Q: Group objects by a property

Input:

```
const items = [
  { name: "Apple", category: "Fruit" },
  { name: "Carrot", category: "Vegetable" },
  { name: "Banana", category: "Fruit" },
  { name: "Broccoli", category: "Vegetable" }
];
```

Expected Output:

```
{
  Fruit: [
    { name: "Apple", category: "Fruit" },
    { name: "Banana", category: "Fruit" }
  ],
  Vegetable: [
    { name: "Carrot", category: "Vegetable" },
    { name: "Broccoli", category: "Vegetable" }
  ]
}
```

Solution:

```
const grouped = items.reduce((acc, item) => {
  acc[item.category] = acc[item.category] || [];
  acc[item.category].push(item);
  return acc;
}, {});
```

Q: Get unique values of a key

Input:

```
const employees = [
  { name: "John", dept: "HR" },
  { name: "Alice", dept: "IT" },
  { name: "Bob", dept: "HR" },
  { name: "Jane", dept: "Finance" }
];
```

Expected Output:

```
["HR", "IT", "Finance"]
```

Solution:

```
const uniqueDepts = [...new Set(employees.map(emp => emp.dept))];
```

Q: Convert array to object keyed by ID

Input:

```
const users = [
  { id: 101, name: "John" },
  { id: 102, name: "Alice" }
];
```

Expected Output:

```
{
  101: { id: 101, name: "John" },
  102: { id: 102, name: "Alice" }
}
```

Solution:

```
const userMap = users.reduce((acc, user) => {
  acc[user.id] = user;
  return acc;
}, {});
```

Q: Count occurrences by property

Input:

```
const employees = [
  { name: "John", dept: "HR" },
  { name: "Alice", dept: "IT" },
  { name: "Bob", dept: "HR" },
  { name: "Jane", dept: "Finance" }
];
```

Expected Output:

```
{ HR: 2, IT: 1, Finance: 1 }
```

Solution:

```
const countByDept = employees.reduce((acc, emp) => {
  acc[emp.dept] = (acc[emp.dept] || 0) + 1;
  return acc;
}, {});
```

Q: Merge objects with the same key

Input:

```
const scores = [
  { id: 1, score: 10 },
  { id: 2, score: 15 },
  { id: 1, score: 20 }
];
```

Expected Output:

```
[
  { id: 1, score: 30 },
  { id: 2, score: 15 }
]
```

Solution:

```
const merged = Object.values(scores.reduce((acc, item) => {
  if (!acc[item.id]) acc[item.id] = { ...item };
  else acc[item.id].score += item.score;
  return acc;
}, {}));
```

Q: Find the object with the highest value

Input:

```
const employees = [
  { name: "John", salary: 50000 },
  { name: "Alice", salary: 75000 },
  { name: "Bob", salary: 60000 }
];
```

Expected Output:

```
{ name: "Alice", salary: 75000 }
```

Solution:

```
const highest = employees.reduce((max, emp) => emp.salary > max.salary ? emp : max);
```