Javascript ES6 concepts:

- String interpolation
- Object
- Arrow Function
- " this "
- Restructuring
- Spread Syntax

String Interpolation

Before ES6:
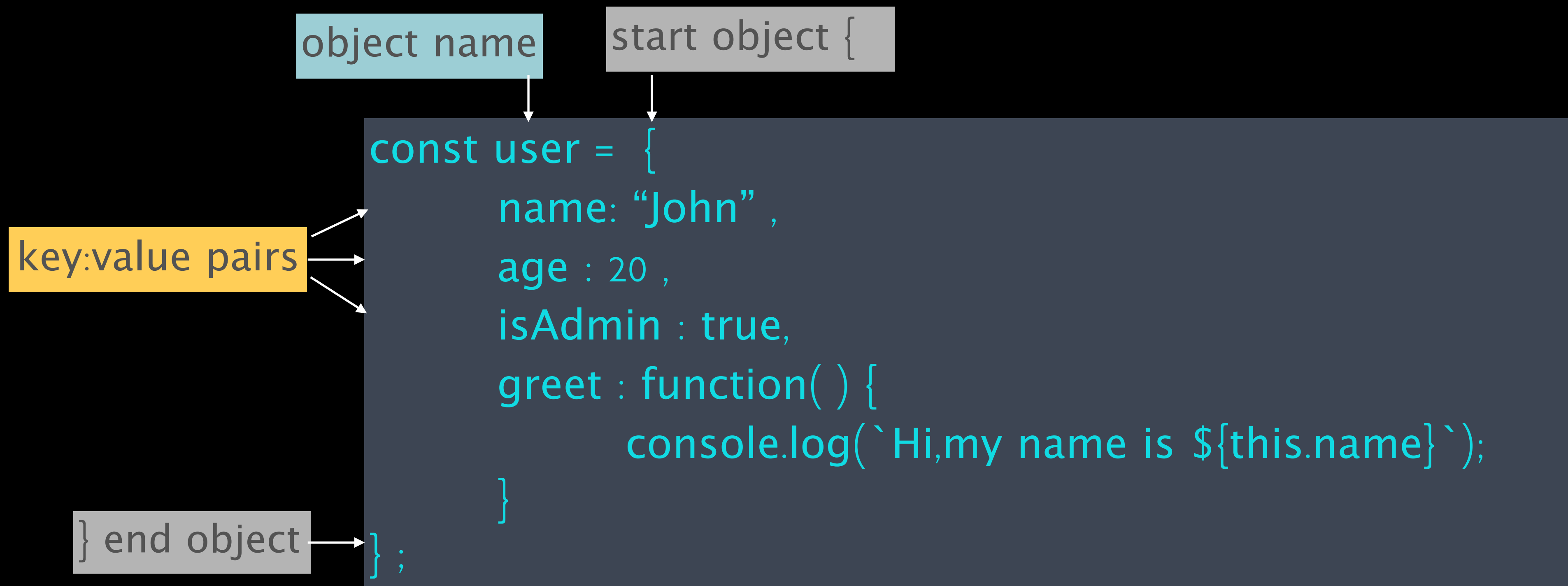
- using + to concatenate string

  let sentence = name + ' like to eat ' + food + '.'

After ES6:

- using back tick(``) and ${variable}

  let sentence = `${name} like to eat ${food}.`

Anatomy of an Object

object name

start object {

```javascript
const user = {
        name: "John" ,
        age : 20 ,
        isAdmin : true,
        greet : function( ) {
                console.log(`Hi,my name is ${this.name}`);
        }
} ;
```

key:value pairs

} end object

```
// "user" object with
// property key "name" store property value "John"
// property key "age" store property value 20
// property key "isAdmin" store property value true
```

Object

- Computed Property allows you to have an expression be computed as a property name on an object.
  Eg. const user = 'matt';
        {[user]: {id: 2, gender: 'male'}}; // {matt: {id:2, gender: 'male'}}
- To access properties on an object, you can either use dot notation or square bracket notation
  Eg. console.log(user.name) //John
        console.log(user['age']) //20
- Function as property value
  The keyword **this** exists whenever a function is defined
  this by default will refer to the owner of the function, in this case, user object owns the function greet

Arrow Function

It allows us to write more concise functions by removing some unnecessary syntax and refining the usage of **this** within functions.

Before ES6:

```
const multiply = function(x, y) {
    return x * y;
};
```

After ES6:

```
const multiply = (x, y) => x * y;
```

What About "this"?

- In regular functions the **this** keyword represented the object that called the function

- With arrow functions, the **this** keyword always represents the object that defined the arrow function

Destructuring

The destructuring assignment syntax is a JavaScript expression
that makes it possible to unpack values from arrays, or properties
from objects, into distinct variables.

Properties:

```
this.state.user.id = 19;
this.state.user.name = "John";
const {id, name} = this.state.user;
console.log(name); //John
```

Array:

```
[a,b] = [10,20];
console.log(a); //10
console.log(b); //20
```

Spread Syntax

- The syntax is three dots(…) followed by the array/object
- it can be used to:
  1. copy an array
  2. merge arrays
     Eg: let num1 = [1,2,3];
         let num2 = [6,7,8];
         let result = […num1,…num2];
         console.log(result); //[1,2,3,6,7,8]
  3. expand the array in a places where zero or more elements are
     expected
     Eg: let fruits = ['Apple','Orange','Banana'];
         let newFruits = ['Cherry', …fruits];
         console.log(newFruits); //['Cherry','Apple','Orange','Banana']

Complete the challenges for:

- Object
- Arrow function
- Destructuring
- Spread syntax
- this