Arrays

Els arrays incorporen una sèrie de mètodes molt útils que es combinen a la perfecció amb els arrow funcions.

Podeu trobar el llistat complet a: Array methods and empty slots

Els més interessants podrien ser:

Foreach

Simplificació del típic bucle for on executarà una acció per a cada element del array:

```
array1.forEach((element) => {
   console.log(element)
});
```

Map

Si volem convertir un array a una d'un altre tipus podem usar el mètode map.

```
// Suposem que volem obtenir un array de totes les persones d'un llistat de
sol·licituds
const persones = solicituts.map((element) => {
   return element.persona;
});
// El nou array persones contendria les persones de les sol·licituds
```

Reduce

En aquest cas ens permet reduir una array a un valor.

```
// Volem obtenir el total econòmic d'un llistat d'ajudes
const valorInicial = 0;
const total = ajudes.reduce((acumulat, element) => {
      return acumulat + element.quantitatAjuda;
   },
   valorInicial // el valor inicial es podria definir directament, és el
valor que tedrà acumulat en el primer element
);
```

Every / some

Semblant a reduce però en aquest cas ens verifica si tots els elements de l'array compleixen una condició.

```
// Volem saber si totes les persones d'un array son majors de 65 anys
const totsMajors = persones.every((element) => {
   return element.edat > 65;
});
// Si ho son totsMajors serà true, sinó serà false
```

El cas de some és igual però ens indicarà si algun element compleix la condició

```
// Volem saber si hi ha algun major de 65 anys
const algunMajor = persones.some((element) => {
   return element.edat > 65;
});
// Si hi ha algun major sera true
```

Sort

El mateix nom indica que ordenarà un llistat en funció de la condició que designem

```
// Volem ordenar un llistat de persones per edat
persones.sort((a, b) => {
   return a.edat > b.edat
});
```