

# Incorporaciones más importantes ES6 (II)

## Arrow functions

sintaxis simplificada

`const x = () => {} //function x () {return}`

contexto con this

`function() {}`

`function () { //establece la referencia c/ el objeto this al momento de ejecutarse la funcion`

Ej. en el metodo de un objeto

`flecha() => {}`

la referencia se toma del ámbito que contiene a la arrow function en el momento de su definición

Ej. en el metodo de un objeto

`("me despido.", this.nombre); //me despido. undefined`

## Clases

plantillas para crear objetos

traen conceptos de la vida real a nuestro código

las prop describen al objeto y los metodos indican acciones de ese objeto

Sintaxis

Antes de ES6

implementacion de clases se hacia con funciones constructoras

`function Persona(nombre, edad) { //la F. constructora se usa para luego crear una instancias`

`const persona1 = new Persona("jorge", 32)`

Despues de ES6

palabra reservada: "class" + Nombre con mayuscula + { constructor(parametros) {propiedades}}

`class Persona { constructor(nombre, edad) {`

instancia: `const x = new nombre de class (argumentos)`

Extends

`class Doctor extends Persona { //clase hija de clase persona (hereda sus prop y metodos)`

Instancias

Los objetos que provienen de dicha plantilla

Ppios Fundamentales

Encapsulamiento

principio de ocultar los detalles internos de la implementación de un objeto, /exponiendo solo las interfaces necesarias para interactuar con él.

Polimorfismo

permite que objetos de diferentes clases respondan a métodos con el mismo nombre, cada uno según su propia implementación

Abstraccion

proceso de ocultar los detalles complejos de la implementación y exponer solo las características esenciales y las funcionalidades de un objeto.

Herencia

permite que una clase (hijo) adopte propiedades y métodos de otra clase (padre)