JavaScript - 04

Array.prototype.reduce()

Syntax

```
array.reduce((accumulator, currentValue, index, array) => {
  // return newAccumulator
}, initialValue);
```

Sum

```
const numbers = [1, 2, 3, 4];
const sum = numbers.reduce((acc, n) => acc + n, 0); // 10
```

Multiplication

```
const product = numbers.reduce((acc, n) => acc * n, 1); // 24
```

Counting occurrences of elements

```
const fruits = ["apple", "banana", "apple", "orange", "banana",
"apple"];
const count = fruits.reduce((acc, fruit) => {
  acc[fruit] = (acc[fruit] || 0) + 1;
  return acc;
}, {});
// Resultado: { apple: 3, banana: 2, orange: 1 }
```

Flatten

```
const arr = [[1, 2], [3, 4], [5]];
const flat = arr.reduce((acc, curr) => acc.concat(curr), []);
// Resultado: [1, 2, 3, 4, 5]
```

Get the maximum value

```
const max = numbers.reduce((acc, n) => n > acc
? n
: acc, numbers[0]); // 4
```

Group objects by property

```
const people = [
  { name: "Ana", age: 20 },
  { name: "Luis", age: 21 },
  { name: "Juan", age: 20 }
];

const grouped = people.reduce((acc, person) => {
  const key = person.age;
  if (!acc[key]) acc[key] = [];
  acc[key].push(person.name);
  return acc;
}, {});
// Resultado: { 20: ["Ana", "Juan"], 21: ["Luis"] }
```