

Programación orientada a objetos

con JavaScript

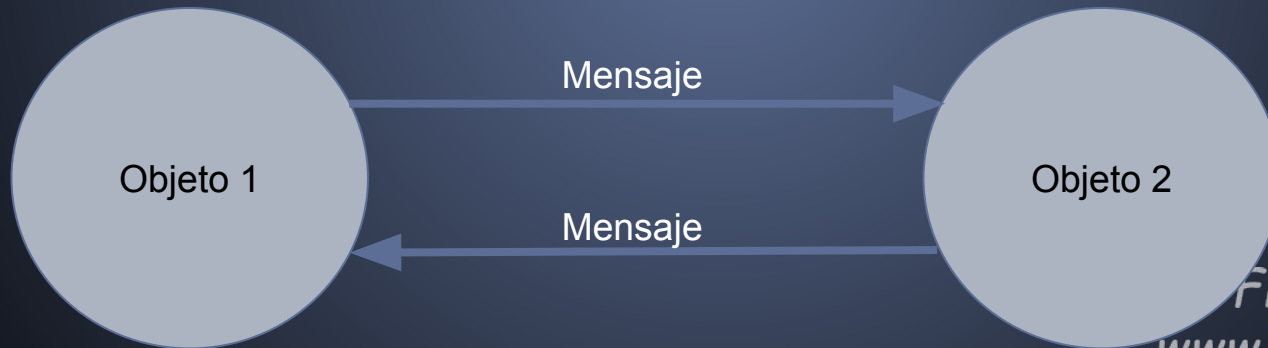
Francisco Arce
www.pacoarce.com

Programación orientada a objetos

- Simula 67
- Smalltalk
- C++
- Java
- C#
- Objective-C
- JavaScript

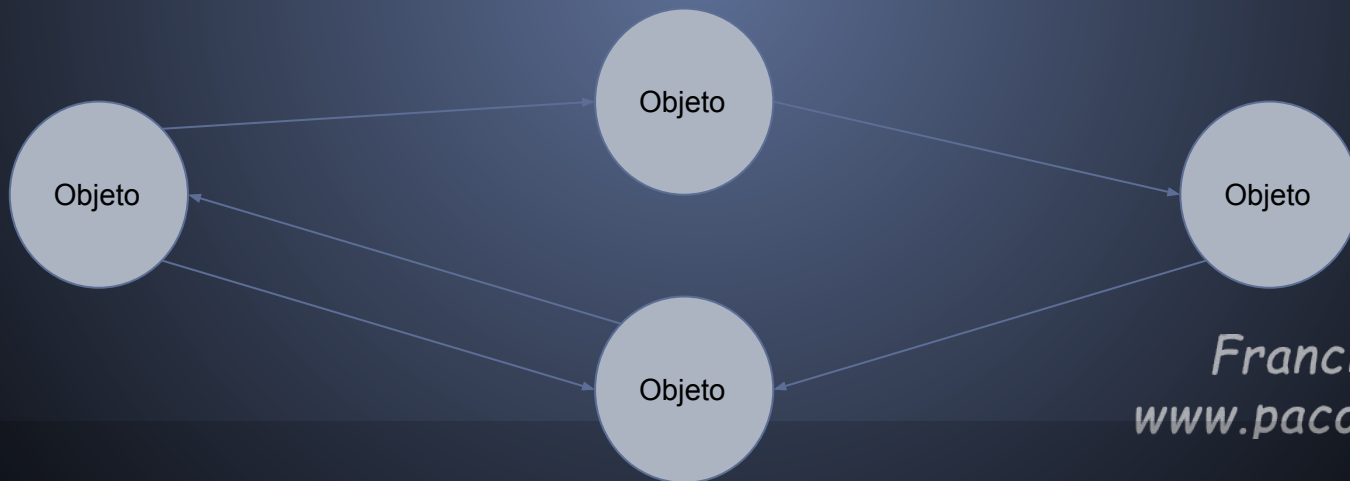
OOP con JavaScript

En la programación orientada a objetos, cada objeto es capaz de recibir mensajes, procesar datos y enviar mensajes a otros objetos.



OOP con JavaScript

Cada objeto puede verse como una pequeña “máquina independiente” con un papel o responsabilidad definida.



Francisco Arce
www.pacoarce.com

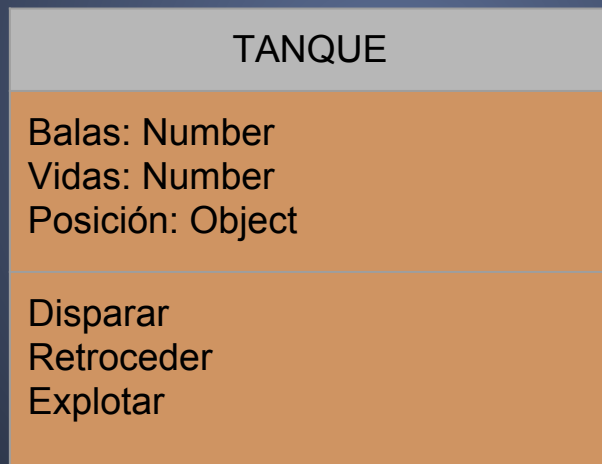
Clases, objetos e instancias

OOP con JavaScript

Francisco Arce
www.pacoarce.com

OOP con JavaScript

Clase: Define las características del Objeto.



OOP con JavaScript

Objeto: Una instancia de una Clase.

TANQUE
Balas: Number Vidas: Number Posición: Object
Disparar Retroceder Explotar

CLASE



INSTANCIAS



Francisco Arce
www.pacoarce.com

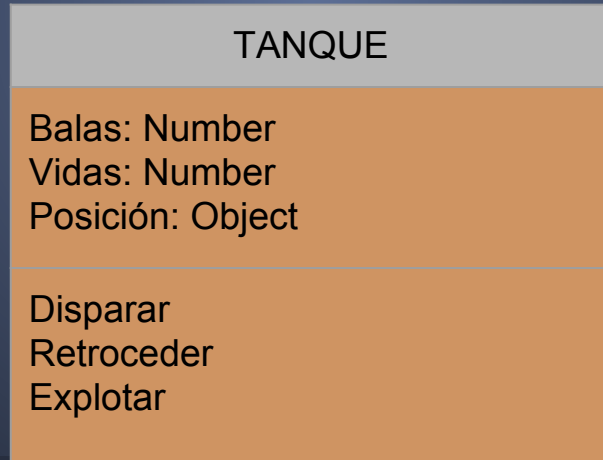
Elementos de un clase

OOP con JavaScript

Francisco Arce
www.pacoarce.com

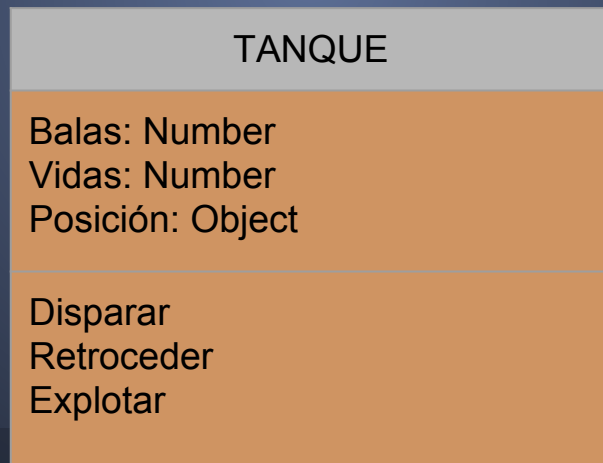
OOP con JavaScript

Propiedad: Una característica del Objeto, como el número de balas de un tanque.



OOP con JavaScript

Método: Una capacidad del Objeto, como disparar().



OOP con JavaScript

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

Principios de la OOP

OOP con JavaScript

Francisco Arce
www.pacoarce.com

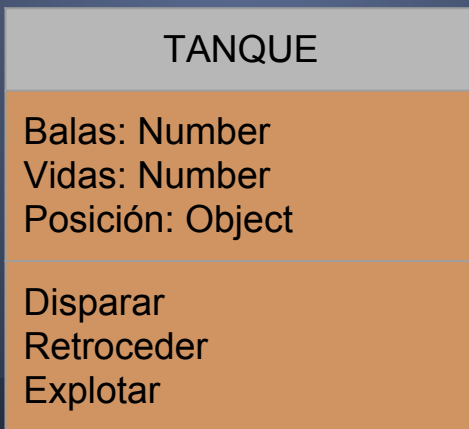
OOP con JavaScript

Herencia: Una Clase puede heredar características de otra Clase.



OOP con JavaScript

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



OOP con JavaScript

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

OOP con JavaScript

Polimorfismo: Diferentes Clases podrían definir el mismo método o propiedad.

TANQUE
Balas: Number Vidas: Number Posición: Object
Disparar Retroceder Explotar

AVION
Balas: Number Vidas: Number Posición: Object
Disparar Explotar Despegar

La propiedad *prototype*

OOP con JavaScript

Francisco Arce
www.pacoarce.com

OOP con JavaScript

Las propiedades deben establecerse a la propiedad prototipo de la clase (función), para que la herencia funcione correctamente.

```
var tanque1 = new Tanque();
```

OOP con JavaScript

Para trabajar con propiedades dentro de la clase se utiliza la palabra reservada this , que se refiere al objeto actual.

OOP con JavaScript

El acceso (lectura o escritura) a una propiedad desde fuera de la clase se hace con la sintaxis:

```
tanque1.balas = 100;
```

OOP con JavaScript

Los métodos siguen la misma lógica que las propiedades, la diferencia es que son funciones y se definen como funciones.

```
instancia.metodo();
```

OOP con JavaScript

Llamar a un método es similar a acceder a una propiedad, pero se agrega () al final del nombre del método, posiblemente con argumentos.

```
tanque1.avanza(10,10);
```

Programación orientada a prototipos

OOP con JavaScript

Francisco Arce
www.pacoarce.com

OOP con JavaScript

La programación basada en prototipos es un estilo de programación orientada a objetos en la que las clases no están presentes y la reutilización de comportamiento (conocido como herencia en lenguajes basados en clases) se lleva a cabo a través de un proceso de decoración de objetos existentes que sirven de prototipos.

OOP con JavaScript

Este modelo también se conoce como programación sin clases, *orientada a prototipos* o basada en funciones.

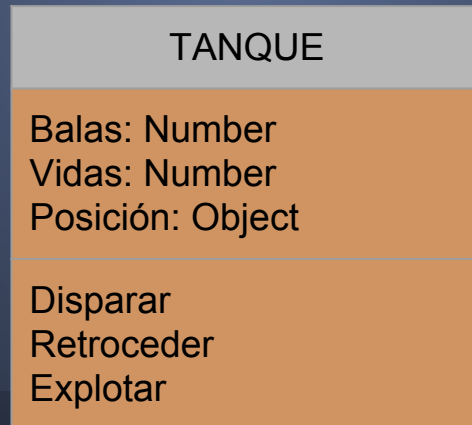
UML

OOP con JavaScript

Francisco Arce
www.pacoarce.com

UML

El lenguaje unificado de modelado o Unified Modeling Language (UML) permite visualizar las clases y las relaciones entre ellas.



UML

En JavaScript, el constructor y el nombre de la clase son los mismos.

Luego se enumeran las propiedades y posteriormente los métodos.

UML

Aunque en JS no se definen los tipos de dato, en el UML se indica el tipo de mismo.

Si el método no regresa ningún valor, se indica con :void.

Links

<https://developer.mozilla.org/en-US/docs/Web/JavaScript/Reference/Statements/class>

<https://en.wikipedia.org/wiki/ECMAScript>

Links

https://developer.mozilla.org/es/docs/Web/JavaScript/Guide/Details_of_the_Object_Model

https://developer.mozilla.org/es/docs/Web/JavaScript/Introducción_a_JavaScript_orientado_a_objetos

<https://developer.mozilla.org/es/docs/Web/JavaScript/Guide/Funciones>

Francisco Arce
www.pacoarce.com