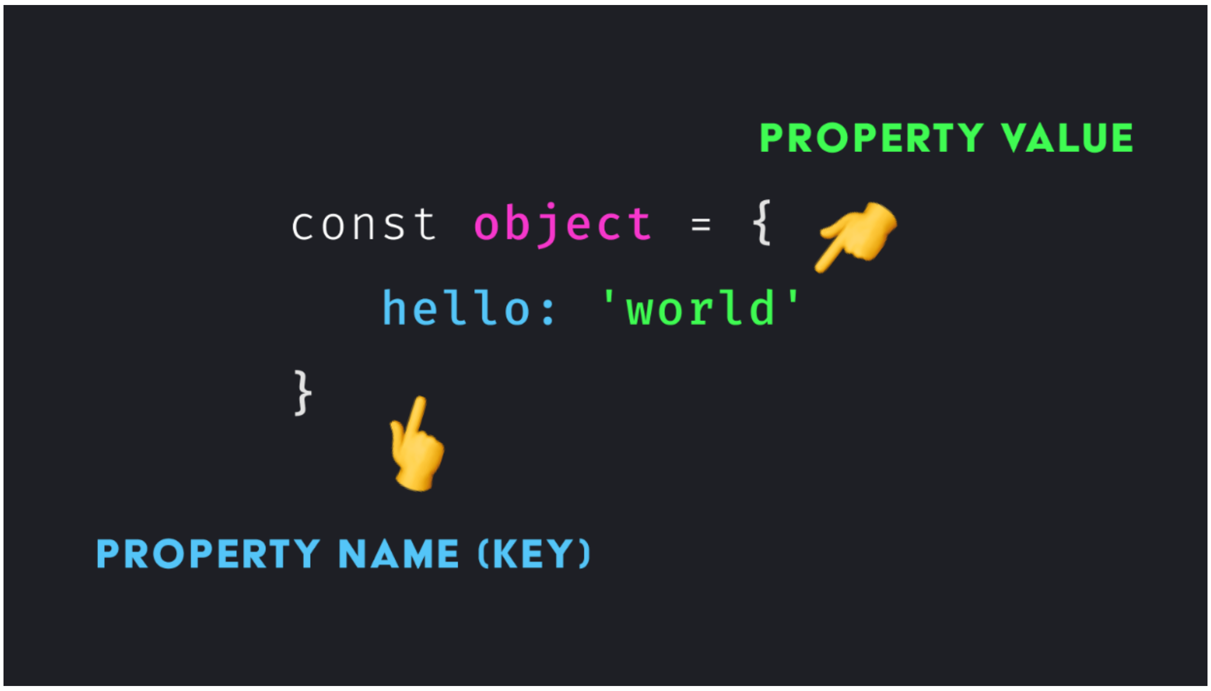
Object

Everything can be an object and objects do have properties and properties have values, so an object is a key value pair. The order of the key is not reserved, or there is no order. To create an object literal, we use two curly brackets.



```
const object = {
    hello: 'world'
}
```

Creating an empty object

An empty object

```
const person = {}
```

Creating an objecting with values

Now, the person object has firstName, lastName, age, location, skills and isMarried properties. The value of properties or keys could be a string, number, boolean, an object, null, undefined or a function.

Let us see some examples of object. Each key has a value in the object.

```
const rectangle = {
  length: 20,
  width: 20
}
console.log(rectangle) // {length: 20, width: 20}

const person = {
  firstName: 'Afaf',
  lastName: 'kelai',
  age: 25,
  country: 'Algeria',
  city: 'Algiers',
  skills: [
    'HTML',
    'CSS',
    'JavaScript',
    'React',
    'Node',
    'MongoDB',
    'Python',
    'D3.js'
  ],
  isMarried: false
}
console.log(person)
```

Getting values from an object

We can access values of object using two methods:

- using . followed by key name if the key-name is a one word
- using square bracket and a quote

```
const person = {
  firstName: 'Afaf',
  lastName: 'kelai',
  age: 25,
  country: 'Algeria',
  city: 'Algiers',
  skills: [
    'HTML',
    'CSS',
    'JavaScript',
    'React',
    'Node',
    'MongoDB',
    'Python',
    'D3.js'
  ],
  getFullName: function() {
    return `${this.firstName}${this.lastName}`
  },
  'phone number': '+2134545454545'
}

// accessing values using .
console.log(person.firstName)
console.log(person.lastName)
console.log(person.age)
console.log(person.location)

// value can be accessed using square bracket and key name
console.log(person['firstName'])
console.log(person['lastName'])
console.log(person['age'])
console.log(person['age'])
console.log(person['location'])

// for instance to access the phone number we only use the square bracket method
console.log(person['phone number'])
```

Creating object methods

Now, the person object has getFullName properties. The getFullName is function inside the person object and we call it an object method. The *this* key word refers to the object itself. We can use the word *this* to access the values of different properties of the object. We can not use an arrow function as object method because the word this refers to the window inside an arrow function instead of the object itself. Example of object:

```
const person = {
  firstName: 'Afaf',
  lastName: 'kelai',
  age: 25,
  country: 'Algeria',
  city: 'Algiers',
  skills: [
    'HTML',
    'CSS',
    'JavaScript',
    'React',
    'Node',
    'MongoDB',
    'Python',
    'D3.js'
  ],
  getFullName: function() {
    return `${this.firstName} ${this.lastName}`
  }
}

console.log(person.getFullName());
// kelai afaf
```

Object Methods

There are different methods to manipulate an object. Let us see some of the available methods.

Object.assign: To copy an object without modifying the original object

```
const person = {
  firstName: 'Afaf',
  age: 25,
  country: 'Algeria',
  city:'Algiers',
  skills: ['HTML', 'CSS', 'JS'],
  title: 'Instractor',
  address: {
    street: 'somewhere 16',
    pobox: 2021,
    city: 'Algiers'
  },
  getPersonInfo: function() {
    return `I am ${this.firstName} and I live in ${this.city}, ${this.country}. I am ${this.age}.`
  }
}

//Object methods: Object.assign, Object.keys, Object.values, Object.entries
//hasOwnProperty

const copyPerson = Object.assign({}, person)
console.log(copyPerson)
```

Getting object keys using Object.keys()

Object.keys: To get the keys or properties of an object as an array

```
const keys = Object.keys(copyPerson)
console.log(keys) //['name', 'age', 'country', 'skills', 'address', 'getPersonInfo']
const address = Object.keys(copyPerson.address)
console.log(address) //['street', 'pobox', 'city']
```

Getting object values using Object.values()

Object.values:To get values of an object as an array

```
const values = Object.values(copyPerson)
console.log(values)
```

Getting object keys and values using Object.entries()

Object.entries:To get the keys and values in an array

```
const entries = Object.entries(copyPerson)
console.log(entries)
```

Checking properties using hasOwnProperty()

hasOwnProperty: To check if a specific key or property exist in an object

```
console.log(copyPerson.hasOwnProperty('name'))
console.log(copyPerson.hasOwnProperty('score'))
```

Exercises

Exercises: Level 1

1. Create an empty object called dog
2. Print the the dog object on the console
3. Add name, legs, color, age and bark properties for the dog object. The bark property is a method which return *woof woof*
4. Get name, legs, color, age and bark value from the dog object
5. Set new properties the dog object: breed, getDogInfo

Exercises: Level 2

1. Find the person who has many skills in the users object.
2. Count logged in users,count users having greater than equal to 50 points from the following object.

```
const users = {
  Alex: {
    email: 'alex@alex.com',
    skills: ['HTML', 'CSS', 'JavaScript'],
    age: 20,
    isLoggedIn: false,
    points: 30
  },
  Asab: {
    email: 'asab@asab.com',
    skills: ['HTML', 'CSS', 'JavaScript', 'Redux', 'MongoDB', 'Express', 'React', 'Node'],
    age: 25,
    isLoggedIn: false,
    points: 50
  },
  Brook: {
    email: 'daniel@daniel.com',
    skills: ['HTML', 'CSS', 'JavaScript', 'React', 'Redux'],
    age: 30,
    isLoggedIn: true,
    points: 50
  },
  Daniel: {
    email: 'daniel@alex.com',
    skills: ['HTML', 'CSS', 'JavaScript', 'Python'],
    age: 20,
    isLoggedIn: false,
    points: 40
  },
  John: {
    email: 'john@john.com',
    skills: ['HTML', 'CSS', 'JavaScript', 'React', 'Redux', 'Node.js'],
    age: 20,
    isLoggedIn: true,
    points: 50
  },
  Thomas: {
    email: 'thomas@thomas.com',
    skills: ['HTML', 'CSS', 'JavaScript', 'React'],
    age: 20,
    isLoggedIn: false,
    points: 40
  },
  Paul: {
    email: 'paul@paul.com',
    skills: ['HTML', 'CSS', 'JavaScript', 'MongoDB', 'Express', 'React', 'Node'],
    age: 20,
    isLoggedIn: false,
    points: 40
  }
}```
```

1. Find people who are MERN stack developer from the users object
2. Set your name in the users object without modifying the original users object
3. Get all keys or properties of users object
4. Get all the values of users object
5. Use the countries object to print a country name, capital, populations and languages.

Exercises: Level 3

1. Create an object literal called *personAccount*. It has *firstName*, *lastName*, *incomes*, *expenses* properties and it has *totalIncome*, *totalExpense*, *accountInfo*,*addIncome*, *addExpense* and *accountBalance* methods. Incomes is a set of incomes and its description and expenses is a set of incomes and its description.
2. *** Questions:2, 3 and 4 are based on the following two arrays:users and products ()

```
const users = [
    {
        _id: 'ab12ex',
        username: 'Alex',
```

```
        email: 'alex@alex.com',
        password: '123123',
        createdAt:'08/01/2020 9:00 AM',
        isLoggedIn: false
    },
    {
        _id: 'fg12cy',
        username: 'Asab',
        email: 'asab@asab.com',
        password: '123456',
        createdAt:'08/01/2020 9:30 AM',
        isLoggedIn: true
    },
    {
        _id: 'zwf8md',
        username: 'Brook',
        email: 'brook@brook.com',
        password: '123111',
        createdAt:'08/01/2020 9:45 AM',
        isLoggedIn: true
    },
    {
        _id: 'eefamr',
        username: 'Martha',
        email: 'martha@martha.com',
        password: '123222',
        createdAt:'08/01/2020 9:50 AM',
        isLoggedIn: false
    },
    {
        _id: 'ghderc',
        username: 'Thomas',
        email: 'thomas@thomas.com',
        password: '123333',
        createdAt:'08/01/2020 10:00 AM',
        isLoggedIn: false
    }
    ];

    const products = [
  {
    _id: 'eedfcf',
    name: 'mobile phone',
    description: 'Huawei Honor',
    price: 200,
    ratings: [
      { userId: 'fg12cy', rate: 5 },
      { userId: 'zwf8md', rate: 4.5 }
    ],
    likes: []
  },
  {
    _id: 'aegfal',
    name: 'Laptop',
    description: 'MacPro: System Darwin',
    price: 2500,
    ratings: [],
    likes: ['fg12cy']
  },
  {
    _id: 'hedfcg',
    name: 'TV',
    description: 'Smart TV:Procaster',
    price: 400,
    ratings: [{ userId: 'fg12cy', rate: 5 }],
    likes: ['fg12cy']
  }
]
```

magine you are getting the above users collection from a MongoDB database. a. Create a function called signUp which allows user to add to the collection. If user exists, inform the user that he has already an account.b. Create a function called signIn which allows user to sign in to the application

1. The products array has three elements and each of them has six properties. a. Create a function called rateProduct which rates the product b. Create a function called averageRating which calculate the average rating of a product
2. Create a function called likeProduct. This function will helps to like to the product if it is not liked and remove like if it was liked.