ES6 Enhanced Object Literals

ES6 introduced several enhancements to object literals, making them more concise and easier to work with. Here's a breakdown of these enhancements:

1. Shorthand Property Names

- If a property name and its value have the same name, you can omit the colon and the value.

JavaScript

```javascript
const name = 'John';
const age = 30;

const person = {
  name,
  age
};

// Equivalent to:
// const person = {
//   name: name,
//   age: age
// };
```

2. Shorthand Method Names

- If a property is a function, you can omit the `function` keyword and the colon.

JavaScript

```javascript
const person = {
  name: 'John',
  greet() {
    console.log(`Hello, my name is ${this.name}`);
  }
};

// Equivalent to:
// const person = {
//   name: 'John',
//   greet: function() {
//     console.log(`Hello, my name is ${this.name}`);
//   }
// };
```

3. Computed Property Names

- Use square brackets to dynamically create property names based on variables or expressions.

  JavaScript

```
const key = 'firstName';
const person = {
  [key]: 'John',
  [`last${key}`]: 'Doe'
};
console.log(person); // Output: { firstName: 'John', lastName: 'Doe' }
```

Benefits of Enhanced Object Literals:

- **Improved Readability:** The shorthand syntax makes object literals more concise and easier to read.
- **Reduced Boilerplate:** Less typing is required, leading to faster development.
- **Increased Flexibility:** Computed property names allow for more dynamic object creation.

In Summary

ES6 enhanced object literals provide a more concise and expressive way to create objects in JavaScript. By utilizing these features, you can write cleaner, more maintainable, and more efficient code.