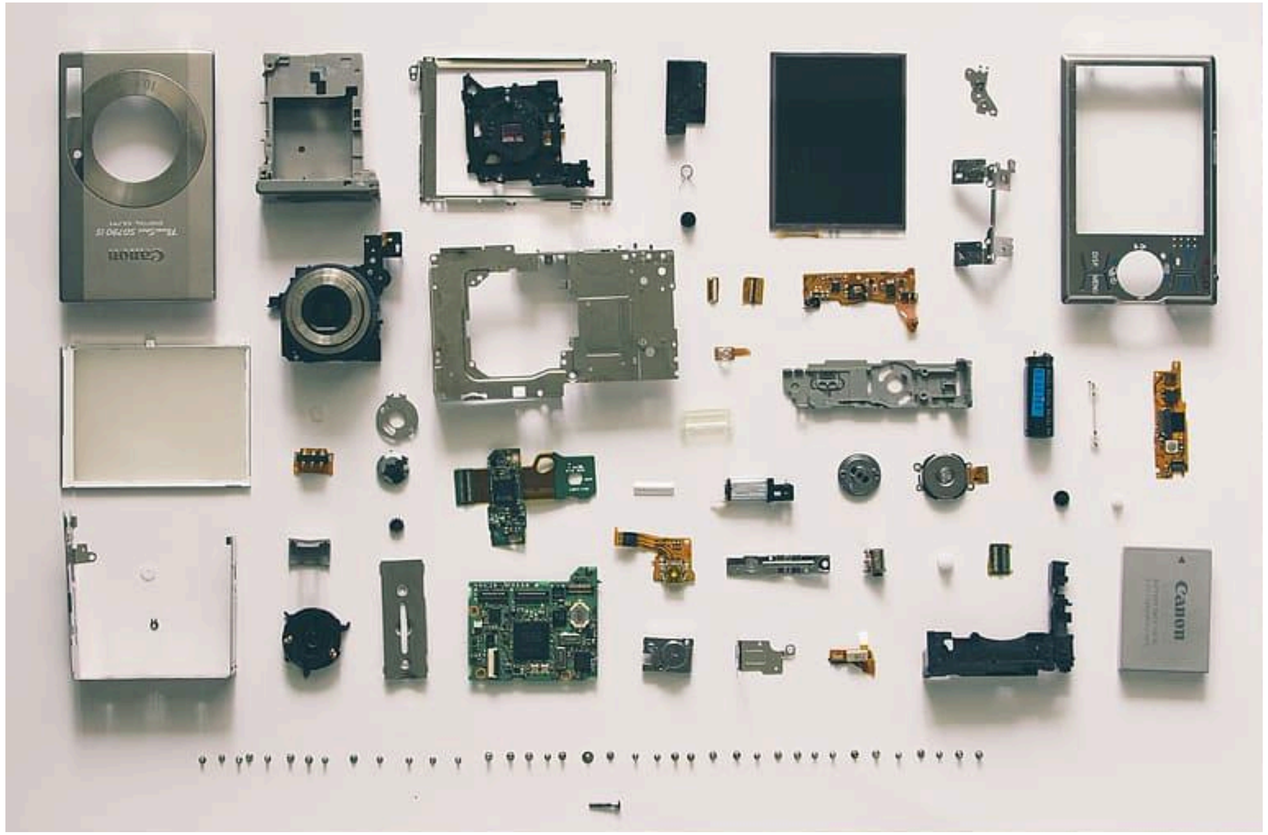
MÓDULOS EN JS

→ Fragmento de código reusable que encapsula los detalles de la implementación

→ Usualmente es un archivo por separado (no necesariamente)

MÓDULOS EN JS



→ **Legos:** Los módulos se comportan como los bloques de construcción del código.

→ **Aislamiento de los componentes:** Los módulos se desarrollan en aislamiento sin tener que pensar en el resto del código.

→ **Abstracción del código:** El código requerido para cada uno de los módulos se tiene por separado y solo se importa la abstracción a otros módulos.

→ **Organización del código:** Los módulos naturalmente ayudan a la organización del código.

→ **Reutilización del código:** Los módulos permiten reutilizar el mismo código (hasta en diferentes proyectos).

MÓDULOS ES06 VS SCRIPTS

	Módulos ES06	Scripts
Variables globales	scope de módulo	global
Modo por defecto	strict mode	-
`this` global	`undefined`	`window`
valores del **`import`** y **`export`**	✅	🚫
Link en el HTML	`<script type=”module”>`	`<script>`

CHEAT SHEET

NAME EXPORT

```
export const variable = 'valor'
```

NAME IMPORT

```
import { variable } from '...'
```

DEFAULT EXPORT

```
export default 'valor'
```

DEFAULT IMPORT

```
import elNombreQueQuiera from '...'
```

RENAME EXPORT

```
export { variable as nuevoNombre }
```

NAME IMPORT

```
import { nuevoNombre } from '...'
```

NAME MULTIPLE EXPORTS

```
export {
  variable1,
  variable2,
}
```

NAME MULTIPLE IMPORTS

```
import {
  variable1,
  variable2,
} from '...'
```

EXPORT LIST + RENAME

```
export {
  variable1,
  variable2 as nuevaVariable2,
}
```

IMPORT LIST + RENAME

```
import {
  variable1 as nuevaVariable1,
  nuevaVariable2,
} from '...'
```

EVOLUCIÓN DE LAS PRÁCTICAS EN JAVASCRIPT

1

Programación Procedimental

Enfoque tradicional con un flujo de ejecución lineal, que a menudo conduce a código complejo y difícil de mantener.

2

Programación Orientada a Objetos

Se centra en los objetos y sus interacciones, introduciendo conceptos como clases y herencia.

3

Desarrollo Modular

Fomenta la división del código en módulos reutilizables, promoviendo la organización y reutilización del código.

MÓDULOS EN JS

1

Encapsulamiento

Los módulos ocultan los detalles de implementación interna, exponiendo solo las funcionalidades necesarias.

2

Reusabilidad

Los módulos pueden utilizarse en múltiples proyectos, promoviendo la eficiencia del código y reduciendo la redundancia (DRY).

3

Fácil Mantenimiento

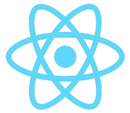
El código modular es más fácil de entender, modificar y debuggear, lo que lleva a proyectos más manejables.

4

Colaboración

Los módulos facilitan el trabajo en equipo al permitir que los desarrolladores trabajen de forma independiente en partes separadas del proyecto.

INTEGRACIÓN DE MÓDULOS Y PAQUETES DE TERCEROS



React

Una popular biblioteca de JavaScript para construir interfaces de usuario, conocida por su arquitectura basada en componentes.

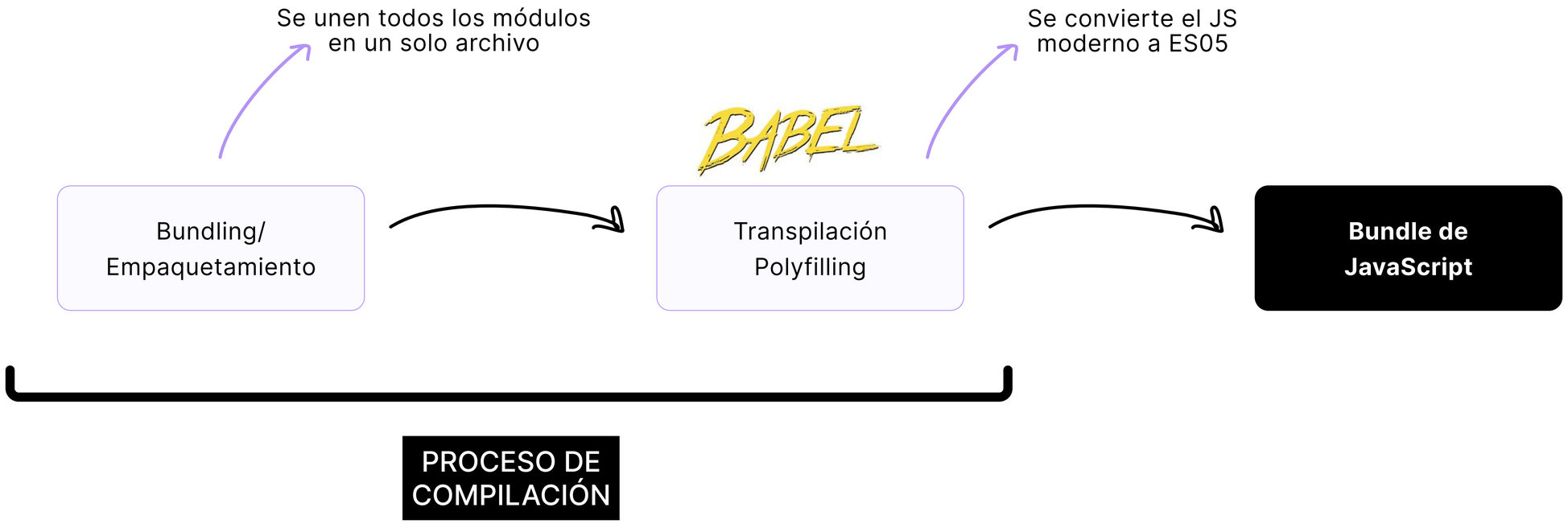
jQuery

Una potente biblioteca de JavaScript que simplifica la manipulación del DOM, el manejo de eventos y las operaciones AJAX.

Leaflet

Una biblioteca de JavaScript ligera diseñada para crear mapas interactivos, permitiendo a los usuarios mostrar datos geográficos de manera visual.

PROCESO DE COMPILACIÓN DE JAVASCRIPT



webpack

PARCEL