

Conceptos fundamentales

OBJETOS EN JS

CONCEPTOS BASICOS

Clase Define las características del Objeto.

Objeto Una instancia de una Clase.

Propiedad Una característica del Objeto, como el color.

Método Una capacidad del Objeto, como caminar.

Constructor Es un método llamado en el momento de la creación de instancias.

Herencia Una Clase puede heredar características de otra Clase.

CONCEPTOS BASICOS

Encapsulamiento

Una Clase sólo define las características del Objeto,
un Método sólo define cómo se ejecuta el Método.

Abstracción

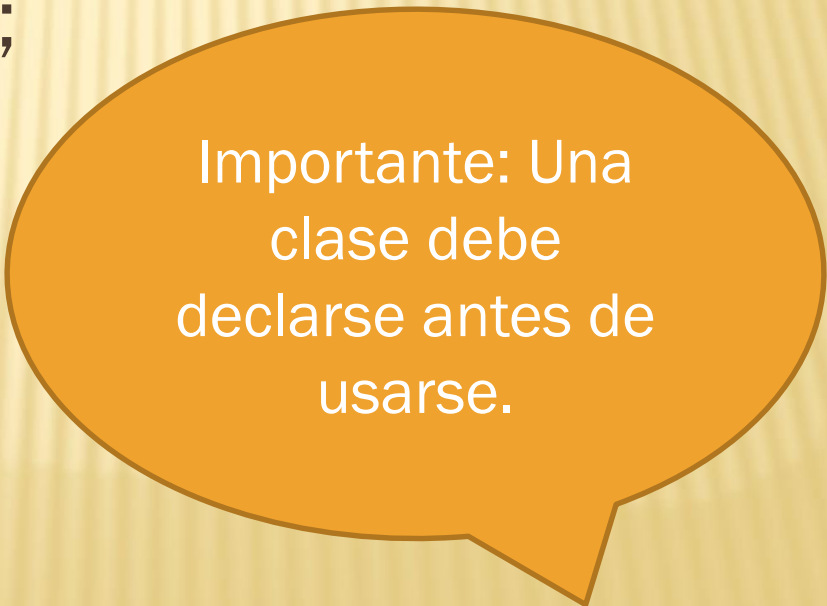
La conjunción de herencia compleja, métodos y propiedades que un objeto debe ser capaz de simular en un modelo de la realidad.

Polimorfismo

Diferentes Clases podrían definir el mismo método o propiedad.

DECLARACIÓN DE CLASES

```
class Poligono {  
    constructor (alto, ancho) {  
        this.alto = alto;  
        this.ancho = ancho;  
    }  
}
```



Importante: Una
clase debe
declararse antes de
usarse.

EL CONSTRUCTOR

- ✗ El método ***constructor*** es un método especial para crear e inicializar un objeto creado con una clase. Solo puede haber un método especial con el nombre "constructor" en una clase. Si esta contiene mas de una ocurrencia del método **constructor**, se arrojará un *Error SyntaxError*.
- ✗ Un ***constructor*** puede usar la *palabra reservada super* para llamar al **constructor** de una *superclase*.

MÉTODOS PROTOTIPO

```
class Poligono {  
    constructor(height, width)  
    {  
        this.height = height;  
        this.width = width;  
    }  
    get area() {  
        return this.calcArea();  
    }  
    calcArea() {  
        return this.height * this.width;  
    }  
}
```

MÉTODOS PROTOTIPO

```
var cuadrado = new Poligono(10, 10);
```

```
console.log(cuadrado.area);
```



Se accede al valor
como si fuera un
atributo.

SETTERS Y GETTERS

```
class Perro{  
    constructor(nombre){  
        this._nombre = nombre;  
    }  
    set nombre(value){  
        this._nombre = value;  
    }  
}
```


SETTERS Y GETTERS

```
var p = new Perro("Roberto");  
p.nombre = "Fido";
```

Accedo al método no al atributo.

El nombre del atributo y del método prototipo no deben coincidir

Reutilizando código

HERENCIA

DEFINICIÓN DE HERENCIA

La palabra clave **extends** es usada en la declaración o expresión de clases, para crear una clase hija de otra.

```
class claseHija extends clasePadre
```

IMPLEMENTACIÓN

```
class Square extends Polygon {  
    constructor(length, color) {  
        super(length, length);  
        this.color = color;  
    }  
}
```

Nota: En las clases extendidas, se debe llamar a ***super()***, antes de poder usar '***this***'. El no hacerlo provocará un reference error. `this.color = color;`