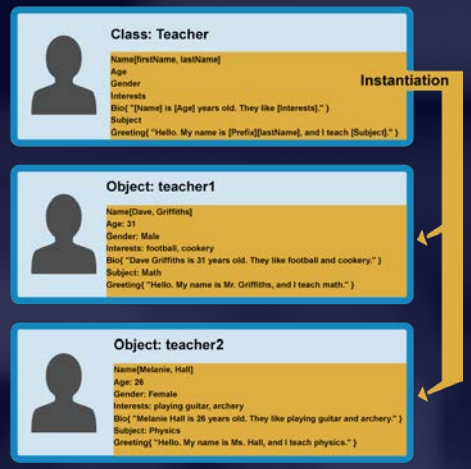
Objects and Classes



Using Objects and Classes Defining Simple Classes

Table of Contents

Objects

Definition, Properties and Methods

What Are Objects ?

- **Collection** of related data or functionality

- Consists of several **variables**
  - called **properties**
- Consists of several **functions**
  - called **methods**

- In JavaScript, at **run time** you can **add** and **remove** properties of any object

```
let obj = { name:'Peter', age: 20 };
console.log(obj.name); // Peter
```

Object name

Property name

Property value

Object Definition

- We can create an object with an **object literal**, using the following syntax:

```
let person = {name:'Peter', age: 20, hairColor: 'black'};
```

- We can define empty object and add the properties later

```
let person = {};
person.name ='Peter';
person["lastName"] = 'Parker';
person.age = 20;
person.hairColor = 'black';
```

You can access and set properties using both ways

Object Methods

- Functions **within a JavaScript object are called** methods
- We can **define** methods using several syntaxes:

```
let person = {
  sayHello : function() {
     console.log('Hi, guys');
  }
}
```

```
let person = {
  sayHello() {
     console.log('Hi, guys');
  }
}
```

- We can **add** a method to an already defined object

```
let person = { name: 'Peter', age: 20 };
person.sayHello = () => console.log('Hi, guys');
```

The Object Methods

Methods:

- `Object.entries()` - returns array of all properties and their values of an object
- `Object.keys()` - returns array with all the properties
- `Object.values()` - returns array with all the values of the properties

```
Object.entries(cat); // [['name', 'Tom'], ['age', 5]]
```

```
Object.keys(cat); // ['name', 'age']
```

```
Object.values(cat); // ['Tom', 5]
```

Iterate Through Keys

- Use **for-in** loop to iterate over the object properties by key:

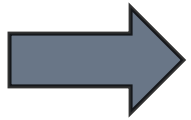
```
let obj = { name:'Peter', age:'18', grade:'5.50' };
for (let key in obj) {
    console.log(`${key}: ${obj[key]}`);
}
```

Returns the value of the property

Problem: Person Info

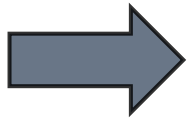
- Create an object that has first name, last name and age
- Print the entries of a given object

```
Peter
Pan
20
```



```
firstName: Peter
lastName: Pan
age: 20
```

```
Jack
Sparrow
unknown
```



```
firstName: Jack
lastName: Sparrow
age: unknown
```

Solution: Person Info

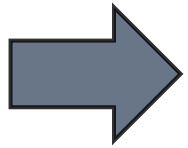
- Create an object
- Set the properties first name, last name and age
- Loop through the object properties using `for-in` loop
- Print the object keys and values

```
// TODO: Create the person object and set the properties
for (let key in person) {
    console.log(`${key}: ${person[key]}`);
}
```

Problem: City

- **Create an** object, **which will hold** area, population, country **and**
  postal code
- Loop through all the keys and print them with their values

```
Atlanta
347
420003
USA
30060
```



```
name -> Atlanta
area -> 347
population -> 420003
country -> USA
postCode -> 30060
```

Solution: City

- Create an object and set the properties
- Get the object entries
- Loop through the object entries using **for-of** loop
- Print the object keys and values

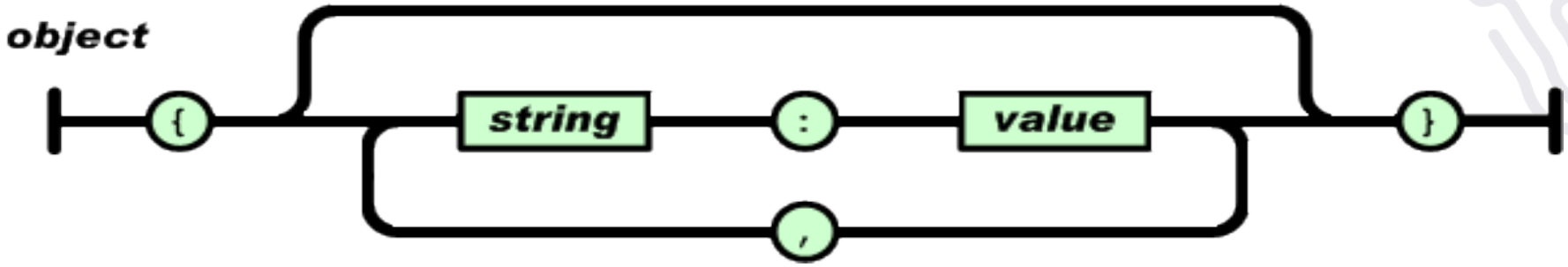
```
// TODO: Create the city object and set the properties
let entries = Object.entries(city);
for (let [ key, value ] of entries) {
    console.log(`${key} -> ${value}`);
}
```

JSON

JavaScript Object Notation

What is JSON

- **JSON** stands for **J**ava**S**cript **O**bject **N**otation
- **Open-standard** file format that uses text to transmit data objects
- JSON is **language independent**
- JSON is "**self-describing**" and easy to understand



JSON Usage

- Exchange data between **browser** and **server**
- JSON is a **lightweight** format compared to XML
- JavaScript has built in functions to **parse JSON** so it's easy to use
- JSON uses **human-readable** text to transmit data

JSON Example

Brackets define a JSON

Keys are in double quotes

Keys and values separated by :

```
{
  "name": "Ivan",
  "age": 25,
  "grades": {
    "Math": [2.50, 3.50],
    "Chemistry": [4.50]
  }
}
```

It is possible to have nested objects

In JSON we can have arrays

JSON Methods

- We can convert object into **JSON** string using **`JSON.stringify(object)`** method

```
let text = JSON.stringify(obj);
```

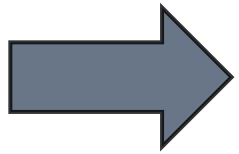
- We can convert JSON string into object using **`JSON.parse`**`(text)` method

```
let obj = JSON.parse(text);
```

Problem: Convert to Object

- Write a function, that receives a string in **JSON** format and converts it to object
- Print the entries of the object

```
'{
"name": "George",
"age": 40,
"town": "Atlanta"
}'
```



```
name: George
age: 40
town: Atlanta
```

Tips: Convert to Object

- Use **`JSON.parse()`** method to parse JSON string to an object
- Use **`Object.entries()`** method to get object's properties names and values
- Loop through the entries and print them

```
function objConverter(json) {
    // TODO: Use the tips to write the function
}
```

Solution: Convert to Object

```
function objConverter(json) {
    let person= JSON.parse(json);

    let entries = Object.entries(person);

    for (let [key, value] of entries) {
        console.log(`${key}: ${value}`);
    }
}
```

Problem: Convert to JSON

- **Write a** function **that receives** first name, last name, hair color **and sets them to an object**
- Convert the object to **JSON string** and print it

```
'George',
'Jones',
'Brown'
```

```
{"firstName": "George",
"lastName": "Jones",
"hairColor": "Brown"}
```

Tips: Convert to JSON

- Create an object with the given input
- Use **`JSON.stringify()`** method to parse object to JSON string
- Keep in mind that the property name in the JSON string will be
  exactly the same as the property name in the object

```
function solve(name, lastName, hairColor){
    // TODO: Use the tips and write the code
}
```

Solution: Convert to JSON

```
function convertJSON(name, lastName, hairColor) {
    let person = {
        name,
        lastName,
        hairColor
     };
    console.log(JSON.stringify(person));
}
```

Classes

Object Models

What Are Classes

- Extensible program-code-template for creating objects
- Provides **initial values** for the state of an object
- An object created by the class pattern is called an **instance** of that class
- A class has a constructor - subroutine called to create an object
  - It prepares the new object for use

Class Declaration

To declare a class we use the **`class`** keyword with the name of the class.

```
class Student {
  constructor(name) {
    this.name = name;
  }
}
```

The **`constructor`** is a special method for creating and initializing an object

Class Example

- Creating a class:

this keyword is used to set a property of the objects to a given value

```
class Student {
  constructor(name, grade) {
    this.name = name;
    this.grade = grade;
  }
}
```

- Creating an **instance** of the class:

```
let student = new Student('Peter', 5.50);
```

Functions in a Class

- Classes can also have functions as property, called methods:

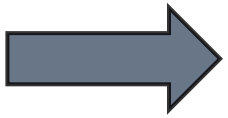
```
class Dog {
  constructor() {
    this.speak = () => {
      console.log('Woof');
    }
  }
}

let dog = new Dog();
dog.speak(); // Woof
```

We access the method as a regular property

Problem: Cat

- Write a function that receives **array of strings** in the following format: **`'{cat name} {age}'`**
- Create a class **`Cat`** that receives the **name** and the **age** parsed from the input
- It should also have a function named **`meow()`** that will print **`"{cat name}, age {age} says Meow"`** on the console
- For each of the strings provided you must create a cat object

```
['Mellow 2','Tom 5']
```



```
Mellow, age 2 says Meow
Tom, age 5 says Meow
```

Tips: Cat

- **Create a** class
- **Set** properties **name and age**
- Set property '**`meow`**' to be a function that prints the result
- **Parse** the input data
- Create all objects using class **`constructor`** and the parsed input data and store them in an array
- Loop through the array using **for...of** loop and invoke **`.meow()`** method

Solution: Cat

```
function catCreator(arr) {
    // TODO: Create the Cat class
     let cats = [];
     for (let i = 0; i < arr.length; i++) {
        let catData = arr[i].split(' ');
        let [name, age] = [catData[0], catData[1]];
         cats.push(new Cat(name, age));
     }
     // TODO: Iterate through cats[] and invoke .meow()
    using for…of loop
}
```

Live Exercises

Summary

- Objects hold **`key-value` pairs**
  - Access key and value **by `index`** in loops
  - Access value with `['key name']`
  - Access value with **`obj.key`**
- Use Object **`Methods`** such as:
  - **`Object.keys`**
  - **`Object.values`**
- **Parse** and **`stringify`** objects in **`JSON`**

Questions?

License

- This course (slides, examples, demos, exercises, homework, documents, videos and other assets) is **copyrighted content**
- Unauthorized copy, reproduction or use is illegal
- © Kingsland University – https://kingslanduniversity.com


KINGSLAND
UNIVERSITY
THANK YOU