

Undefined Academy

Bootcamp de JavaScript Full-stack

Semana 4, Clase 1:

Biblioteca de utilidades



Metodología de trabajo

Formato de las sesiones (1h 30m)

- 10 min Revisión de ejercicio anterior
- 60 min Presentación teórica-práctica
- 10 min Preguntas y discusión

Herramientas del Bootcamp

- **Twitch:** <https://undf.sh/en-vivo>
- **Discord:** <https://undf.sh/discord>
- **Notion:** <https://undf.sh/base>

Agenda

- Introducción a Lodash
- Práctica con Lodash

Revisión de ejercicio anterior



Introducción a Lodash

¿Qué es Lodash?

Lodash es una biblioteca de JavaScript que proporciona funciones utilitarias para tareas comunes de programación. Está diseñada para trabajar con estructuras de datos como objetos, arreglos, cadenas de texto (strings) y números.

Existen alternativas como: Underscore, Ramda y métodos nativos¹.

1. <https://github.com/you-dont-need/You-Dont-Need-Lodash-Underscore>



JS utils.js

```
map([🌽, 🐮, 🐔], cook)  
⇒ [🍿, 🍔, 🍳]
```

```
filter([🍿, 🍔, 🍳], isVegetarian)  
⇒ [🍿, 🍳]
```

```
reduce([🍿, 🍳], eat)  
⇒ 🍌
```

Práctica con Lodash

Práctica con Lodash

1. **groupBy()**: Permite agrupar las entradas del blog por autor o por fecha, por ejemplo.

```
_.groupBy(entries, 'autor').
```

2. **filter()**: Permite filtrar las entradas del blog que cumplan ciertos criterios, como las que tengan ciertos tags específicos.

```
_.filter(entries, entry =>  
_.includes(entry.tags, 'CSS3')).
```

3. **map()**: Permite transformar los datos del set, por ejemplo, para obtener un array solo con los títulos de las entradas.

```
_.map(entries, 'titulo').
```

4. **orderBy()**: te permite ordenar las entradas por fecha o por algún otro criterio.

```
_.orderBy(entries, ['fecha', 'titulo'], ['desc', 'asc']).
```

5. **reduce()**: te permite realizar alguna operación con los datos, como calcular la cantidad total de tags en todas las entradas.

```
_.reduce(entries, (totalTags, entry) =>  
totalTags + entry.tags.length, 0).
```

Práctica con Lodash

1. **merge()**: puedes combinar dos o más entradas del blog en un solo objeto, por ejemplo:

```
_.merge(entries[0], entries[1]).
```

2. **pick()**: puedes seleccionar ciertas propiedades de una entrada del blog y crear un nuevo objeto con ellas. Por ejemplo, para obtener un objeto solo con el título y los tags de la primera entrada: `_.pick(entries[0], ['titulo', 'tags'])`.

3. **omit()**: puedes eliminar ciertas propiedades de una entrada del blog y crear un nuevo objeto sin ellas. Por ejemplo, para eliminar la descripción de la segunda entrada:

```
_.omit(entries[1], ['descripcion']).
```

4. **has()**: puedes verificar si una entrada del blog tiene una propiedad específica, por ejemplo: `_.has(entries[2], 'fecha')`.

5. **invert()**: Puedes intercambiar las claves y los valores de las propiedades de una entrada del blog. `_.invert(_.mapValues(_.groupBy(entries, 'autor'), entradas => _.first(_.map(entradas, 'titulo'))));`

Recursos

- <https://lodash.com/>
- <https://ramdajs.com/>
- <https://github.com/you-dont-need/You-Dont-Need-Lodash-Underscore>

Ejercicio

1. Crear un nuevo proyecto `npm init -y` e instala `lodash` como dependencia: `npm install lodash`.
2. Recuerda agregar `"type": "module"` al `package.json`
3. Crea un archivo `index.js` y copia las entradas del repositorio `https://github.com/glrodasz/undefined-academy/blob/main/biblioteca-utilidades/index.js#L3-L60`
4. Utiliza los métodos de **Lodash** para *Filtrar* las entradas que tenga el tag de `Desarrollo`, *Mapea* por título y *Agrupar* por autor.

¿Preguntas?