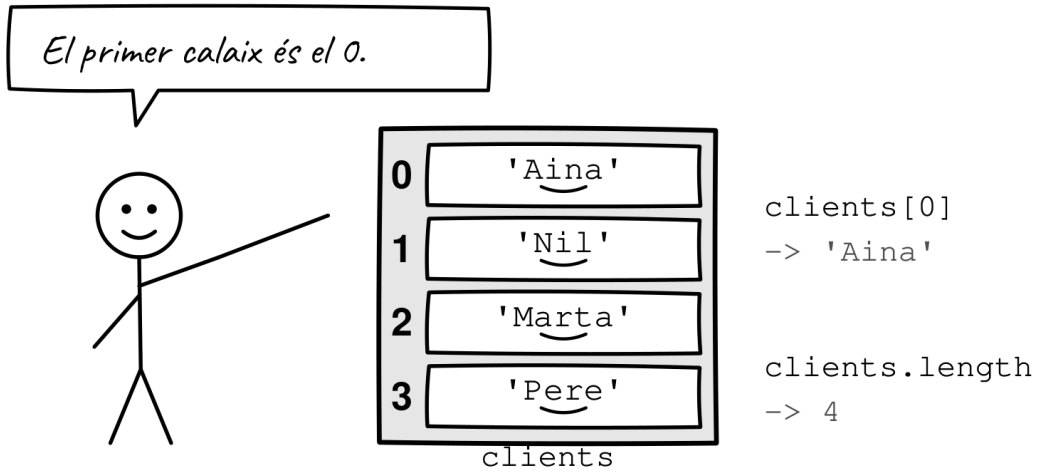
TEMA 3

Arrays i funcions

De la llista a la pàgina



Què farem

1. Guardar **llistes** de dades en arrays.
2. Recórrer-les amb `for...of`.
3. Agrupar codi en **funcions**.
4. Fer **fitxes** de clients amb objectes.
5. `filter`, `map`, `find`, `reduce`.
6. Pintar la llista **a la pàgina**.

Sense classes: els objectes, aquí, són fitxes de dades.

Els índexs comencen pel 0



L'últim índex sempre és length - 1.

Canviar un array

EL CODI

push i pop

```
const compra = ['pa', 'llet'];
compra.push('ous');
compra[0] = 'pa integral';
const darrer = compra.pop();
```

EL QUE QUEDA

Pas a pas

```
['pa', 'llet', 'ous']
['pa integral', 'llet', 'ous']
['pa integral', 'llet']
darrer: 'ous'
```

Un array const es pot canviar per dins. El que no pots és reassignar-lo.

PREGUNTA PER A LA CLASSE

«for (let i = 0; i <= noms.length; i++)»

Què passa a l'última volta?

Recórrer un array

AMB ÍNDEX

`for`

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

SENSE ÍNDEX

`for...of`

```
for (const nota of notes) {
  console.log(nota);
}
```

Si no et cal l'índex, for...of: és més curt i no t'equivoques amb el <.

Buscar el màxim

```
const t = [18, 23, 21, 25, 19];
let maxima = t[0];
for (const x of t) {
  if (x > maxima) {
    maxima = x;
  }
}
```

X	MAXIMA
18	18
23	**23**
21	23
25	**25**
19	25

Comences amb el primer i el canvies quan en trobes un de més gran.

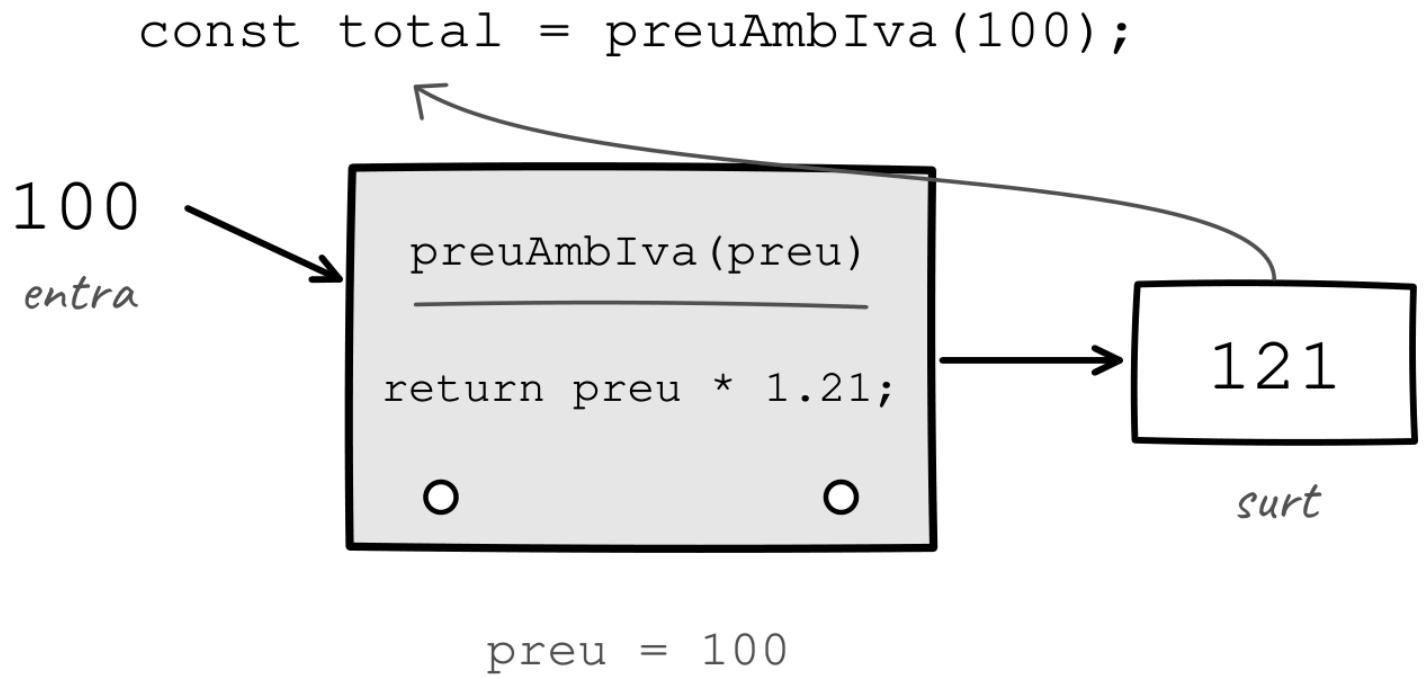
Funcions (functions)

```
function saluda(nom) {
  console.log(`Hola, ${nom}!`);
}

saluda('Aina');
saluda('Nil');
```

`function`	**defineix**: encara no fa res
`nom`	el **paràmetre**
`saluda('Aina')`	la **crida**: ara s'executa

`return`: la funció et torna un valor



Sense return, la funció torna undefined. console.log només mostra.

PREGUNTA PER A LA CLASSE

«function suma(a, b) { console.log(a + b); }»

Quant val const r = suma(2, 3)?

Funcions fletxa (arrow functions)

LA DE SEMPRE

```
function doble(n) {
  return n * 2;
}
```

LA MATEIXA, EN FLETXA

```
const doble = (n) => n * 2;
```

Una sola expressió: sense claus i sense return.

Objectes: fitxes de dades

```
const client = { nom: 'Aina', ciutat: 'Palma', saldo: 120 };
```

ESCRIUS	DONA
`client.nom`	`'Aina'`
`client.saldo + 30`	`150`
`client.saldo = 200`	canvia el saldo

Clau: valor. Es llegeix amb un punt.

Arrays d'objectes

```
const clients = [
  { nom: 'Aina', ciutat: 'Palma', saldo: 120 },
  { nom: 'Nil', ciutat: 'Inca', saldo: -40 },
  { nom: 'Marta', ciutat: 'Palma', saldo: 75 },
];
```

```
for (const client of clients) {
  console.log(client.nom);
}
```

```
clients[3].nom
TypeError: Cannot read
properties of undefined
```

Filtrar, transformar, buscar, sumar

MÈTODE	EXEMPLE	TORNA
`filter`	`(c) => c.saldo < 0`	els deutors (array)
`map`	`(c) => c.nom`	els noms (array)
`find`	`(c) => c.nom === 'Marta'`	la Marta (un objecte)
`reduce`	`(suma, c) => suma + c.saldo, 0`	155 (un valor)

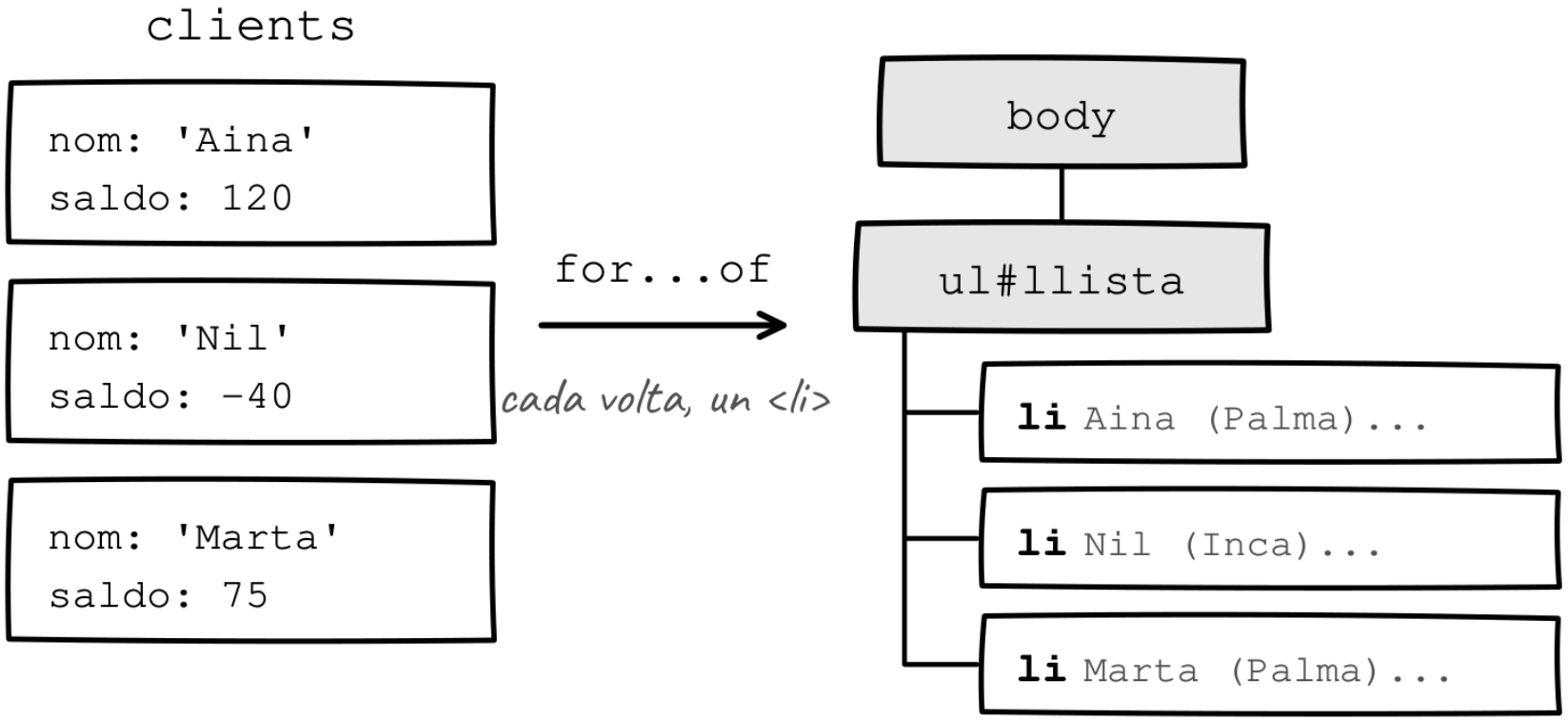
Cap d'aquests mètodes canvia l'array original.

PREGUNTA PER A LA CLASSE

«Vull els productes de menys de 20 €.»

filter, map, find o reduce?

De la llista a la pàgina



Els <li> no són a l'index.html: els posa el bucle.

Tres ordres del DOM

```
const llista = document.getElementById('llista');
for (const client of clients) {
  const element = document.createElement('li');
  element.textContent = client.nom;
  llista.appendChild(element);
}
```

ORDRE	QUÈ FA
`getElementById`	busca l'element
`createElement`	en crea un de nou, **encara invisible**
`appendChild`	el penja a l'arbre: **ara es veu**

Una funció per pintar

```
function pintaClient(client) {
  const element = document.createElement('li');
  element.textContent = `${client.nom}: ${client.saldo} €`;
  if (client.saldo < 0) {
    element.textContent = element.textContent + ' (deutor)';
  }
  llista.appendChild(element);
}

for (const client of clients) {
  pintaClient(client);
}
```

Si un tros de codi fa **una feina**, posa-li **nom**.

El que has de saber sí o sí

[0]	el primer; l'últim és length - 1.
for...of	per recórrer sense índex.
return	torna el valor; sense, undefined.
client.nom	un objecte és una fitxa de dades.
filter / map	triar uns quants / transformar-los tots.
appendChild	fins aquí, l'element no es veu.