Creación de objetos

class

class

```
// Declaración de una clase

class Animal {}
// Crear o instanciar un objeto

Let/var/const pato = new Animal();
```

Métodos

```
class Animal {
// Métodos
 hablar() { return "Cuak"; }
 }
```

Se puede usar parámetros como normalmente los utilizamos.

Métodos estáticos

- Para usar un método de una clase, como por ejemplo hablar(), debemos crear el objeto basado en la clase haciendo un new de la clase. Lo que se denomina crear un objeto o una instancia de la clase.
- En algunos casos, nos puede interesar crear **métodos estáticos** en una clase porque para utilizarlos no hace falta crear ese objeto, sino que se pueden ejecutar directamente sobre la clase directamente.

Ejemplo

```
class Animal {
 static despedirse() { return "Adiós"; }
hablar() { return "Cuak"; } }

Animal.despedirse(); // 'Adiós
```

Constructor

- Se le llama **constructor** a un tipo especial de método de una clase, que se ejecuta automáticamente a la hora de hacer un new de dicha clase.
- Una clase **solo puede tener un constructor**, y en el caso de que no se especifique un constructor a una clase, tendrá uno vacío de forma implícita.

ejemplo

```
class Point {
 constructor ( x = 0, y = 0 ) {
    this.x = x;
    this.y = y;
 }
```

propiedades

```
class Animal {
constructor(n) {
this.nombre = n; }
```

Propiedades computadas

En algunos casos nos puede interesar utilizar lo que se llaman **propiedades computadas**.
Las **propiedades computadas** son un tipo de propiedades a las que queremos realizarle ligeros cambios antes de guardarla o antes de obtenerla.

Propiedades computadas getters y setters

```
class Animal {
  constructor(n) {
    this._nombre = n;
  }

  get nombre() {
    return "Sr. " + this._nombre;
  }

  hablar() {
    return "Cuak";
  }
  quienSoy() {
    return "Hola, soy " + this.nombre;
  }
}

// Creación de objetos
const pato = new Animal("Donald");

pato.nombre; // 'Sr. Donald'
pato.nombre = "Pancracio"; // 'Pancracio'
pato.nombre; // 'Sr. Donald'
```

las propiedades con `get` no se pueden cambiar, son de sólo lectura.

ejemplo

```
class Animal {
  constructor(n) {
    this.nombre = n;
  }

  get nombre() {
    return "Sr. " + this._nombre;
  }

  set nombre(n) {
    this._nombre = n.trim();
  }

  hablar() {
    return "Cuak";
  }
  quienSoy() {
    return "Hola, soy " + this.nombre;
  }
}

// Creación de objetos
const pato = new Animal("Donald");

pato.nombre; // 'Sr. Donald'
pato.nombre = "   Lucas  "; // '   Lucas  '
pato.nombre; // 'Sr. Lucas'
```

1-hacer el ejemplo alumnos anterior con class.
Tener en cuenta que el nombre es una propiedad computada de solo lectura, y lo que devuelve es el nombre siempre en mayúsculas.