

Héritage de « classes »

- Créer un « véritable héritage de classe » ?
- Exemple :
 - Classe mère **Pers** (Personnages)
 - Attribut : **nom**
 - Méthode : **toString** (produit le nom)
 - *Fonction-constructeur **Pers** et prototype **Pers.prototype***
 - Classe fille **PersF** (Personnages fictifs)
 - Attributs : **nom**, **origine**
 - Méthode : **toString** (produit le nom)
 - *Fonction-constructeur **PersF** et prototype **PersF.prototype***

Héritage de « classes »

- Code pour la **classe mère** Pers

```
function Pers (nom) {  
  this.nom = nom;  
}
```

```
Pers.prototype.toString = function () {  
  return this.nom;  
}
```

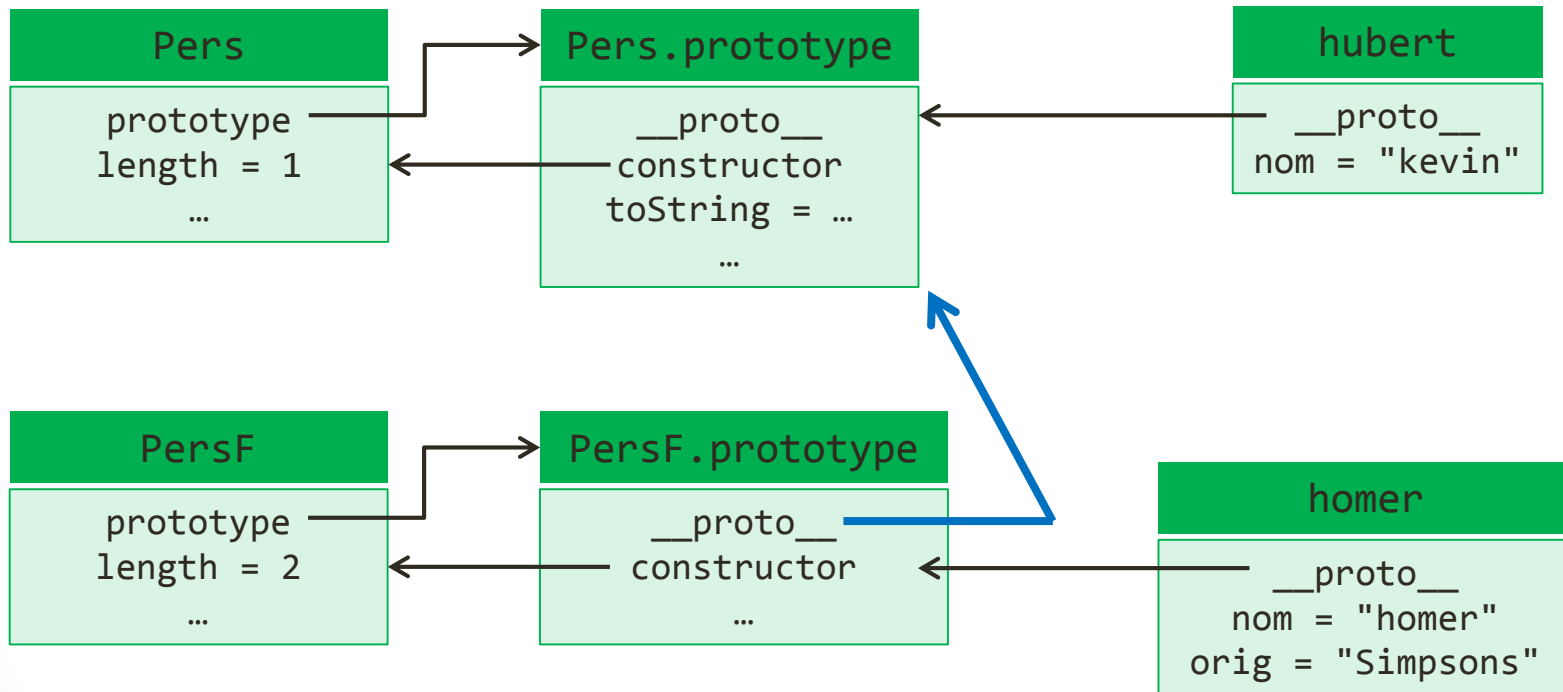
Héritage de « classes »

- Code pour la classe fille PersF
 - Fonction constructrice
 - arguments : nom, origine
 - devrait faire appel à Pers (pour le nom)
- ```
function PersF (nom, origine) {
 Pers.call(this, nom);
 this.origine = origine;
}
```
- Alternative :

```
this.tmpFnc = Pers; this.tmpFnc(nom);
```

# Héritage de « classes »

- Cela suffit-il ?



- Il faut encore établir un lien entre les prototypes !

# Héritage de « classes »

- Établir le lien entre les prototypes :

```
PersF.prototype.__proto__ = Pers.prototype;
```

- Ou encore (sous certains navigateurs) :

```
Object.setPrototypeOf(PersF.prototype,
Pers.prototype);
```

- Version alternative (sans \_\_proto\_\_) :

```
PersF.prototype = Object.create(Pers.prototype);
PersF.prototype.constructor = PersF;
```