

# Clases en JavaScript

Carlos J. Pérez Sánchez

# Clase

Una clase es una especie de "plantilla" en la que se definen los atributos y métodos predeterminados de un tipo de objeto.

Esta “plantilla” se crea para poder crear y manejar objetos fácilmente.



# Azúcar Sintáctico

Sintaxis de un lenguaje de programación diseñada para hacer algunas construcciones más fáciles de leer o expresar.

Esto hace el lenguaje "más dulce" para el uso por programadores: las cosas pueden ser expresadas de una manera más clara, más concisas, o de un modo alternativo que se prefiera, sin afectar a la funcionalidad del programa.

# La gran “mentira”

JavaScript es un lenguaje orientado a objetos, basado en prototipos.

Cuando creamos las clases, con sus propiedades, métodos, etc...  
Luego JavaScript “por debajo” lo está tratando como un prototipo.

Simplemente, las clases son una forma más sencilla e intuitiva de crear prototipos para los programadores.



# Instanciación

Al método de crear nuevos objetos mediante la lectura y recuperación de los atributos y métodos de una clase se le conoce como instanciación.

# Clase

```
class User {  
    name;  
    email;  
    password;  
    isAdmin;
```

} Propiedades

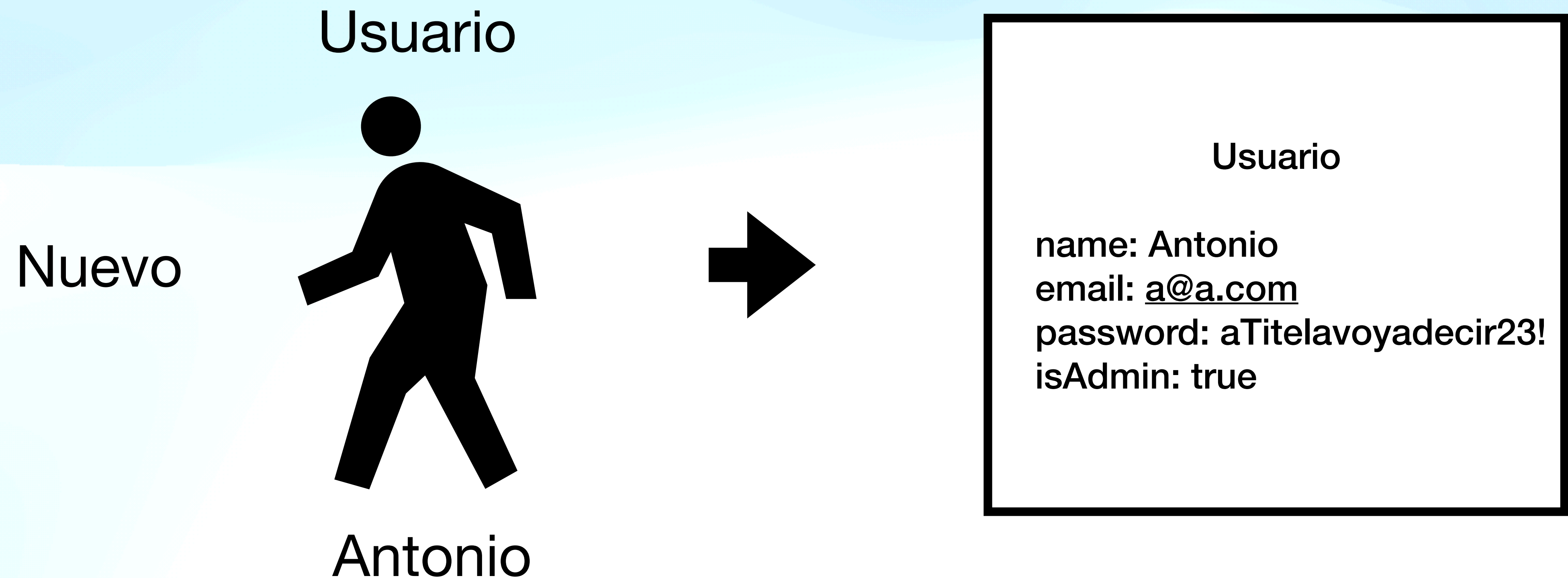
```
    constructor(name, email, password, isAdmin) {  
        this.name = name;  
        this.email = email;  
        this.password = password;  
        this.isAdmin = isAdmin;  
    }  
}
```

En el cuerpo de la clase se ejecuta todo en modo strict



# Constructor

Es el método que se ejecuta en el momento de instanciar la clase



# Métodos

```
class User {  
    .....  
  
    saludar() {  
        console.log( `Hola, soy `+this.name) ;  
    }  
}
```



# Herencia

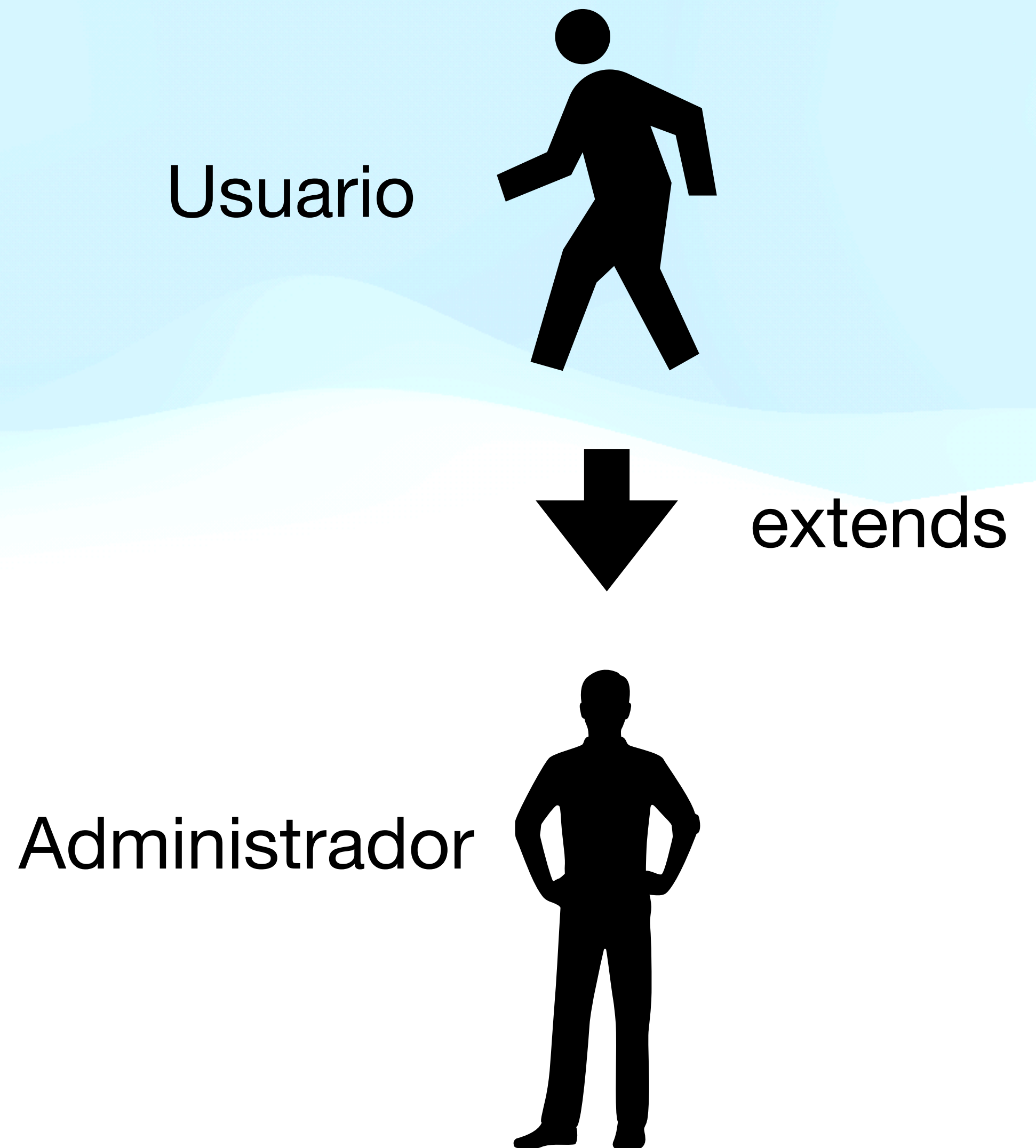




# Herencia

Un clase hereda los atributos y métodos públicos de otra, como si se hubieran definido para ella, y puede hacer uso de ellos.

Es útil si se quiere crear una clase que deriva de otra, pero que tiene atributos y métodos propios.





# Herencia

```
class Administrador extends User{  
    constructor(name, email, password, category) {  
        super(name, email, password);  
        this.category=category;  
    }  
}
```

Puede usar los métodos ya definidos en User, crear otros nuevos y sobrescribir métodos de User para sí misma

# Privatización de Clases

Una clase con atributos o métodos privados implica que estos sólo se podrán usar dentro de la clase.

Si intentamos acceder a algún atributo o invocar a un método privado, no podremos.

No existen las clases privadas en JavaScript.

¿Qué opciones tenemos?



# Métodos estáticos

Un método estático es aquel que puede invocarse sin necesidad de instanciar la clase.

# Getters y Setters

Permiten acceder (get) y modificar (set) valores de los atributos de las clases.

Deben escribirse como un método normal de una clase, pero con la palabra clave get o set delante.

Se invocan como un atributo de clase, no como un método (Las cosas de JavaScript)

```
class User{  
    .....  
    get getName() {  
        return this.name;  
    }  
  
    set setName(name) {  
        this.name=name;  
    }  
}
```

```
const nombre1 = user1.getName;  
  
user1.setName='John Appleseed'
```



# Propiedades Privadas

Sólo se puede acceder a ellas y modificarlas mediante get y set, siempre que se especifique así.

Permite otro filtro más de seguridad.

Se puede comprobar que un valor sea correcto antes de ser modificado un atributo, por ejemplo.

# Clases en JavaScript

Carlos J. Pérez Sánchez