

Understanding JavaScript Prototypes



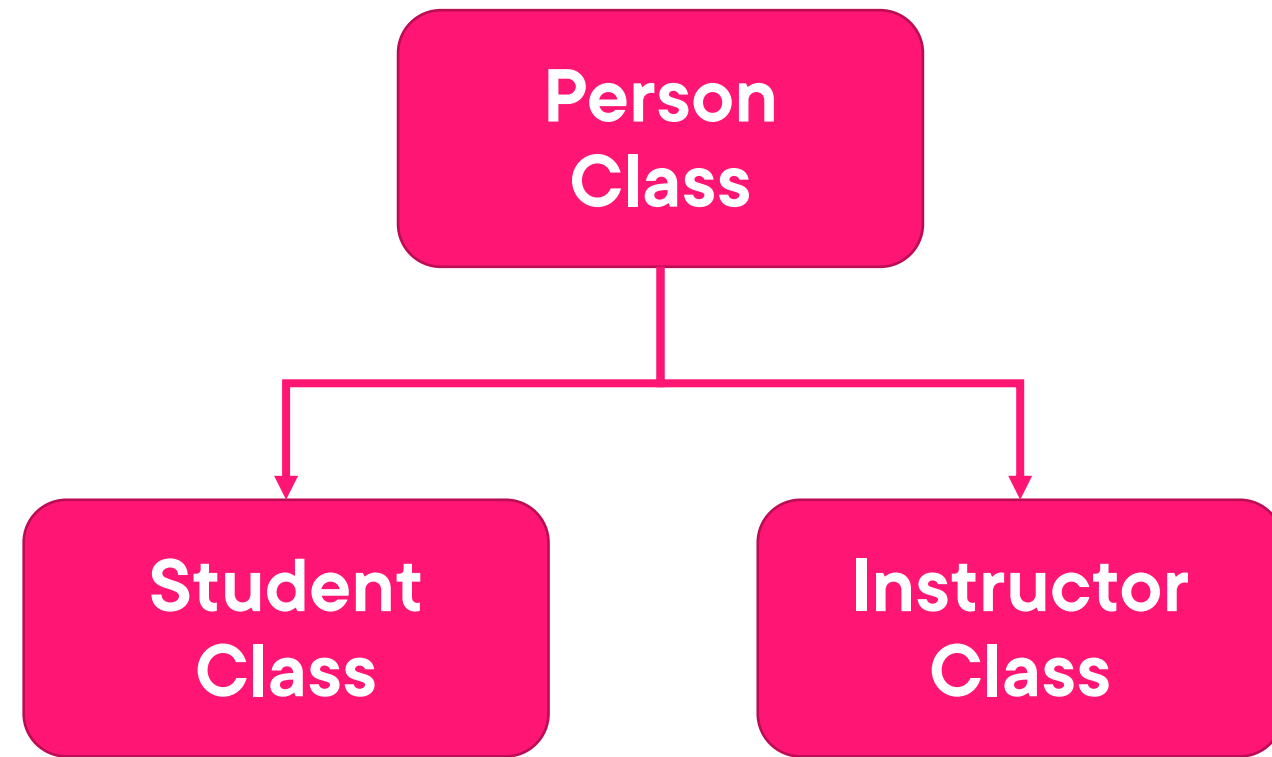
Jim Cooper

Software Engineer

@jimthecoop | jcoop.io

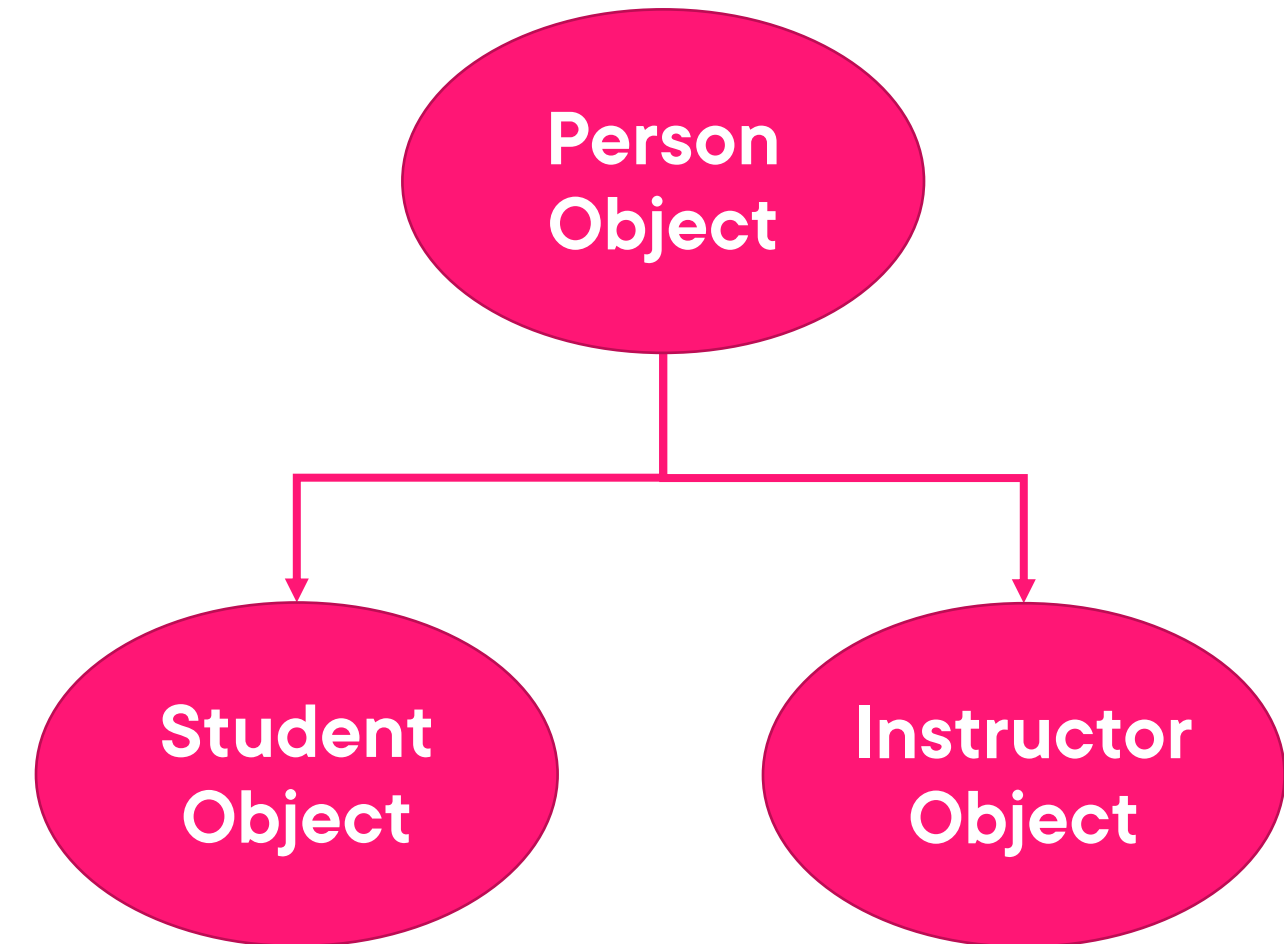


Classes



```
class Person { ... }  
class Student extends Person { ... }  
class Instructor extends Person { ... }  
  
let student = new Student(...);
```

Prototypes



```
let person = {...};  
let student = { prototype: person };  
let instructor = { prototype: person };
```

```
let Person = {
  firstName: '',
  lastName: '',
  age: 0
}

let Student = {
  courses: [],
  enroll(...) { ... }
}

Object.setPrototypeOf(Student, Person);

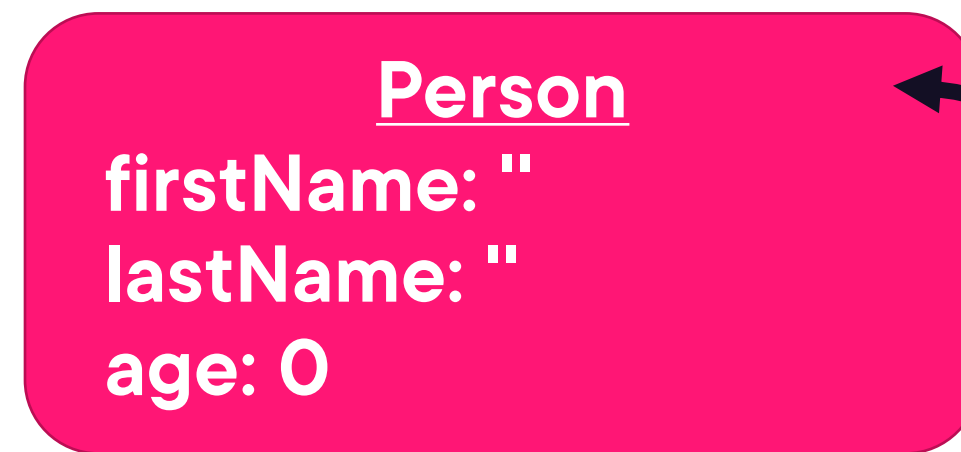
let jim = {
  firstName: 'Jim',
  lastName: 'Cooper'
}

Object.setPrototypeOf(jim, Student);

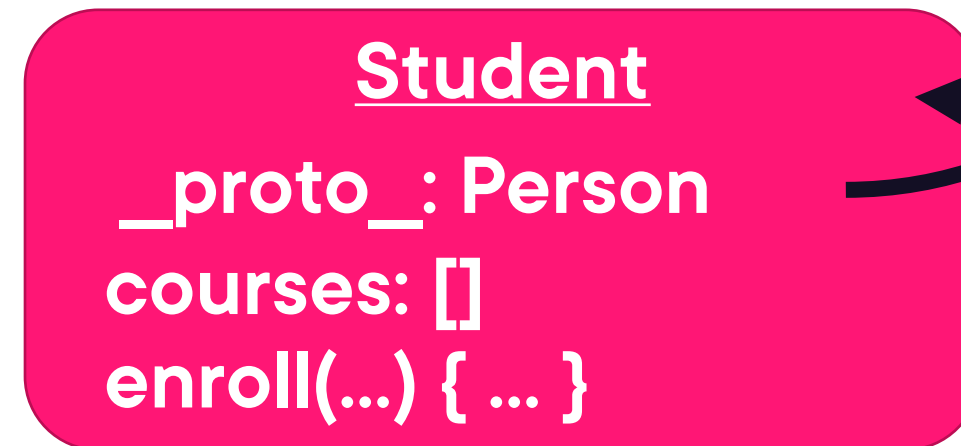
jim.age; // returns 0

jim.firstName; // returns 'Jim'
```

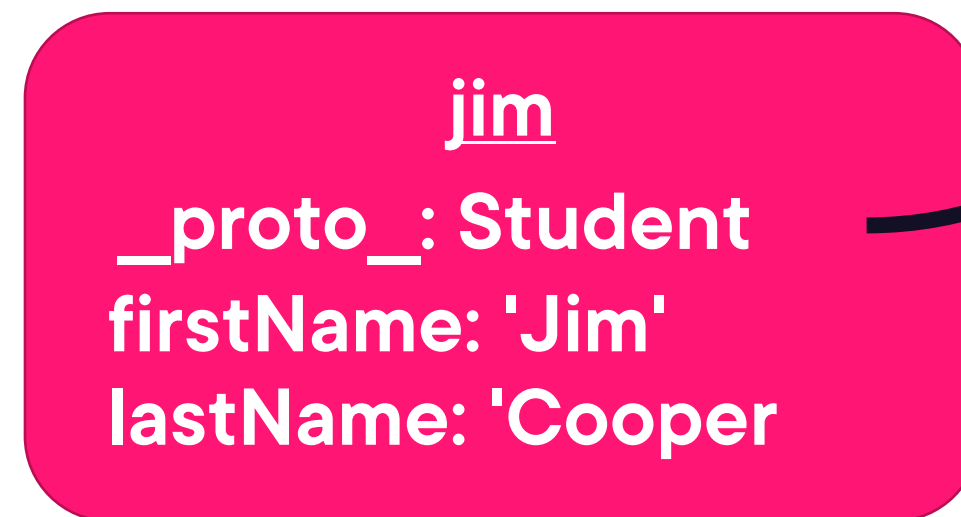
?



?



?



```
let Person = {
  firstName: '',
  lastName: '',
  age: 18
}

let Student = {
  courses: [],
  enroll(...) { ... }
}

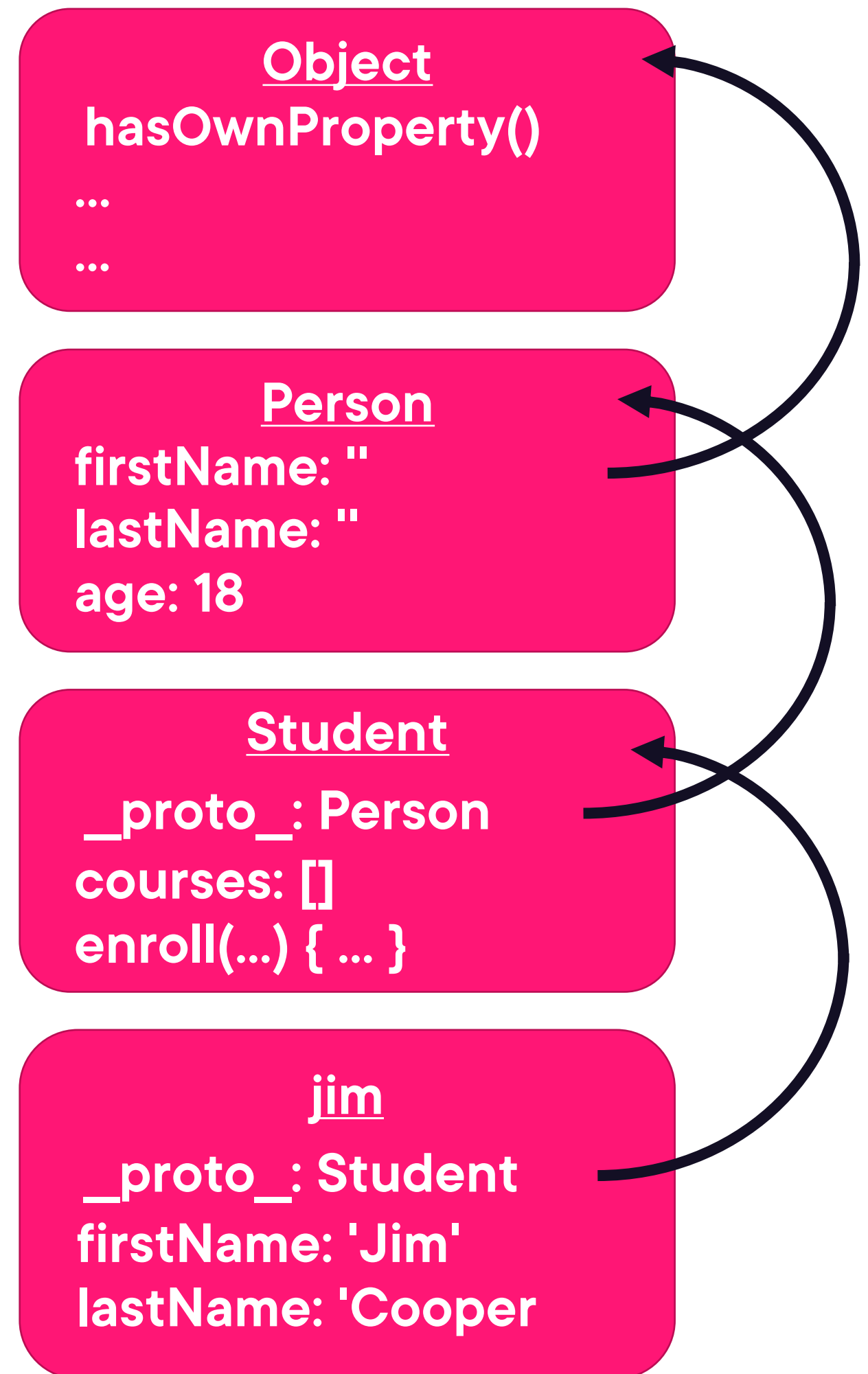
Object.setPrototypeOf(Student, Person);

let jim = {
  firstName: 'Jim',
  lastName: 'Cooper'
}

Object.setPrototypeOf(jim, Student);

jim.age;

jim.hasOwnProperty('age'); //false
```



Up Next:

Using JavaScript Constructor Functions

