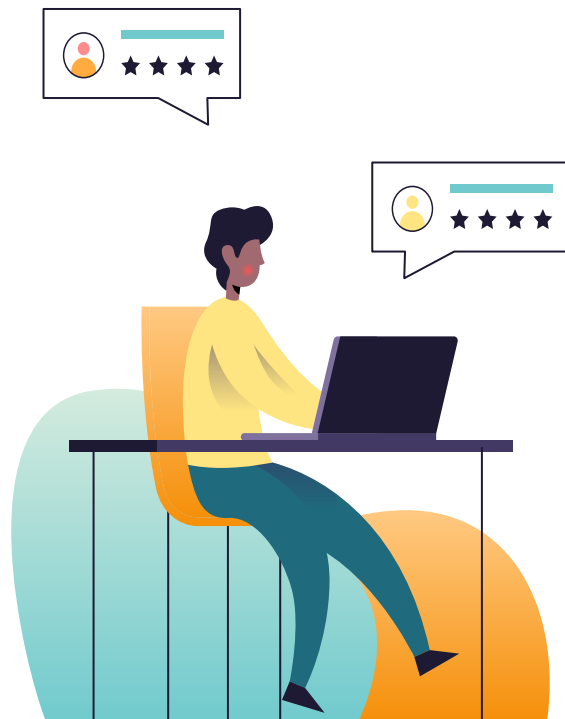


# Bienvenidos a Fundamentos de JavaScript



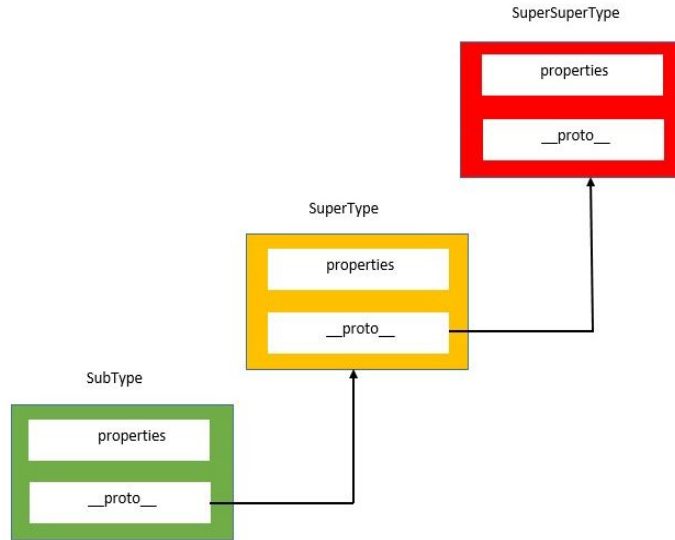
## *Prototype*

- Los prototipos son un mecanismo mediante el cual los objetos en JavaScript heredan características entre sí

Todos los objetos tienen una referencia a otro objeto llamado proto

## *Prototype*

JavaScript es a menudo descrito como un lenguaje basado en prototipos - para proporcionar mecanismos de herencia, los objetos pueden tener un objeto prototipo, el cual actúa como un objeto plantilla que hereda métodos y propiedades.



# Fundamentos de Javascript

```
var persona = {  
  nombre: "default",  
  apellido: "default",  
  getNombre: function(){  
    return this.nombre + " " + this.apellido;  
  },  
}  
  
var juan = {  
  nombre: "Juan",  
  apellido: "quiroz"  
}  
  
juan.__proto__ = persona  
console.log(juan.getNombre())
```

no usar proto

# Fundamentos de Javascript

```
function Persona(nombre, apellido){  
  this.nombre = nombre || "Juan";  
  this.apellido = apellido || "Quiroz";  
}  
Persona.prototype.getNombre = function(){  
  return this.nombre + " " + this.apellido  
}  
var marcos = new Persona("Marcos", "Caballero")  
var Willy = new Persona("Cristian", "Gonzales")  
var seba = new Persona("Sebastian", "morrone")  
console.log(marcos.getNombre())  
console.log(Willy.getNombre())  
console.log(seba.getNombre())
```

→ Nos vamos  
acercando a  
como usarlo

# Fundamentos de Javascript

```
var persona = {  
  nombre: "default",  
  apellido: "default",  
} //objeto cualquiera
```

```
var marcos = Object.create(persona)  
console.log(marcos)  
console.log(marcos.__proto__) //Objeto persona que creamos al principio
```

Object.create recibe un objeto como parámetro y crea un nuevo objeto cuyo prototipo es el primero que le pasamos.

Por lo tanto, vamos a poder acceder a todas las propiedades y métodos del objeto base en los nuevos objetos creados.

¡Nos veremos en la  
próxima clase!

