{ JavaScript }

JavaScript Object Methods

soumyabalamaala@gmail.com

soumyabalamaala

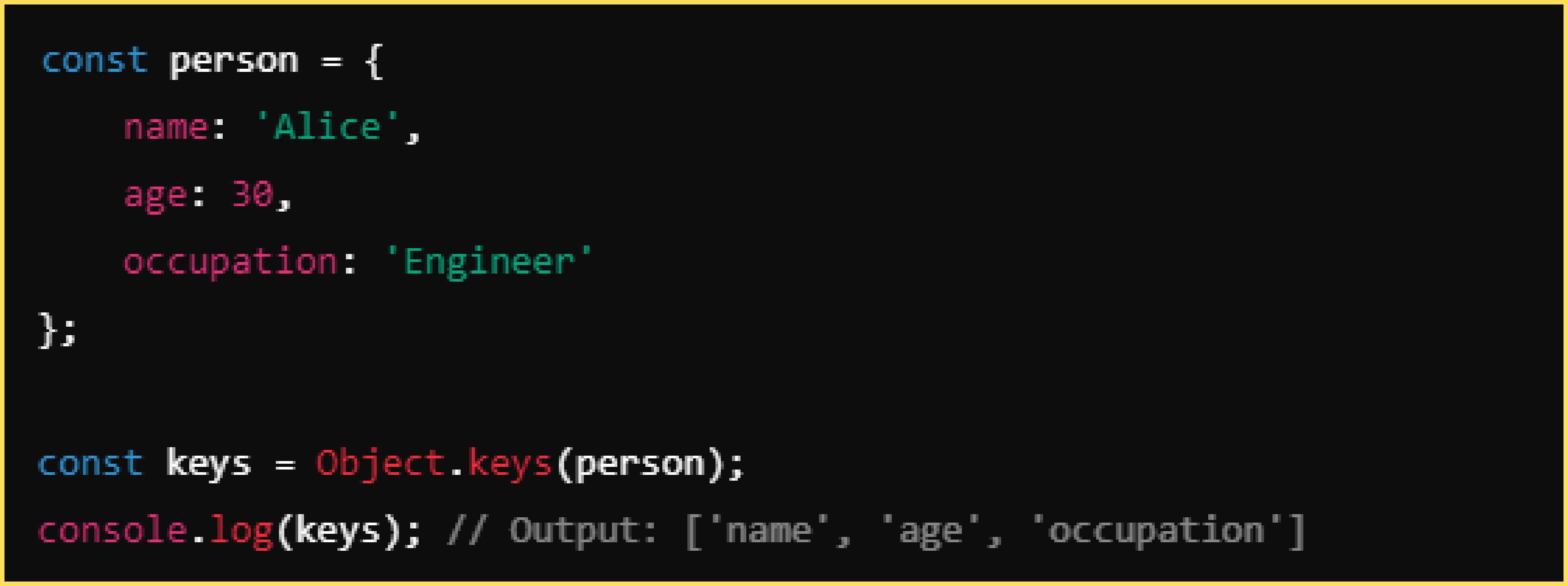
soumyabalamaala

{ JavaScript }

1. Object.keys()

Definition:

Returns an array of a given object’s property names.



```javascript
const person = {
    name: 'Alice',
    age: 30,
    occupation: 'Engineer'
};

const keys = Object.keys(person);
console.log(keys); // Output: ['name', 'age', 'occupation']
```

soumyabalamaala@gmail.com

soumyabalamaala

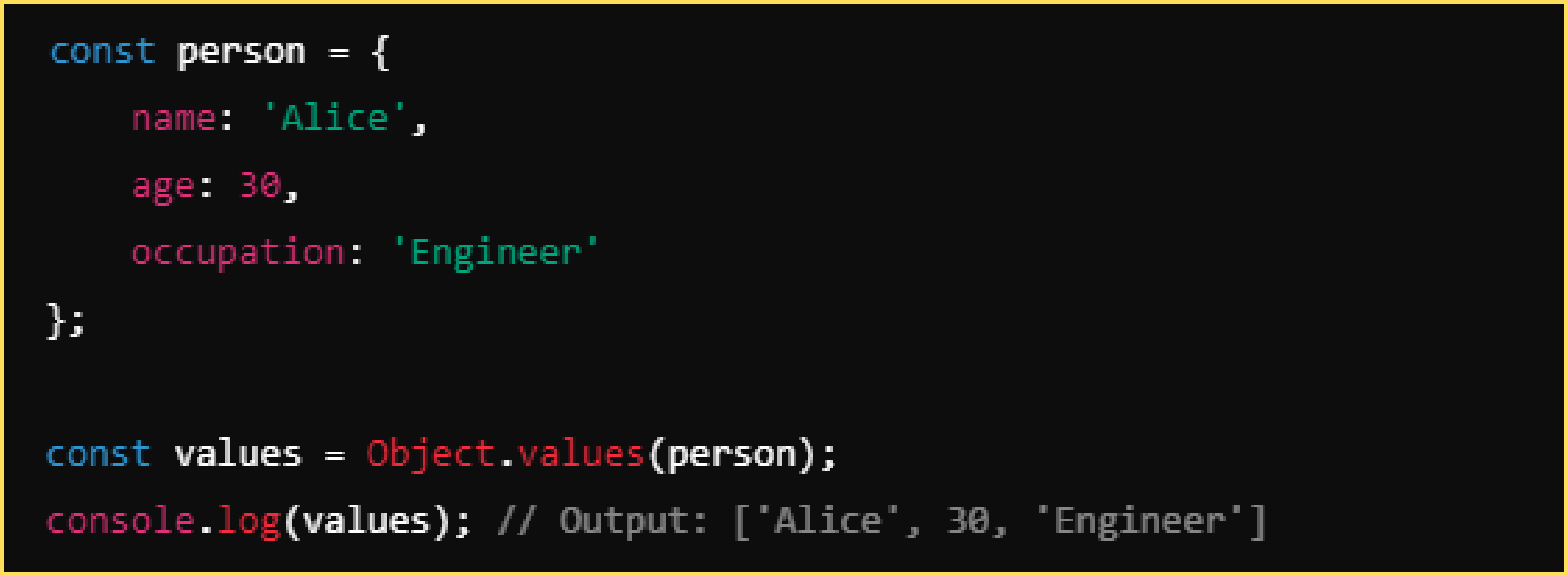
soumyabalamaala

{ JavaScript }

2. Object.values()

Definition:

Returns an array of a given object’s property values.



```
const person = {
    name: 'Alice',
    age: 30,
    occupation: 'Engineer'
};

const values = Object.values(person);
console.log(values); // Output: ['Alice', 30, 'Engineer']
```

soumyabalamaala@gmail.com

soumyabalamaala

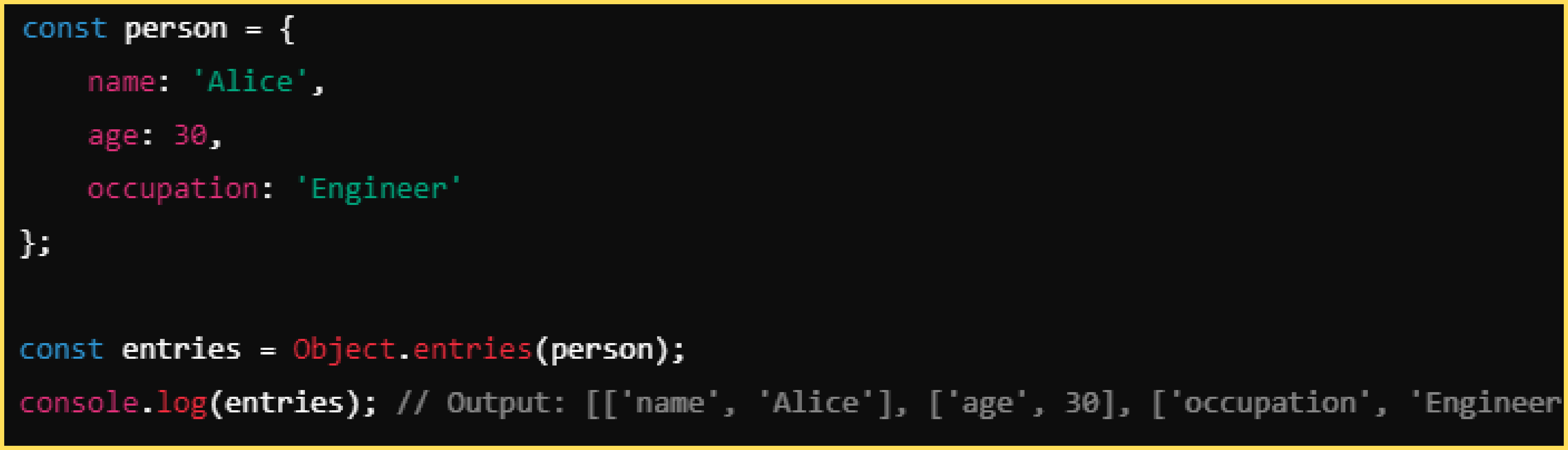
soumyabalamaala

{ JavaScript }

3. Object.entries()

Definition:

Returns an array of a given object’s own enumerable string-keyed property **[key, value]** pairs.



```
const person = {
    name: 'Alice',
    age: 30,
    occupation: 'Engineer'
};

const entries = Object.entries(person);
console.log(entries); // Output: [['name', 'Alice'], ['age', 30], ['occupation', 'Engineer
```

soumyabalamaala@gmail.com

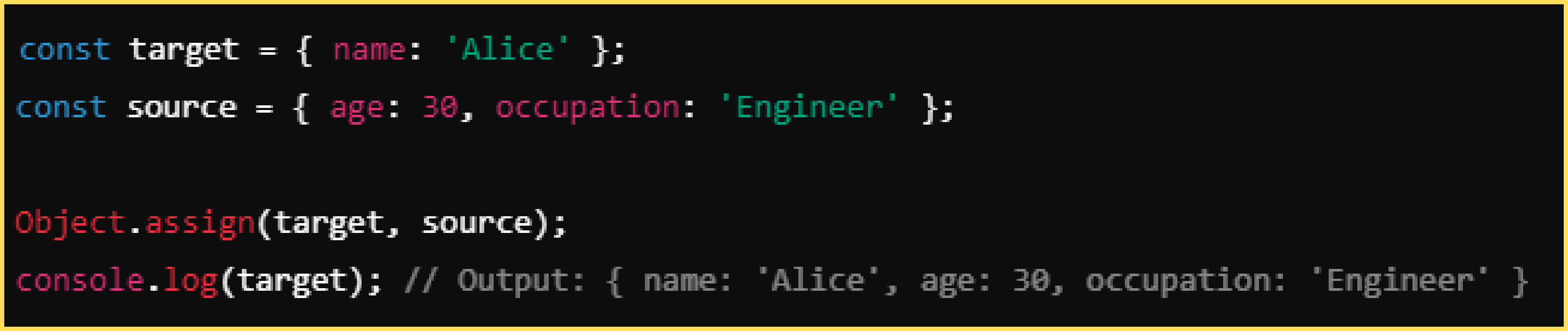
soumyabalamaala soumyabalamaala

{ JavaScript }

4. Object.assign()

Definition:

Copies the values of all enumerable own properties from one or more source objects to a target object. It returns the target object.



```
const target = { name: 'Alice' };
const source = { age: 30, occupation: 'Engineer' };

Object.assign(target, source);
console.log(target); // Output: { name: 'Alice', age: 30, occupation: 'Engineer' }
```

soumyabalamaala@gmail.com

soumyabalamaala

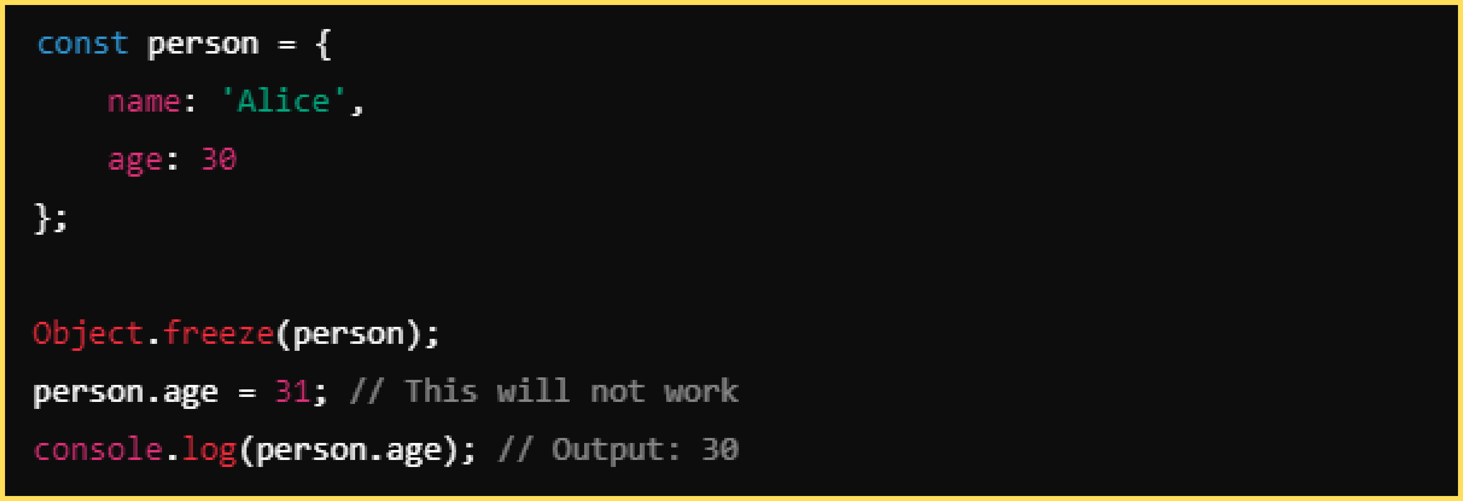
soumyabalamaala

{ JavaScript }

5. Object.freeze()

Definition:

Freezes an object, preventing new properties from being added, existing properties from being removed, or existing properties from being modified.



```
const person = {
    name: 'Alice',
    age: 30
};

Object.freeze(person);
person.age = 31; // This will not work
console.log(person.age); // Output: 30
```

soumyabalamaala@gmail.com

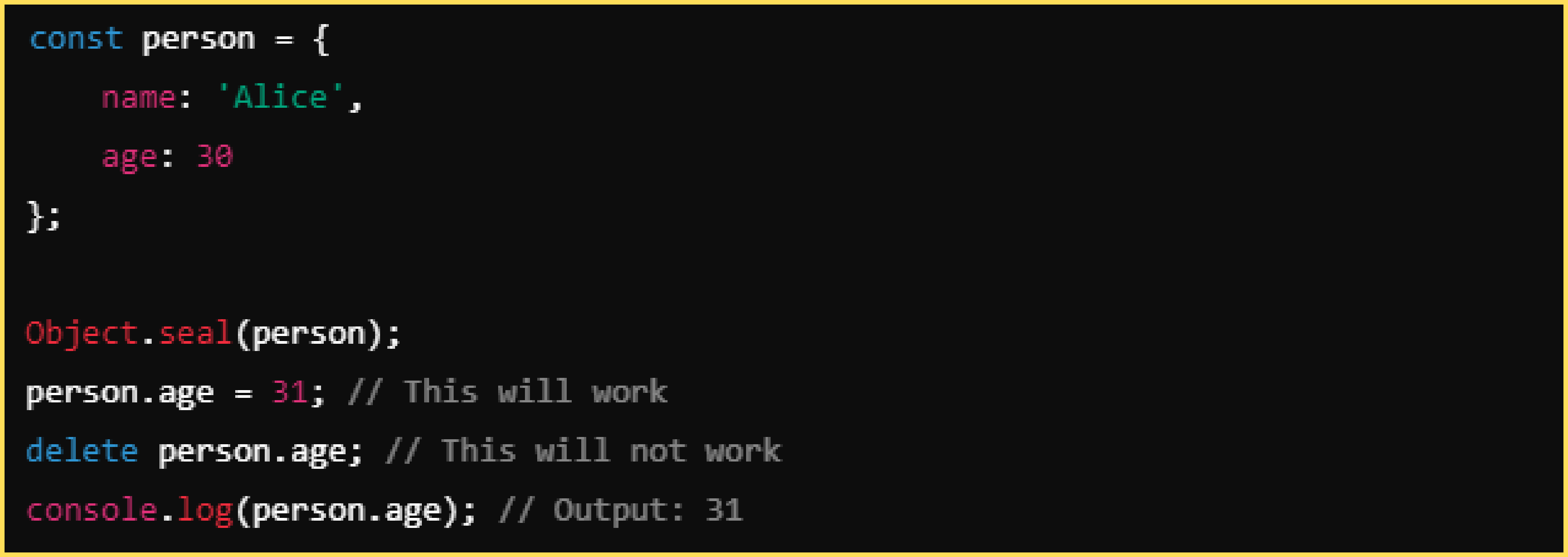
soumyabalamaala   soumyabalamaala

{ JavaScript }

6. Object.seal()

Definition:

Seals an object, preventing new properties from being added and marking all existing properties as non-configurable.



```
const person = {
    name: 'Alice',
    age: 30
};

Object.seal(person);
person.age = 31; // This will work
delete person.age; // This will not work
console.log(person.age); // Output: 31
```

soumyabalamaala@gmail.com

in soumyabalamaala

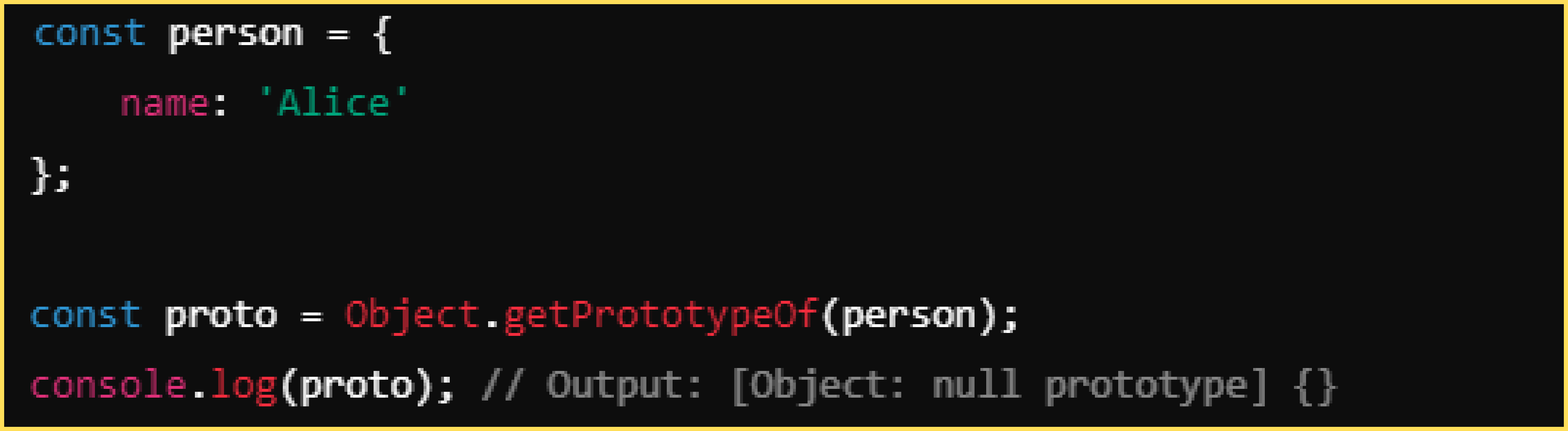
soumyabalamaala

{ JavaScript }

7. Object.getPrototypeOf()

Definition:

Returns the prototype of the specified object.



```
const person = {
    name: 'Alice'
};

const proto = Object.getPrototypeOf(person);
console.log(proto); // Output: [Object: null prototype] {}
```

soumyabalamaala@gmail.com

soumyabalamaala

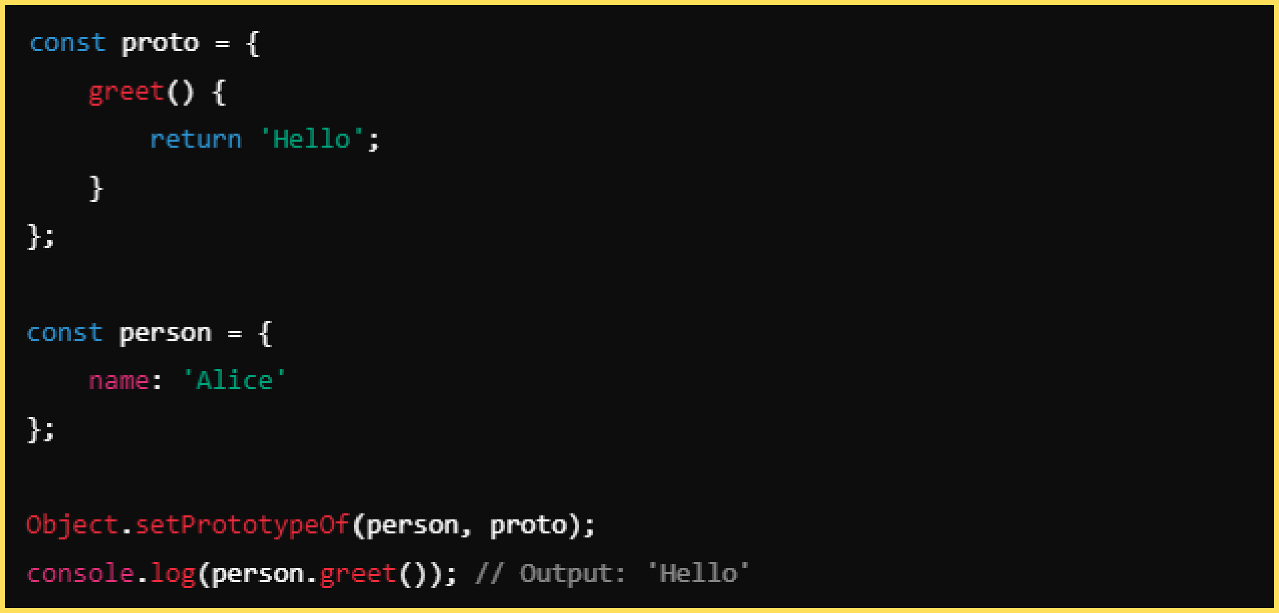
soumyabalamaala

{ JavaScript }

8. Object.setPrototypeOf()

Definition:

Sets the prototype of a specified object to another object or null.



```
const proto = {
    greet() {
        return 'Hello';
    }
};

const person = {
    name: 'Alice'
};

Object.setPrototypeOf(person, proto);
console.log(person.greet()); // Output: 'Hello'
```

soumyabalamaala@gmail.com

soumyabalamaala

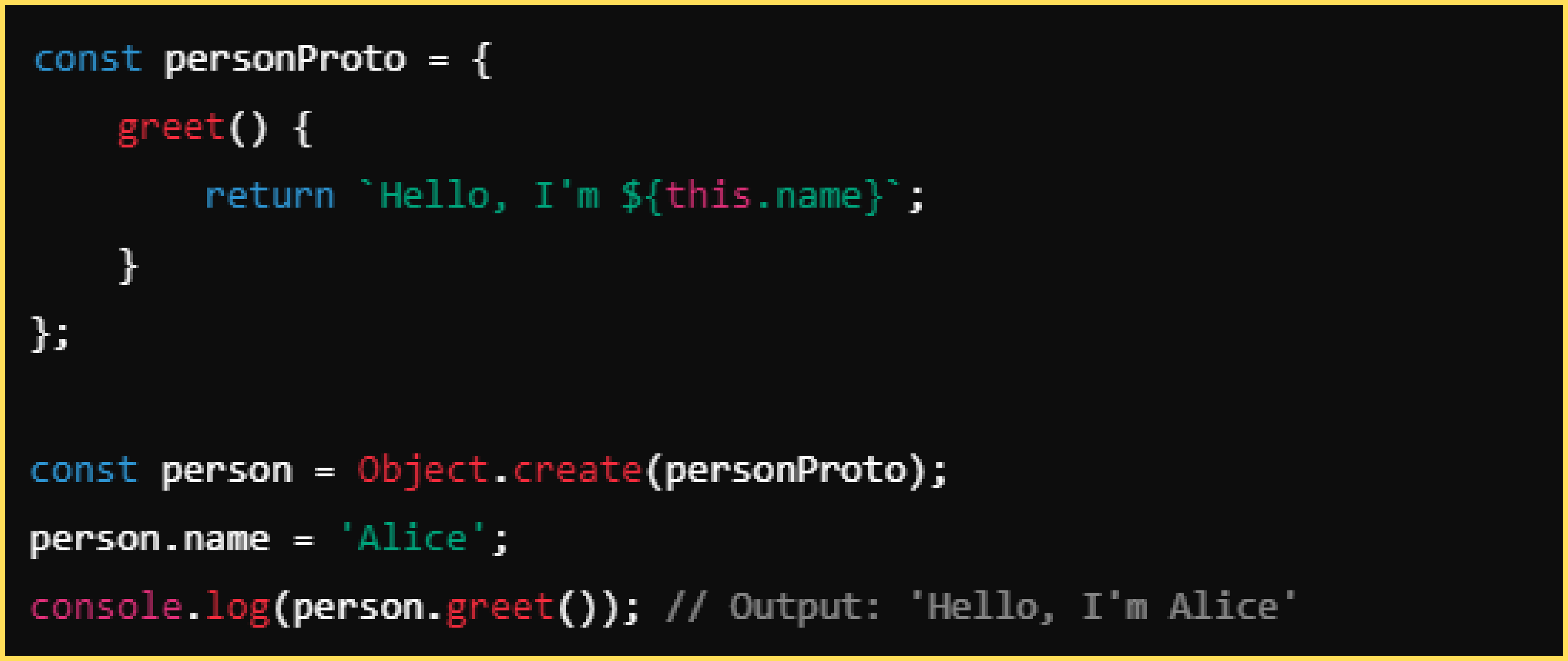
soumyabalamaala

{ JavaScript }

9. Object.create()

Definition:

Creates a new object with the specified prototype object and properties.



```
const personProto = {
    greet() {
        return `Hello, I'm ${this.name}`;
    }
};

const person = Object.create(personProto);
person.name = 'Alice';
console.log(person.greet()); // Output: 'Hello, I'm Alice'
```

soumyabalamaala@gmail.com

soumyabalamaala

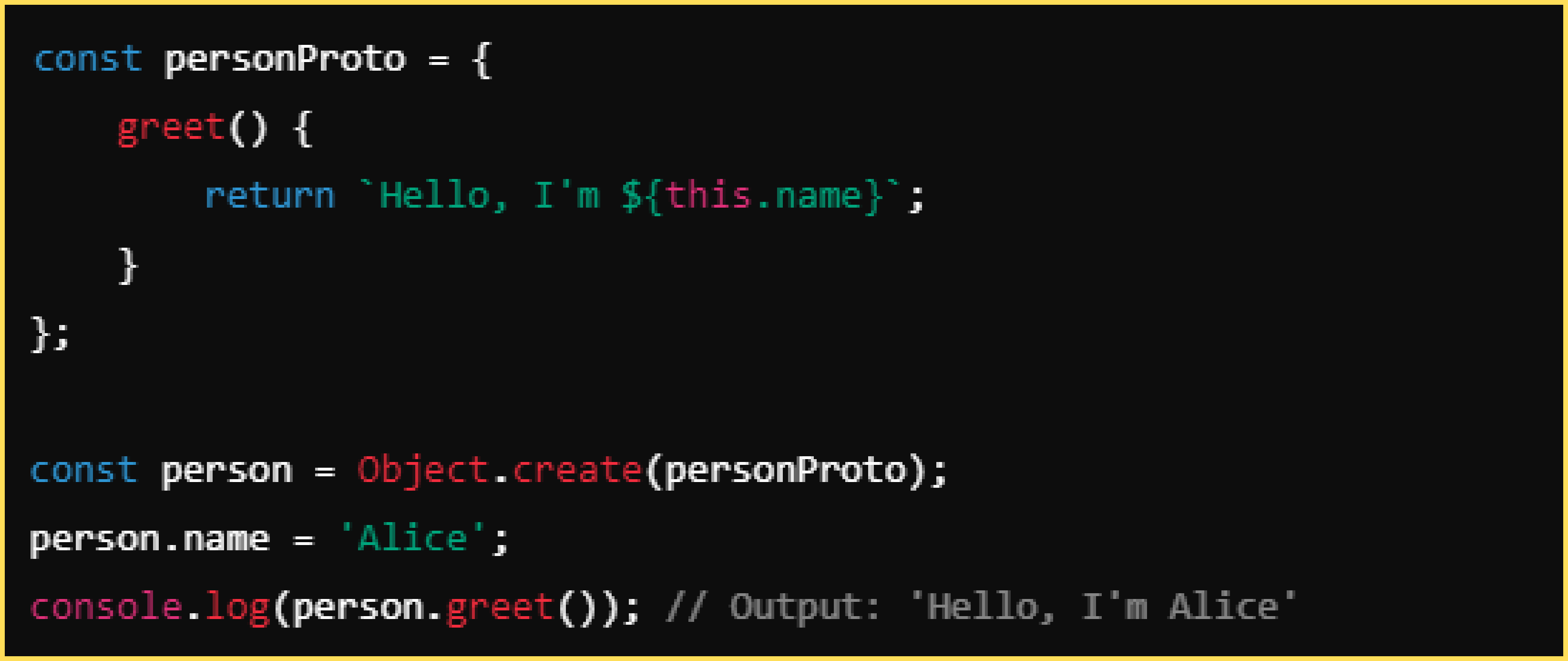
soumyabalamaala

{ JavaScript }

10. Object.create()

Definition:

Creates a new object with the specified prototype object and properties.



```
const personProto = {
    greet() {
        return `Hello, I'm ${this.name}`;
    }
};

const person = Object.create(personProto);
person.name = 'Alice';
console.log(person.greet()); // Output: 'Hello, I'm Alice'
```

soumyabalamaala@gmail.com

soumyabalamaala

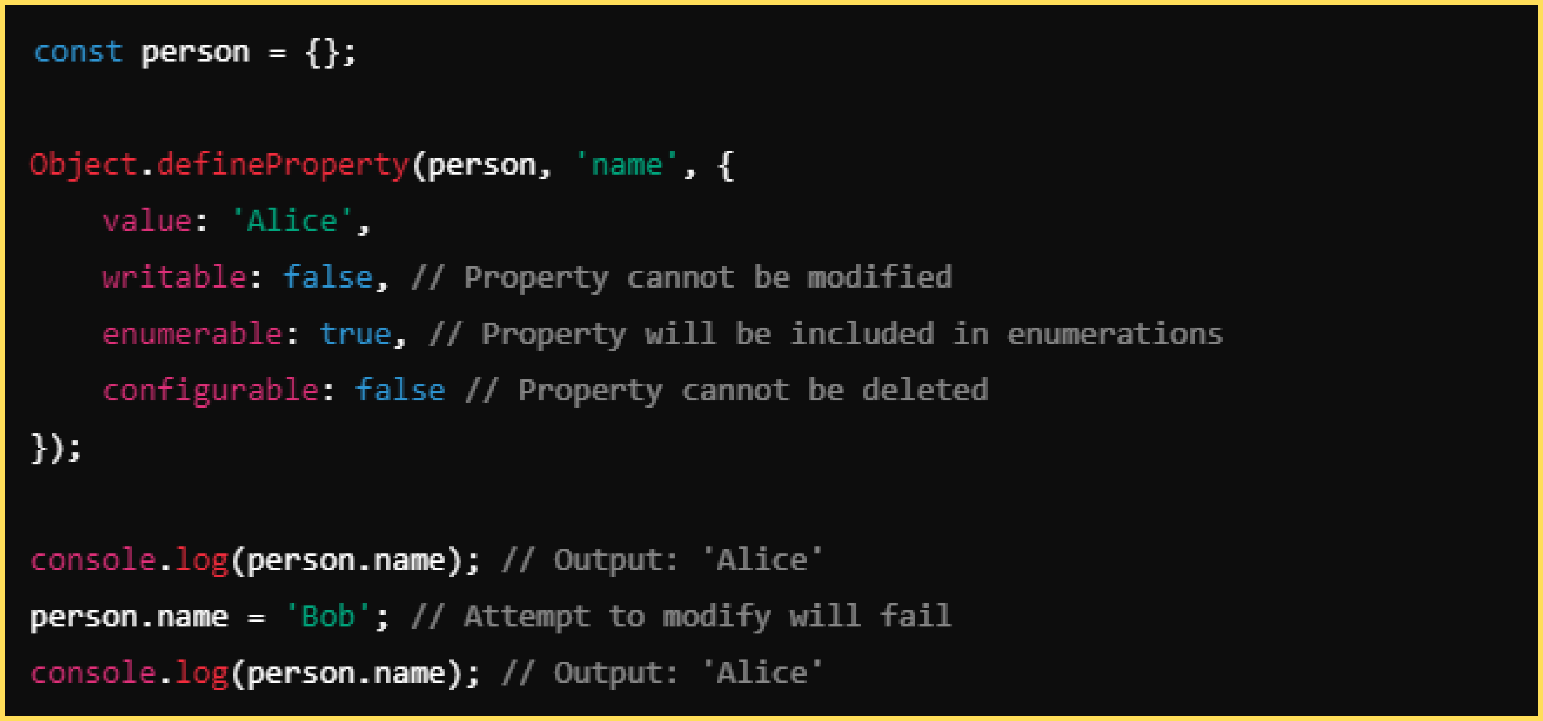
soumyabalamaala

{ JavaScript }

11. Object.defineProperty()

Definition:

Adds a property to an object, or modifies an existing property, and returns the object.



```javascript
const person = {};

Object.defineProperty(person, 'name', {
    value: 'Alice',
    writable: false, // Property cannot be modified
    enumerable: true, // Property will be included in enumerations
    configurable: false // Property cannot be deleted
});

console.log(person.name); // Output: 'Alice'
person.name = 'Bob'; // Attempt to modify will fail
console.log(person.name); // Output: 'Alice'
```

soumyabalamaala@gmail.com

soumyabalamaala

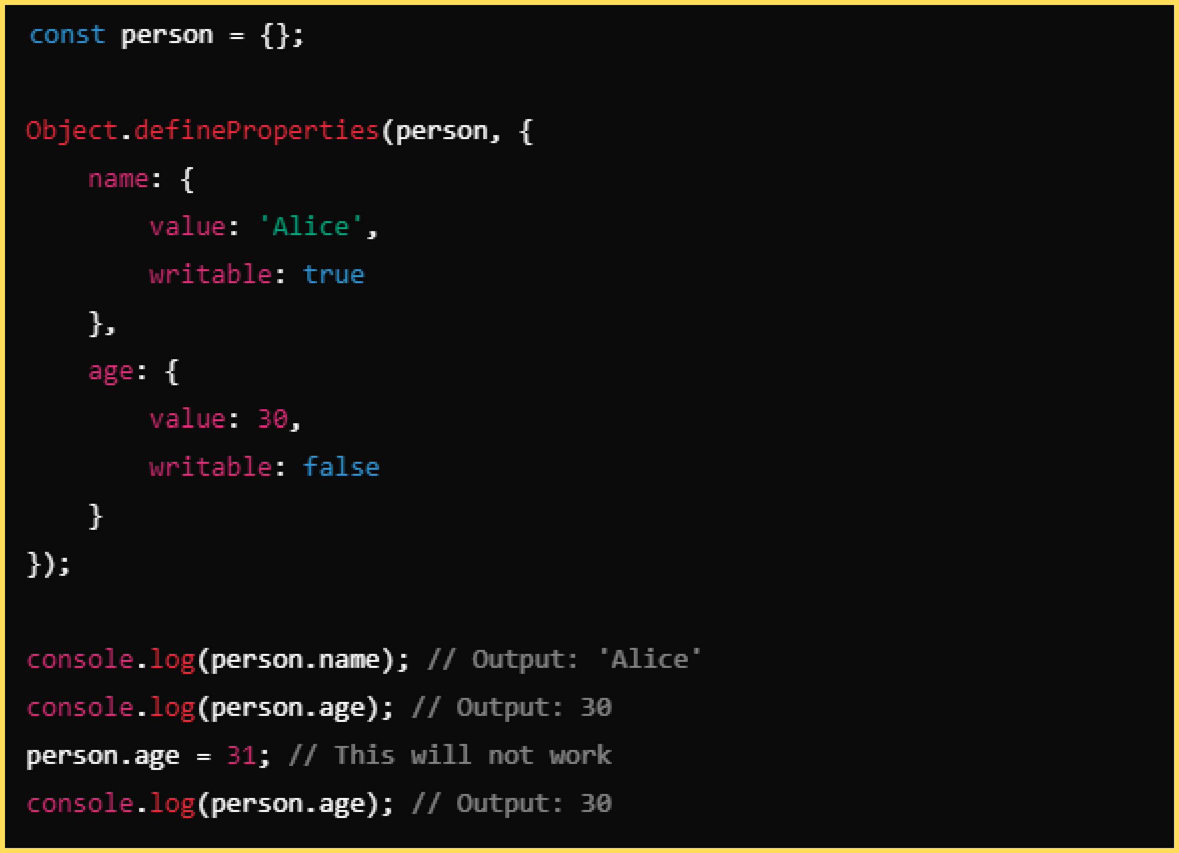
soumyabalamaala

{ JavaScript }

12. Object.defineProperties()

Definition:

Defines new properties or modifies existing properties on an object, and returns the object.



```
const person = {};

Object.defineProperties(person, {
    name: {
        value: 'Alice',
        writable: true
    },
    age: {
        value: 30,
        writable: false
    }
});

console.log(person.name); // Output: 'Alice'
console.log(person.age); // Output: 30
person.age = 31; // This will not work
console.log(person.age); // Output: 30
```

soumyabalamaala@gmail.com

soumyabalamaala

soumyabalamaala

{ JavaScript }

Follow for more such content on JavaScript

soumyabalamaala@gmail.com

soumyabalamaala

soumyabalamaala