

COD - EANDO

Aquí empieza nuestro desafío





Nuestro Recorrido

01

Intro a la Programación

Moviendo nuestra neuronas.
Aprendiendo JS

02

El FrontEnd

Las Aplicaciones Web, estética y
dinamismo. HTML + Css y JS

03

Controlando el código

Una herramienta super-poderosa
llamada Git

04

El BackEnd

Procesando la información.
Ahora JS en el BackEnd con Node

05

Bases de Datos

Un poco de Sql con Mysql
Nuestro repositorio de datos.

06

Mejorando el FrontEnd

La Sazón con ReactJs
JS en front es más ordenado.



01

Intro a la Programación

Clase 4





Agenda

Repaso

Funciones para creación de objetos.

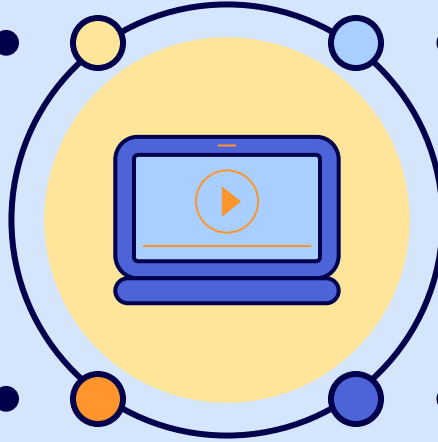
Repasando

Condicionales

Ciclos

Ambito de
Variables

Arrays y ciclos



Puntos importantes

La mayoría de las palabras reservadas son en inglés

Puedo usar un ciclo en incremento o decremento

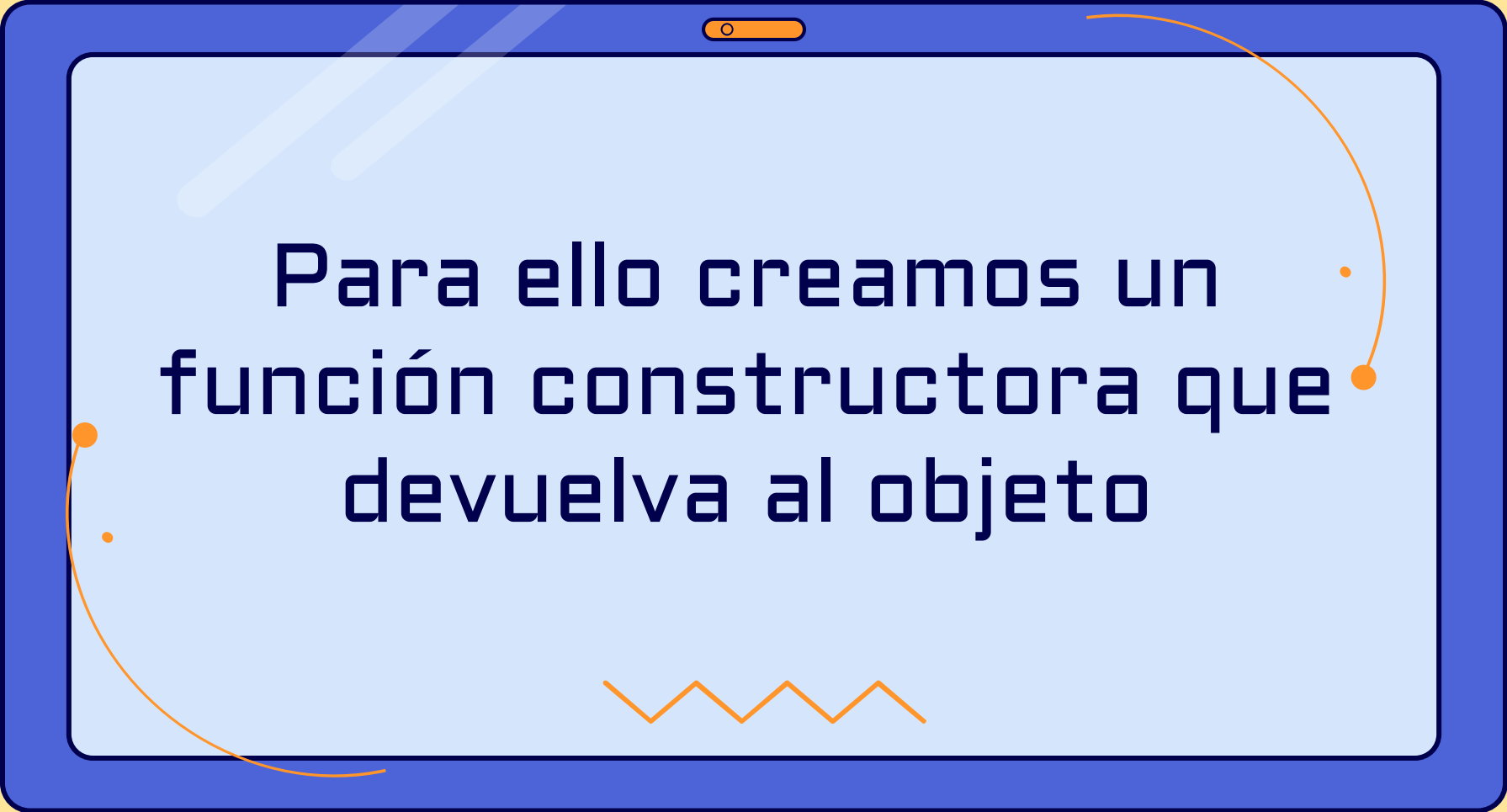
Los ciclos deben tener una condición de fin

Palabras reservadas



A blue rectangular frame with rounded corners and a dark blue border. Inside the frame, there are several orange decorative elements: a horizontal bar with a circle at the top center, a curved line with a dot on the right side, a curved line with a dot on the left side, and a zigzag line at the bottom center.

Vamos a optimizar a
nuestros objetos literales

A blue rectangular frame with rounded corners and a dark blue border. Inside the frame, there are several orange decorative elements: a horizontal bar with a circle at the top center, a curved line with dots on the right side, a curved line with dots on the left side, and a zigzag line at the bottom center.

Para ello creamos un
función constructora que
devuelva al objeto

Funciones constructoras

```
function Objeto( ) {  
    this.propiedad = ''  
}  
  
function Persona(nombre) {  
    this.nombre = nombre  
}
```

Creamos una función con nombre y la primera letra en mayúscula, es habitualmente usar esa convención.

En este ejemplo tenemos a una función **Persona**, la cual recibe el **nombre** y lo guarde dentro de su propiedad **nombre**

Hacemos referencia a **this** para que lo almacene dentro de la función.

No usamos el **return** en este caso, ya que la función retornara todo lo dentro de **this**

Instanciando un objeto

```
const perso = new Persona('pepe')  
// lo imprimo a ver que tiene  
console.log(perso)  
  
// puedo crear más  
const ana = new Persona('ana')  
const luis = new Persona('luis')
```

Para crear un objeto `Persona` usamos la palabra reservada **new**, en programación se conoce como instanciar al objeto, le pasamos como parámetro el **nombre**.

Lo guardamos en la constante **perso**.

Puedo seguir usando la función constructora de manera arbitraria, así crear más objetos.

Además es más fácil, crear nuestros objetos de esta manera.

Métodos en Funciones constructoras

```
function Persona(nombre) {  
  this.nombre = nombre  
  this.saludar = function () {  
    return 'Hola! mi nombre es: '  
+ this.nombre  
  }  
}  
  
const perso = new Persona('pepe')  
perso.saludar()  
//devuelve Hola! mi nombre es pepe
```

Aquí incorporamos funciones o métodos en nuestra función constructora.

Definimos el método saludar(), que devolverá el un string o texto con el saludo, dentro de ese texto estará el nombre.

Para acceder al método uso la notación punto. Este método devuelve un string que puedo consologear o guardar en una variable.

Métodos en Funciones constructoras

```
function Persona(nombre, apellido) {  
  this.nombre = nombre  
  this.apellido = apellido  
  this.telefono = null  
  this.saludar = function () {  
    return 'Hola! mi nombre es: ' +  
    this.nombre  
  }  
  this.cambiarTelefono =  
  function(numeroTelefonico) {  
    this.telefono = numeroTelefonico  
  }  
}  
  
const perso = new Persona('pepe')  
perso.cambiarTelefono (1145788956)
```

Aquí incorporamos funciones o métodos en nuestra función constructora.

Podemos agregar más parámetros que serán necesarios para crear el objeto.

En este caso no agregamos el teléfono, dado que para crear a una persona no necesito el telefono, pero si creé un método para setear el el mismo, dentro de este método puedo agregar validaciones y demás.

A todos estos métodos los accedo mediante la notación punto.

Fin por hoy!



Sigan practicando

CREDITS: This presentation template was created by **Slidesgo**, including icons by **Flaticon**, infographics & images by **Freepik**

