr__(self):
```

```
    return str(self.valeur)
```

Quelque soit la méthode, le 1<sup>er</sup> paramètre est toujours `self`  
Il désigne l'instance à laquelle on va appliquer cette méthode.

```
a = Dé(6)
```

Création d'une nouvelle instance de l'objet Dé.  
On met toujours des parenthèses.

```
print(a.valeur)  
print(a.nombre_de_faces)  
print(a)
```

On a accès aux attributs.  
Pas de parenthèses.

```
print(a.relancer())  
print(a.relancer(True))
```

On a également accès aux méthodes.  
Toujours des parenthèses.