

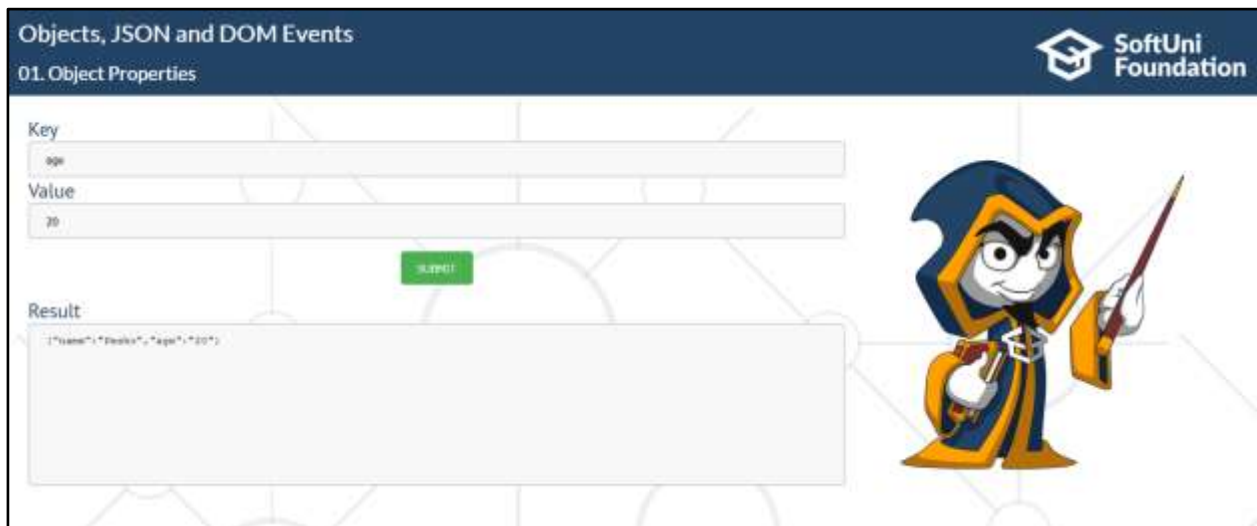
Lab: Objects and JSON, DOM Events

Problems for in-class lab for the [“JavaScript Fundamentals” course @ SoftUni](#).

1. Object Properties

You will be provided with a **skeleton** which contains a **form for adding object properties**. When submitting the form, you should be able to **add a property** to an object and **display it in a read-only text box**.

Example



Hints

Let us see what we have first. This is where you should complete the html you are provided with:

```
<script>
    document.getElementById("button").addEventListener("click", function(event) {
        event.preventDefault()
    });
    let obj = {}
    function addProperty() {
        // TODO
    }
</script>
```

- You are given an empty object to start with.
- The page is prevented from reloading when you click the button, because we want to see the result in our result checkbox.

Now, let us complete the script. To begin with, each time the button is clicked, we want to **get what was submitted** in the form. To do that, we need to check the id of both fields:

```

<div id="exercise">
  <form>
    <label>Key</label>
    <input type="text" id="key"></input>
    <label>Value</label>
    <input type="text" id="value"></input>
    <div class="center">
      <input type="submit" id="button" onclick="addProperty()"
        value="Submit"></input>
    </div>
  </form>
  <label>Result</label>
  <textarea readonly rows="10" cols="50" id="result"></textarea>
</div>

```

So we add that to our code:

```

function addProperty() {
  let objKey = document.getElementById("key").value;
  let objValue = document.getElementById("value").value;
}

```

Then, we add the property to our object and fill the textbox with the stringified version of the object. To do that, we search for the id of the textbox ("result") and we fill it with the object. We use **JSON.stringify**:

```

function addProperty() {
  let objKey = document.getElementById("key").value;
  let objValue = document.getElementById("value").value;
  obj[objKey] = objValue;
  document.getElementById("result").innerHTML = JSON.stringify(obj);
}

```

Ready! You can now experiment with what we have.

Check your solution here: [Link to judge](#)

1. Students Register

You will be given names of students with their grades in the format: "{firstName} {secondName} {grade}". Store the information about the students and at the end print the result in the textbox in the following format:

"Students are: {students names joined by ", "}"

"Average grade: {averageGrade formatted to second decimal point}"

Example

Objects, JSON and DOM Events

02. Students Register

Students

```
Ivan Ivanov 6.00
Gosho Goshev 6.00
Pesho Peshov 3.40
```

ADD STUDENTS

Result

```
Students are: Ivan Ivanov, Gosho Goshev, Pesho Peshov
Average grade: 5.13
```

Hints

First, we have to get our input and create an empty array to store the students:

```
function addStudents() {
    let students = document.getElementById("input").value.split("\n");
    let result = [];
}
```

- We split the received data by a new line.

```
function addStudents() {
    let students = document.getElementById("input").value.split("\n");
    let result = [];
    for (let student of students) {
        let info = student.split(" ");
        let obj = {};
        obj["first name"] = info[0];
        obj["second name"] = info[1];
        obj["grade"] = Number(info[2]);
        result.push(obj);
    }
}
```

- We split the info on each line by a space.
- We create new object.

- We push the object to the array.
- Finally, we create the resulting string to put in the text box.

```
function addStudents() {
  let students = document.getElementById("input").value.split("\n");
  let result = [];
  for (let student of students) {
    let info = student.split(" ");
    let obj = {};
    obj["first name"] = info[0];
    obj["second name"] = info[1];
    obj["grade"] = Number(info[2]);
    result.push(obj);
  }

  let resultStr = "";
  resultStr += "Students are: ";
  resultStr += result.map(x => x["first name"] + " " + x["second name"]).join(", ");
  resultStr += "\n";
  resultStr += "Average grade: " + (result.map(x => x["grade"]).reduce((a, b) => a + b) /
  result.length).toFixed(2);
  document.getElementById("result").innerHTML = resultStr;
}
```

- We take only the **first and second name** of each student, using the **map** function.
- We join them by space.
- Then, we calculate the average grade by using the **reduce** function (this returns the **sum of all the grades**) and we have to **divide the sum by the count of the students**, which gives us the **average grade**.
- We format the grade to the second decimal point and we display it in the text box.

2. Rent

You will be given information in separate input fields. The first field will contain flat numbers, separated by a space. The second will be the families and the third will be the rents. The length of all three input fields will be with same length. For each family, you have to print the flat and the rent in the format:

"In flat {number} family {family} has to pay {rent}".

At the end, print the total rent paid in the format: **"Total rent paid: {totalRent}"**

Example

Objects, JSON and DOM Events

03. Rent

Flat Numbers

2, 5, 10

Families

Johnson, Smith, Willson

Rents

200, 150, 400

CALCULATE RENT

Result

In flat 2 family Johnson has to pay 200
In flat 5 family Smith has to pay 150
In flat 10 family Willson has to pay 400
Total rent paid: 750

Hints

- Read the info from the 3 input fields and split them by ", ".

```
let flatNumbers = document.getElementById("flats").value.split(", ");  
let families = document.getElementById("families").value.split(", ");  
let rents = document.getElementById("rents").value.split(", ");
```

- Create an empty array.

```
let result = [];
```

- Since the three fields will have the same length, create a loop until you reach the size of any of them.

```
for (let i = 0; i < flatNumbers.length; i++) {  
    let obj = {};  
    obj["number"] = flatNumbers[i];  
    obj["family"] = families[i];  
    obj["rent"] = Number(rents[i]);  
    result.push(obj);  
}
```

- Now just build the result.

3. JSON Parser

You will be given an object in JSON format in an input field. In the next input field, you will be given a command which can be one of the following: **"typeof {objKey}"** or **"value of {objValue}"**. In any other case, write **"Invalid command"** in the result text box.

If the command is **"typeof {objKey}"**, search for that key in the object and print its type in the format: **"Type: {typeofTheValue}"**. If there is no such key in the object, print **"No such item"**.

If the command is **"value of {objValue}"**, search for that key and print its value in the format: **"Value: {value}"**. If there is no such key, print **"No such item"**.

Example

Objects, JSON and DOM Events

04. JSON Parser

Input

{ "name": "Pesha", "age": 25 }

Command

value of name

EXTRACT INFO

Result

Value: Pesha

Objects, JSON and DOM Events

04. JSON Parser

Input

{ "name": "Pesha", "age": 25 }

Command

typeof age

EXTRACT INFO

Result

type: number

Input

{ "name": "Posho", "age": 25 }

Command

typeof town

EXTRACT INFO

Result

No such item

Hints

- Read the object from the input field, parse it using `JSON.parse`.
- Get the command from the input field, split it into parameters, get the type of the command and the key you will be looking for.

```
function extractInfo() {  
    let str = document.getElementById("input").value;  
    let info = JSON.parse(str);  
    let command = document.getElementById("command").value;  
    let commandParams = command.split(" ");  
    let commandType = commandParams[0];  
    let commandKey = commandParams[commandParams.length - 1];  
}
```

- Check if the command type is about the type or the value of the key.
- Check if you have that key and print the corresponding result in the text box.

```

if (commandType === "typeof") {
    if (info[commandKey]) {
        // TODO: print value
    } else {
        // TODO: no such key
    }
} else if (commandType === "value") {
    if (info[commandKey]) {
        // TODO: print value
    } else {
        // TODO: no such key
    }
} else {
    // TODO: Invalid
}

```

4. Buttons

You will be given **2 input fields**. The first will contain a **key**, the other will contain a **value**. You will also have **three buttons**. The first will be **"Add"**. When this button is clicked, you have to **add the property with the given key and value** to an object and display a message in the text box: **"New property added!"** The second will be **"Get Value"**. When this button is clicked, you have to **search for the key, given in the input, and display its value in the text box**. The last one will be **"Get Object"**. When this button is clicked, you have to display the **whole object in the text box**. **The text box must not be cleared each time you press a button**. You just have to **print the results on separate lines**.

Example

Objects, JSON and DOM Events

05. Buttons

Key

name

Value

Pesho

ADD

GET VALUE

GET WHOLE OBJECT

Console

New property added!

Objects, JSON and DOM Events

05. Buttons

Key

age

Value

20

ADD

GET VALUE

GET WHOLE OBJECT

Console

New property added!

New property added!

Objects, JSON and DOM Events

05. Buttons

Key

Value

Console

```
New property added!  
New property added!  
20
```

Objects, JSON and DOM Events

05. Buttons

Key

Value

Console

```
New property added!  
New property added!  
20  
{ "name": "Pesho", "age": "20" }
```

Hints

- Add onclick events on the buttons to execute the functions, depending on the button you have clicked.

```

<div class="center">
  <input type="submit" class="button" onclick="" value="Add">
  <input type="submit" class="button" onclick="" value="Get Value">
  <input type="submit" class="button" onclick="" value="Get Whole
  Object">
</div>

```

- Create an object and write three functions to perform the different tasks.

```

<script>
  document.querySelectorAll(".button").forEach(element => {
    element.addEventListener("click", function(event) {
      event.preventDefault()
    });
  });
  let obj = {};
  function addProperty() {
    // TODO: get key and value and add them to the object; display message
  }
  function getProperty() {
    // TODO: get the value by the given key and display it
  }
  function getObject() {
    // TODO: stringify the object and display it
  }
</script>

```