

Objects of Objects vs Array of Objects

1. Objects of Objects (Nested Objects)

javascript

```
JavaScript
const productsObj = {
  product1: {
    id: 1,
    name: "PHANTOM X",
    price: 349
  },
  product2: {
    id: 2,
    name: "WAVE PRO",
    price: 599
  },
  product3: {
    id: 3,
    name: "PULSE MINI",
    price: 179
  }
};
```

2. Array of Objects

javascript

```
JavaScript
const productsArray = [
  {
    id: 1,
    name: "PHANTOM X",
    price: 349
  },
  {
    id: 2,
```

```

    name: "WAVE PRO",
    price: 599
  },
  {
    id: 3,
    name: "PULSE MINI",
    price: 179
  }
];

```

Key Differences & Advantages

Feature	Array of Objects	Objects of Objects
Order	✓ Maintains order	✗ No guaranteed order
Index Access	✓ <code>products[0]</code>	✗ Need to know key name
Loop Methods	✓ <code>.map()</code> , <code>.filter()</code> , <code>.forEach()</code>	✗ Need <code>Object.keys()</code> first
Direct Lookup	✗ Need to loop/find	✓ <code>productsObj.product1</code>
Count Items	✓ <code>products.length</code>	✗ <code>Object.keys(obj).length</code>
Adding Items	✓ <code>.push()</code> easy	✗ Need unique key names

When to Use Each

Use Array of Objects when:

✓ **Order matters** - like a list of products, tasks, or messages
 ✓ **You need to loop** - displaying items, filtering, sorting
 ✓ **You don't know items in advance** - adding/removing dynamically
 ✓ **Counting is important** - "You have 12 products"
 ✓ **Similar items** - all products, all users, all posts

Example Use Cases:

- Product catalog
- Shopping cart items
- Social media posts
- Todo list
- Search results
- User list

Use Objects of Objects when:

✓ **Fast lookup by key** - like finding user by username ✓ **Unique identifiers** - user IDs, country codes ✓ **Configuration/Settings** - app settings grouped by category ✓ **No need to loop often** - just direct access ✓ **Relationships matter** - parent-child connections

Example Use Cases:

- User profiles by ID: `users[userId]`
- Settings: `config.database.host`
- Translations: `translations['en'].welcome`
- Cache/lookup tables