

For each

Concepto general

Vamos a hablar del concepto general ya que el foreach es un ciclo que tienen en común todos los lenguajes modernos.

- Se lo conoce comúnmente como foreach
- En javascript se llama for of (pero lo seguiremos llamando for each).

Usos: sirve para **recorrer** los **elementos** de un array.

For clásico

Recordemos como era el for clásico:

```
for(let i = 0; i < 5; i++){  
  console.log( i );  
}
```

Dentro de los paréntesis creo una variable que va contando desde cero hasta cuatro

Esto me sirve, por ejemplo, si quiero mostrar la tabla del 2:

```
for(let i = 1; i <= 10; i++){  
  console.log( i*2 );  
}
```

Hago uso de la variable para mostrar los resultados

| |
|----|
| 2 |
| 4 |
| 6 |
| 8 |
| 10 |
| 12 |
| 14 |
| 16 |
| 18 |
| 20 |
| |

También nos sirve para recorrer los elementos de un array.

Como hacerlo:

```
const nombres = ["Franco", "Oscar", "Agus", "Owen"];

for(let i = 0; i < nombres.length; i++){
  let nombre = nombres[i];
  console.log(nombre);
}
```

Agarro el elemento y lo guardo en la variable *nombre*

Uso la variable que contiene el nombre obtenido



| |
|--------|
| Franco |
| Oscar |
| Agus |
| Owen |
| |

Hasta acá todo bien.

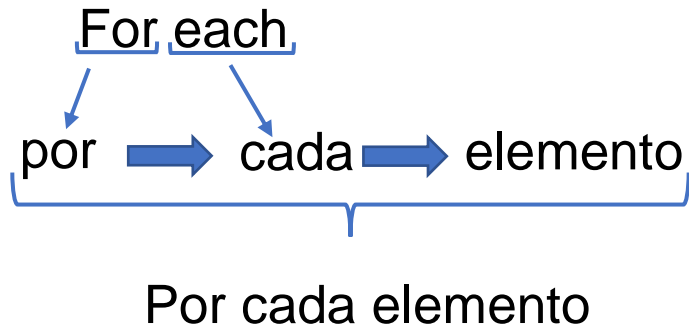
Si queremos recorrer los elementos de un array, podemos usar el for clásico.

O podemos usar el foreach (por cada).

Foreach

Queremos recorrer cada elemento.

El foreach fue creado con ese único propósito: recorrer cada elemento de un array.



For clásico



Repite código por i veces

For each



Repite código por cada elemento

Si el array tiene 5 elementos, el for se repite 5 veces.

Si tiene 8, se repite 8 veces

Sintaxis del foreach

```
const nombres = ["Franco", "Oscar", "Agus", "Owen"];  
for(const nombre of nombres){  
  console.log(nombre);  
}
```

Usamos el objeto que creamos dentro del paréntesis, que guarda justamente el nombre obtenido

Se crea un objeto nombre

Of -> operador que lo que hace es agarrar cada elemento en cada ejecución automáticamente (en la primera ejecución agarra el primer elemento, en la segunda ejecución agarra el segundo elemento y así sucesivamente).

Nombres -> es el array del cual se toman los elementos

¿Cómo se leen los paréntesis?

