

# "Pseudo-Private" Properties

The addition of private fields and properties is **relatively new** - in the past, such a feature was not part of JavaScript.

Hence you might find many scripts that use a concept which you could describe as "**pseudo-private** properties".

It would look like this:

```
class User {  
  constructor() {  
    this._role = 'admin';  
  }  
}  
  
// or directly in an object  
  
const product = {  
  _internalId: 'abc1'  
};
```

## What's that?

It's a quite common convention to prefix private properties with an underscore () to signal that they should not be accessed from outside of the object.

**Important:** It's **just a convention** that should signal something! It does **NOT technically prevent** access. You **CAN** run this code without errors for example:

```
const product = {  
  _internalId: 'abc1'  
};  
console.log(product._internalId); // works!
```

It's really just a hint that developers should respect. It's not as strict as the "real" private properties introduced recently (`#propertyName`).

