



Clases. Miembros de una clase

-
- C.F.G.S. DAW
 - 6 horas semanales
 - Mes aprox. de impartición: Dic
 - iPasen - cjaedia071@g.educaand.es

Carmelo José Jaén Díaz

———— Índice ————



Objetivo

Glosario

Interacción persona - ordenador

Objetivos

Características. Usable.

Características. Visual.

Características. Educativo y actualizado.

OBJETIVO



- Profundizar en el concepto de objeto.
- Conocer y manejar funciones relativas al lenguaje sobre arrays, strings, números.

GLOSARIO



Backtracking. Estrategia utilizada en algoritmos que resuelven problemas que tienen ciertas restricciones. Este término fue creado por primera vez por el matemático D. H. Lehmer en la década de los cincuenta.

BOM (Browser Object Model). Convención específica implementada por los navegadores para que JavaScript pudiese hacer uso de sus métodos y propiedades de forma uniforme.

Expresión regular. Secuencia de caracteres que forman un patrón determinado, generalmente un patrón de búsqueda.

NaN. Propiedad que indica Not a Number (valor no numérico).

Objeto window. Aquel que soportan todos los navegadores y que representa la ventana del navegador. Se estudiará en profundidad en capítulos posteriores.

URI (Uniform Resource Identifier). Cadena de caracteres que identifica un recurso en una red de forma unívoca. Una URI puede ser una URL, una URN o ambas.

GLOSARIO



URN. Localizador de recursos en la web que funciona de forma parecida a una URL, pero su principal diferencia es que no indica exactamente dónde se encuentra dicho objeto.

INTRODUCCIÓN



Una clase tiene diferentes características que la forman, que generalmente se denominan **miembros**, y que normalmente son de dos tipos: **propiedades y métodos**.

Miembro	Descripción
Propiedad	Variable que existe dentro de una clase. Puede ser pública o privada.
Propiedad pública	Propiedad a la que se puede acceder desde fuera de la clase.
Propiedad privada (ES2020)	Propiedad a la que no se puede acceder desde fuera de la clase.
Propiedad computada	Función para acceder a una propiedad con modificaciones (getter/setter).

INTRODUCCIÓN



Miembro	Descripción
Método	Función que existe dentro de una clase. Puede ser pública o privada.
Método público	Método que se puede ejecutar desde dentro y fuera de la clase.
Método privado (ES2020)	Método que sólo se puede ejecutar desde dentro de la clase.
Constructor	Método especial que se ejecuta automáticamente cuando se crea una instancia.
Método estático	Método que se ejecuta directamente desde la clase, no desde la instancia.
Inicializador estático (ES2022)	Bloque de código que se ejecuta al definir la clase, sin necesidad de instancia.

VISIBILIDAD DE PROPIEDADES



Las propiedades de clase siempre van a tener una visibilidad específica, que puede ser pública (por defecto) o privada. En el primer caso, las propiedades pueden ser leídas o modificadas tanto desde dentro de la clase como desde fuera, en el segundo caso, sólo pueden ser leídas o modificadas desde el interior de la clase.

Nombre	Sintaxis	Descripción
Propiedad pública	<code>this.name</code>	Se puede acceder a la propiedad desde dentro y fuera de la clase.
Propiedad privada	<code>this.#name</code>	Se puede acceder a la propiedad sólo desde dentro de la clase.

EJEMPLO PROPIEDAD PÚBLICA



```
class Docente {  
  constructor(nom) {  
    this.nombre = nom;  
  }  
}
```

```
const docenteSAI = new Docente("Carmelo"); // { nombre: "Carmelo" }  
console.log(docenteSAI.nombre); // "Carmelo" (se puede acceder desde fuera)  
docenteSAI.nombre = "Carmelo José";  
console.log(docenteSAI.nombre); // "Carmelo Jose" (se modificó desde fuera)
```

EJEMPLO PROPIEDAD PRIVADA



```
class Docente {  
  #nombre; // Es obligatorio declarar la propiedad privada explícitamente antes  
  de usarla.  
  constructor(nom) {  
    this.#nombre = nom;  
  }  
}
```

```
const docenteSAI = new Docente("Carmelo"); // { nombre: "Carmelo" }
```

```
// error (no se puede acceder a la propiedad privada)  
// Uncaught SyntaxError: Private field '#nombre' must be declared in an  
enclosing class
```

```
console.log(docenteSAI.#nombre);
```

RESUMEN



Recuerda que si quieres evitar que estas propiedades de clase se puedan modificar desde fuera de la clase, añade el `#` antes del nombre de la propiedad al declararla. De esta forma serán propiedades privadas, y sólo se podrá acceder a ellas desde el interior de los métodos de la clase (por ejemplo, mediante los métodos *get* y *set*).

ANALOGÍA PARA LOS MÉTODOS

Para el caso de los métodos ocurre exactamente igual. En el caso de definir el método privado, no es posible ejecutarlo desde fuera de una clase, salvo que lo hagamos a través de un método público que llame internamente al método privado.