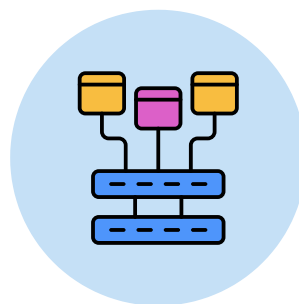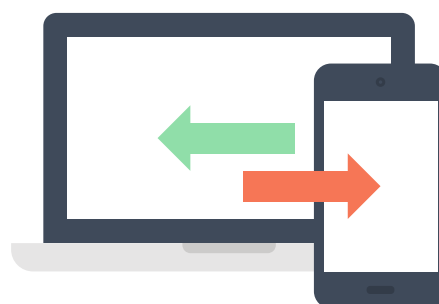# Objects { }

They are **complex data structure** which can contain **more than one value** with **name properties.** They have **properties** and they can perform actions.

Objects are used to **define real life objects more flexibly than arrays**

Applications

Data modelling
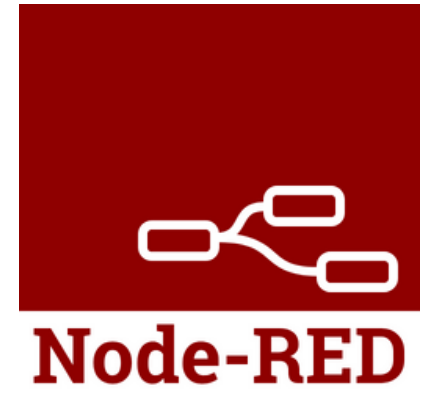(Represent real world objects or products example Vehicles)

Data exchange
being compatible with JSON

OOP
Objects can be easily be used to create classes and instances

www.codeandcompile.com

Objects { }

Example:

Object name

property

value

```javascript
1   var cat = {
2       catName: 'JavaScript',
3       weight: 45,
4       color: 'brown',
5       age: 3,
6   };
7   console.log(cat);
8   return cat;
```
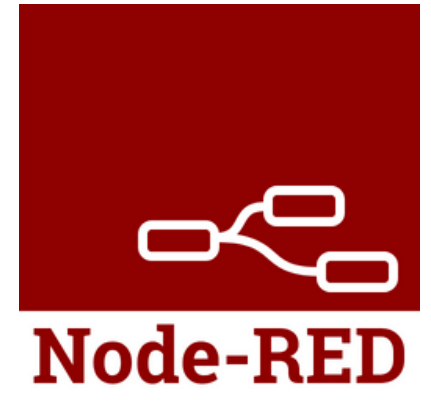
```
msg : Object
▾ object
    catName: "JavaScript"
    weight: 45
    color: "brown"
    age: 3
    _msgid: "0f2b425395e12a2f"
```

```
{ catName: 'JavaScript', weight: 45, color: 'brown', age: 3 }
```
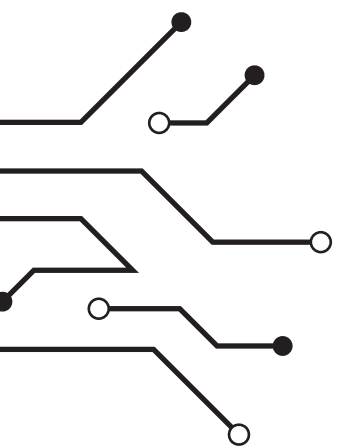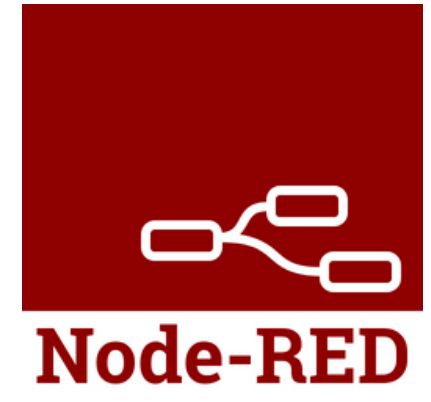
Objects { }

Accessing properties:

The object properties can be accessed using the name of the properties unlike arrays where you have to give the index number.

```javascript
1  var cat = {
2      catName: 'JavaScript',
3      weight: 45,
4      color: 'brown',
5      age: 3,
6  };
7  console.log(cat['catName']);
8  console.log(cat.color);
```

```
JavaScript
brown
```

Objects { }

Updating object

We can easily change the value of the properties as shown below:

```
1    var cat = {
2        catName: 'JavaScript',
3        weight: 45,
4        color: 'brown',
5        age: 3,
6    };
7    cat['weight'] = 32;
8    cat.color = 'white';
9    console.log(cat);
10   return cat;
```
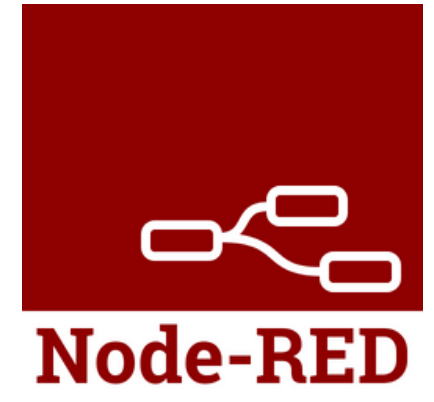
```
msg : Object
▼ object
    catName: "JavaScript"
    weight: 32
    color: "white"
    age: 3
    _msgid: "f84193453e561027"
```

```
{ catName: 'JavaScript', weight: 32, color: 'white', age: 3 }
```

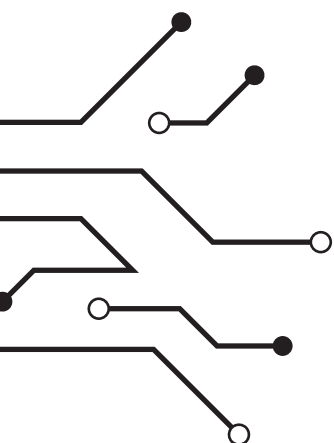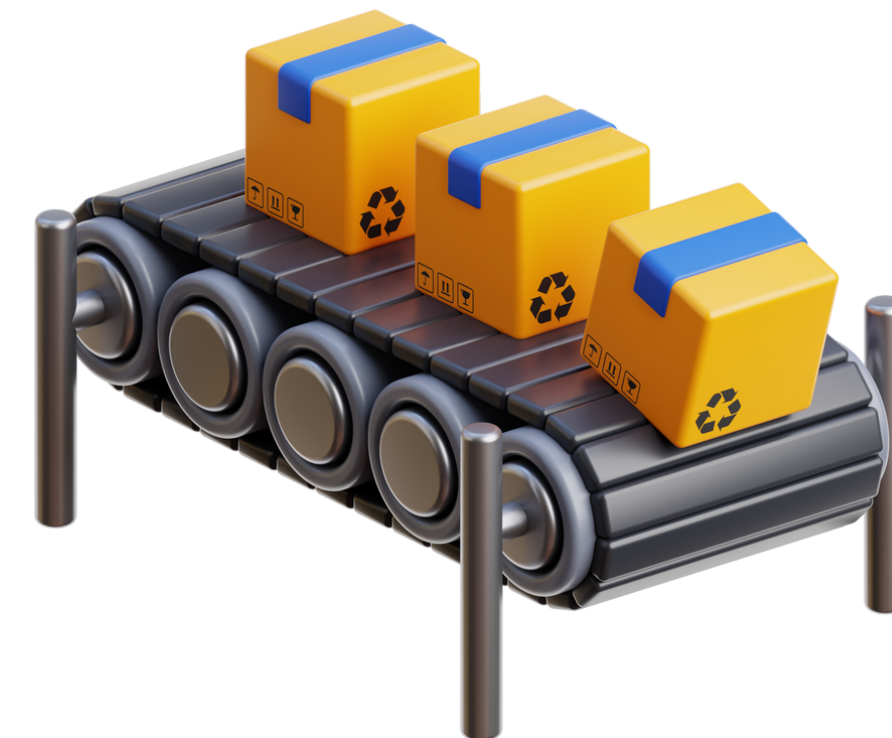It is possible to change the property value to different datatype.
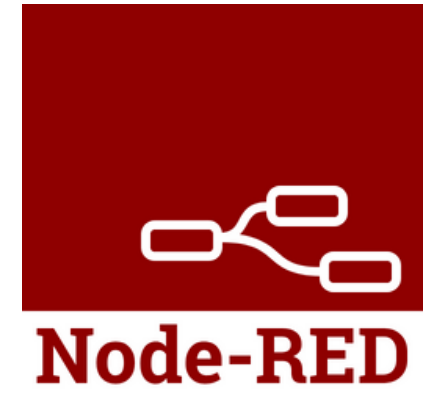
Project 1

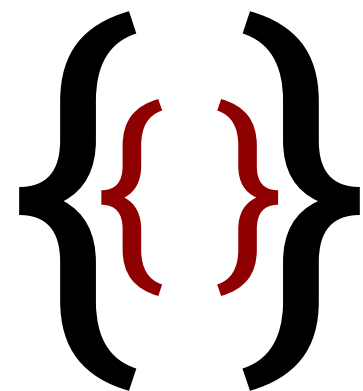- Create a new object named **productionStatus**. Add the following properties:
  - lineName (string)
  - managerName (string)
  - productionCount (number)
  - Started (string)
  - Finished (string)
- Assign a new value to **productionCount**
- Output **productionStatus** to console and debug window
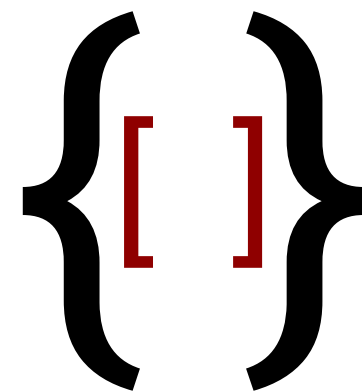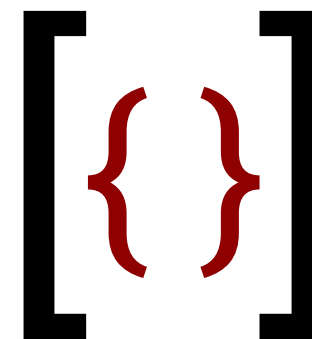
Objects { } & Arrays [ ]

Objects and arrays can be combined and used together in the following structure:
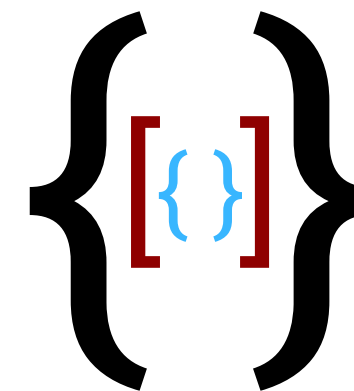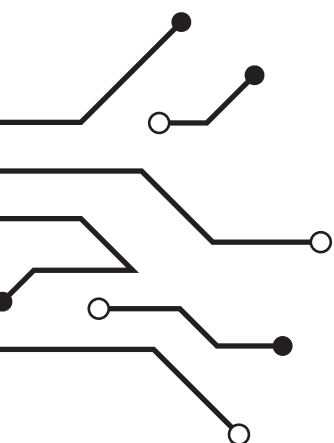
{ { } }
Objects in Objects
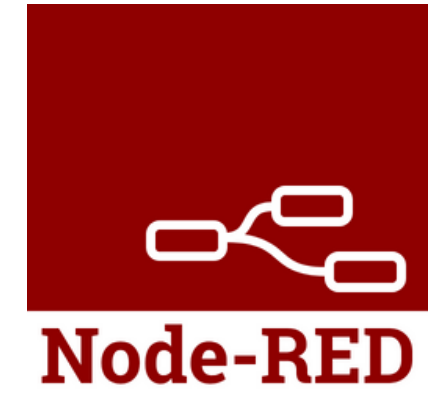
{ [ ] }
Arrays in Objects

[{ }]
Objects in Arrays

{ [{ }] }
Objects in Arrays in Object

Objects in Objects {}

Example: Object 1: Company, **Object 2**: Company's Address

```
1   var company = {
2       name: 'Code and Compile',
3       founder: 'Rajvir Singh',
4       address: {
5           city: 'Überlingen',
6           zipcode: 88662,
7           country: 'Germany'
8       },
9       yearOfEstablishment: 2018
10  };
11  console.log(company);
12  return company;
```

```
{
    name: 'Code and Compile',
    founder: 'Rajvir Singh',
    address: { city: 'Überlingen', zipcode: 88662, country: 'Germany' },
    yearOfEstablishment: 2018
}
```

```
msg : Object
▾ object
    name: "Code and Compile"
    founder: "Rajvir Singh"
▾ address: object
        city: "Überlingen"
        zipcode: 88662
        country: "Germany"
    yearOfEstablishment: 2018
    _msgid: "f01e4613d81b287e"
```
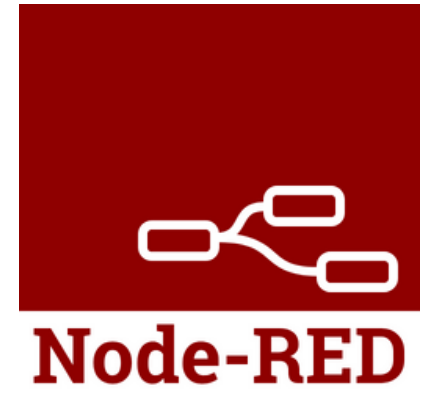
Accessing the property of the object in object

```
11  console.log(company.address.country);
```
Germany

www.codeandcompile.com

Array in Objects {[ ]}

Example: Object: Company, **Array**: Courses

```js
1   var company = {
2       name: 'Code and Compile',
3       founder: 'Rajvir Singh',
4       courses: ['PLC','Node-RED','OPC UA','MQTT'],
5       address: {
6           city: 'Überlingen',
7           zipcode: 88662,
8           country: 'Germany'
9       },
10      yearOfEstablishment: 2018
11  };
12  console.log(company);
13  return company;
```

```
{
  name: 'Code and Compile',
  founder: 'Rajvir Singh',
  courses: [ 'PLC', 'Node-RED', 'OPC UA', 'MQTT' ],
  address: { city: 'Überlingen', zipcode: 88662, country: 'Germany' },
  yearOfEstablishment: 2018
}
```

msg : Object
▼ object
    name: "Code and Compile"
    founder: "Rajvir Singh"
  ▼ courses: array[4]
        0: "PLC"
        1: "Node-RED"
        2: "OPC UA"
        3: "MQTT"
  ▼ address: object
        city: "Überlingen"
        zipcode: 88662
        country: "Germany"
    yearOfEstablishment: 2018
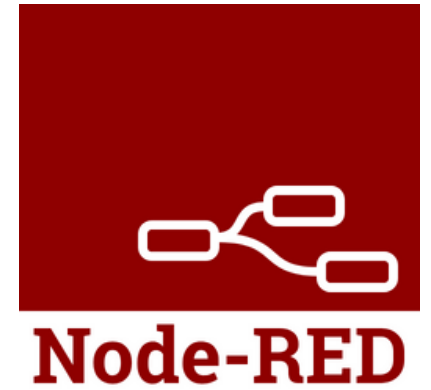   _msgid: "663052f8c0266e7c"

Accessing the property of the array in object

```js
console.log(company.courses[1]);
```
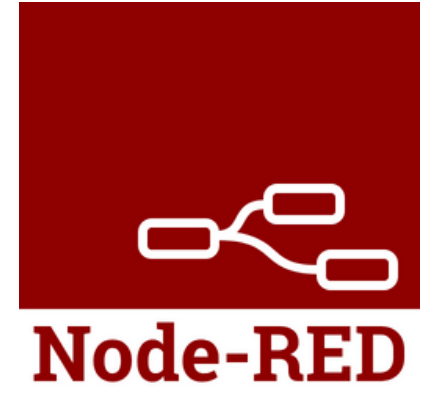Node-RED

Objects in Arrays [{ }]

Example: Array: Courses, **Object**: Course details

```javascript
var courses = [
    {
        name: 'Node-RED',
        duration_hours: 6,
        price_euros: 99,
        access: 'lifetime'
    },
    {
        name: 'IIoT Bundle',
        duration_hours: 35,
        price_euros: 119,
        access: 'lifetime'
    }
]
console.log(courses);
console.log(courses[1].name);
msg.payload = courses;
return msg;
```

```
[
    {
        name: 'Node-RED',
        duration_hours: 6,
        price_euros: 99,
        access: 'lifetime'
    },
    {
        name: 'IIoT Bundle',
        duration_hours: 35,
        price_euros: 119,
        access: 'lifetime'
    }
]
IIoT Bundle
```

```
msg : Object
▼ object
    _msgid: "de9929bf24860e49"
▼ payload: array[2]
    ▼ 0: object
        name: "Node-RED"
        duration_hours: 6
        price_euros: 99
        access: "lifetime"
    ▼ 1: object
        name: "IIoT Bundle"
        duration_hours: 35
        price_euros: 119
        access: "lifetime"
```

Objects in Arrays in Object

Example: Object: Company**, Array**: Courses, **Object**: Course details

```javascript
1   var company = {
2       name: 'Code and Compile',
3       founder: 'Rajvir Singh',
4       courses: [
5           {
6               name: 'Node-RED',
7               duration_hours: 6,
8               price_euros: 99,
9               access: 'lifetime'
10          },
11          {
12              name: 'IIoT Bundle',
13              duration_hours: 35,
14              price_euros: 119,
15              access: 'lifetime'
16          }
17      ],
18      address: {
19          city: 'Überlingen',
20          zipcode: 88662,
21          country: 'Germany'
22      },
23      yearOfEstablishment: 2018
24  };
25  console.log(company);
26  return company;
```

```javascript
{
    name: 'Code and Compile',
    founder: 'Rajvir Singh',
    courses: [
        {
            name: 'Node-RED',
            duration_hours: 6,
            price_euros: 99,
            access: 'lifetime'
        },
        {
            name: 'IIoT Bundle',
            duration_hours: 35,
            price_euros: 119,
            access: 'lifetime'
        }
    ],
    yearOfEstablishment: 2018
}
```

```
msg : Object
▼object
  name: "Code and Compile"
  founder: "Rajvir Singh"
  ▼courses: array[2]
    ▼0: object
        name: "Node-RED"
        duration_hours: 6
        price_euros: 99
        access: "lifetime"
    ▼1: object
        name: "IIoT Bundle"
        duration_hours: 35
        price_euros: 119
        access: "lifetime"
  yearOfEstablishment: 2018
  _msgid: "a15687e422650573"
```
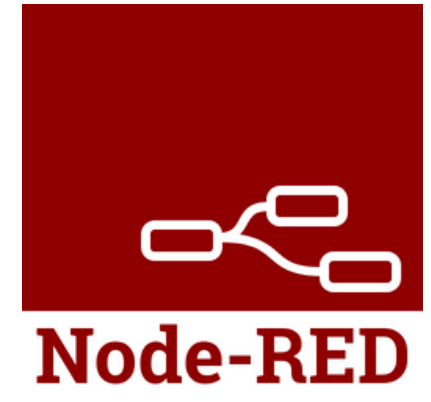
Code </> Compile
learning made easy

Project 2

In this project, update the product catalog from the website
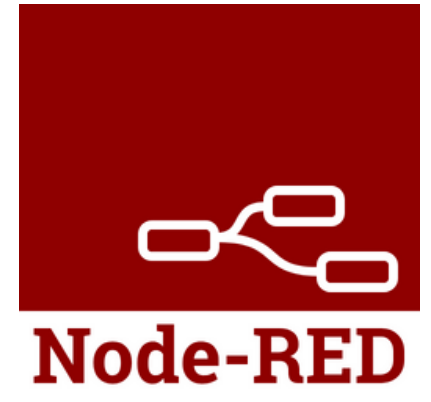**https://www.codeandcompile.com**

- Add atleast 5 more courses
- Take user input to get the course details
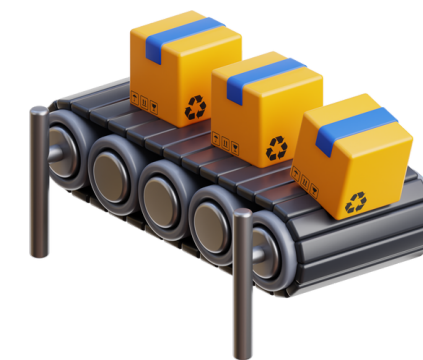 - Name
  - Duration
  - Price
  - Access

Project 3

Optimize the Project 1 output such that the number of lines increases to 3
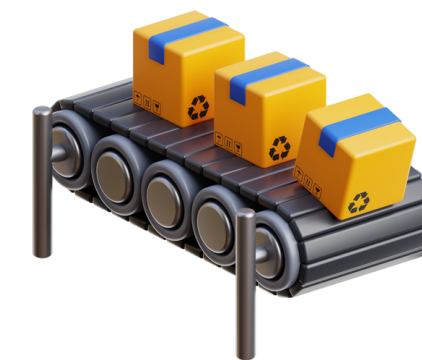
- Organize the object as follows:
  - line1 (object)
    - name (string)
    - managerName (string)
    - productionCount (array)
    - ...
  - line2 (object)
    - ...
  - line3 (object)
    - ...
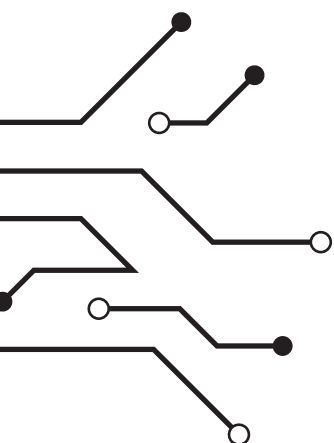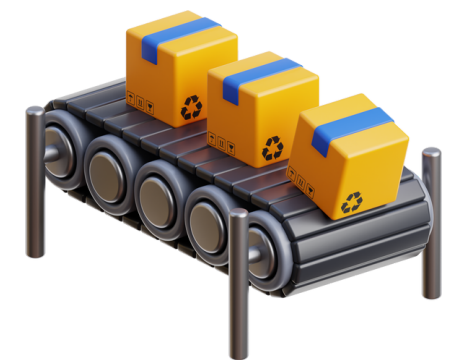- Output **totalProductionStatus** to console and debug window
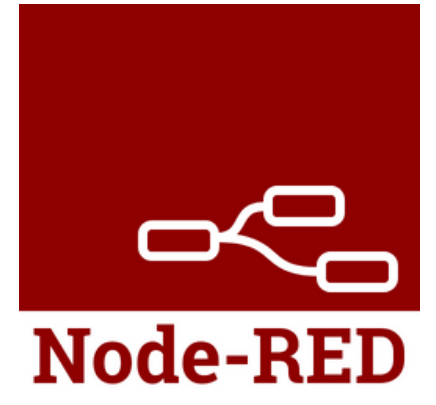
Line 1   **Line 2**   **Line 3**

Code </> Compile

Thank you!