

## Contents

Day-1 .....	4
1.HTML ATTRIBUTES: .....	4
2. HTML INPUT TAGS: .....	5
CSS:.....	6
INLINE CSS: .....	6
INTERNAL CSS: .....	7
EXTERNAL CSS: .....	8
CSS SELECTORS:.....	9
CSS ID SELECTORS: .....	10
CLASS SELECTOR: .....	11
UNIVERSAL SELECTOR: .....	12
CSS VARIABLES .....	13
Local Variables: .....	13
Override/Global Variables: .....	13
DAY-2 .....	14
Internal Js:.....	14
External Js: .....	15
Functions.....	16
SCOPE:.....	16
LOCAL SCOPE: .....	16
GLOBAL SCOPE: .....	17
ES6 STANDARDS:.....	17
let: .....	17
Const: .....	18
Arrow Function: .....	18
Promises:.....	19
JS CLASSES:.....	19
JS class syntax: .....	19
Js Class Methods: .....	20
JavaScript Event Handlers .....	20
addEventListener .....	21

remove: .....	21
DAY-3 .....	22
<i>What are the different types of event handlers provided by JavaScript?</i> .....	22
Pass by Value: .....	24
Pass by Reference: .....	25
JS ASYNC/AWAIT: .....	29
Async/Await: .....	29
Object Keys: .....	30
Object Entries:.....	32
HTML_JS_CUSTOM ATTRIBUTES:.....	34
EXAMPLE-1:.....	34
EXAMPLE-2.....	35
EXAMPLE:3 .....	36
DAY-4 .....	37
LOCAL STORAGE:.....	37
Example1:.....	37
Objects: .....	39
Creating a JavaScript Object .....	39
Using an Object Literal.....	39
Using the JavaScript Keyword new .....	40
Example1:.....	40
Example2:.....	41
Example3:.....	41
Example4:.....	41
ArrayIteration:.....	43
arrayForEach: .....	43
ArrayMap: .....	44
ArrayFilter: .....	44
ArrayReduce:.....	45
ArrayIndexOf:.....	45
ArrayLastIndexOf: .....	46
ArrayFind:.....	46
CreateElement .....	47

JSON: .....	48
Why Use JSON? .....	48
Storing Data .....	48
JSON PARSE: .....	49
Example1: .....	49
Example2: .....	49
Example3: .....	50
JSON Stringify: .....	50
Stringify object: .....	50
Example-1: .....	50
Stringify Array: .....	51
Example-2: .....	51
JS STRING PROPERTIES: .....	52
Sample example: .....	52
Example1: .....	52
JS STRING METHODS: .....	52
indexOf(): .....	52
Example2: .....	52
lastIndexOf(): .....	53
Example3: .....	53
Replace(): .....	54
Example4: .....	54
Slice(): .....	54
Example5: .....	54
toString(): .....	55
Example6: .....	55
toLowerCase(): .....	55
Example7: .....	55
toUpperCase(): .....	56
Example8: .....	56
JS Current Date and Time: .....	57

## Objects:

### Creating a JavaScript Object

With JavaScript, you can define and create your own objects.

There are different ways to create new objects:

- Create a single object, using an object literal.
- Create a single object, with the keyword `new`.
- Define an object constructor, and then create objects of the constructed type.
- Create an object using `Object.create()`.

### Using an Object Literal

This is the easiest way to create a JavaScript Object.

Using an object literal, you both define and create an object in one statement.

An object literal is a list of name:value pairs (like age:50) inside curly braces `{}`.

The following example creates a new JavaScript object with four properties:

```
const person = {firstName:"John", lastName:"Doe", age:50,  
eyeColor:"blue"};
```

Spaces and line breaks are not important. An object definition can span multiple lines:

```
const person = {  
  firstName: "John",  
  lastName: "Doe",  
  age: 50,  
  eyeColor: "blue"  
};
```

This example creates an empty JavaScript object, and then adds 4 properties:

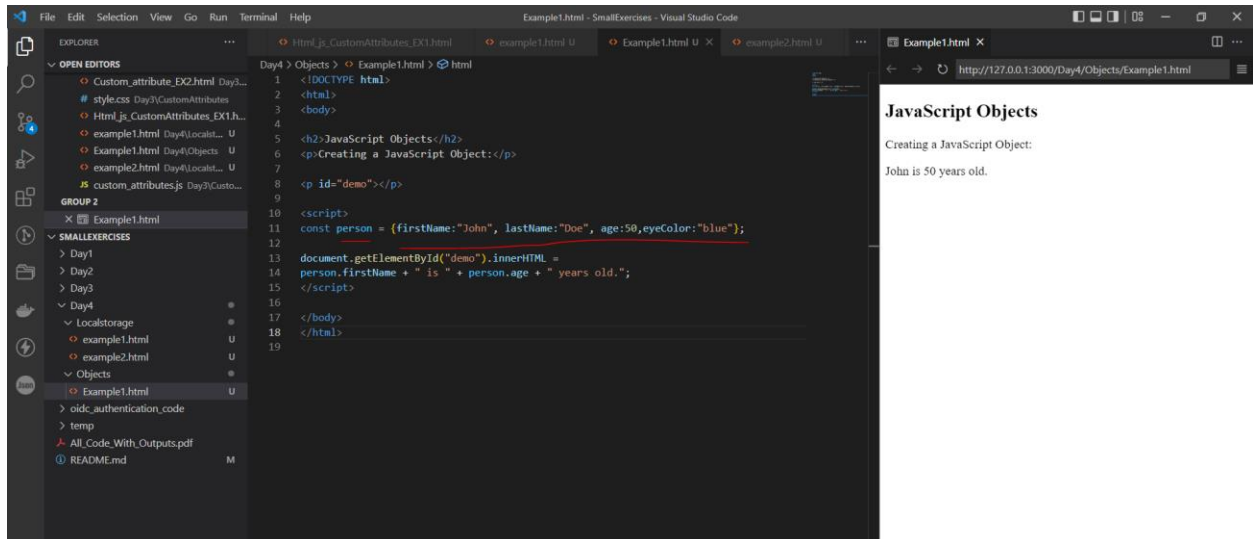
```
const person = {};  
person.firstName = "John";  
person.lastName = "Doe";  
person.age = 50;  
person.eyeColor = "blue";
```

Using the JavaScript Keyword new

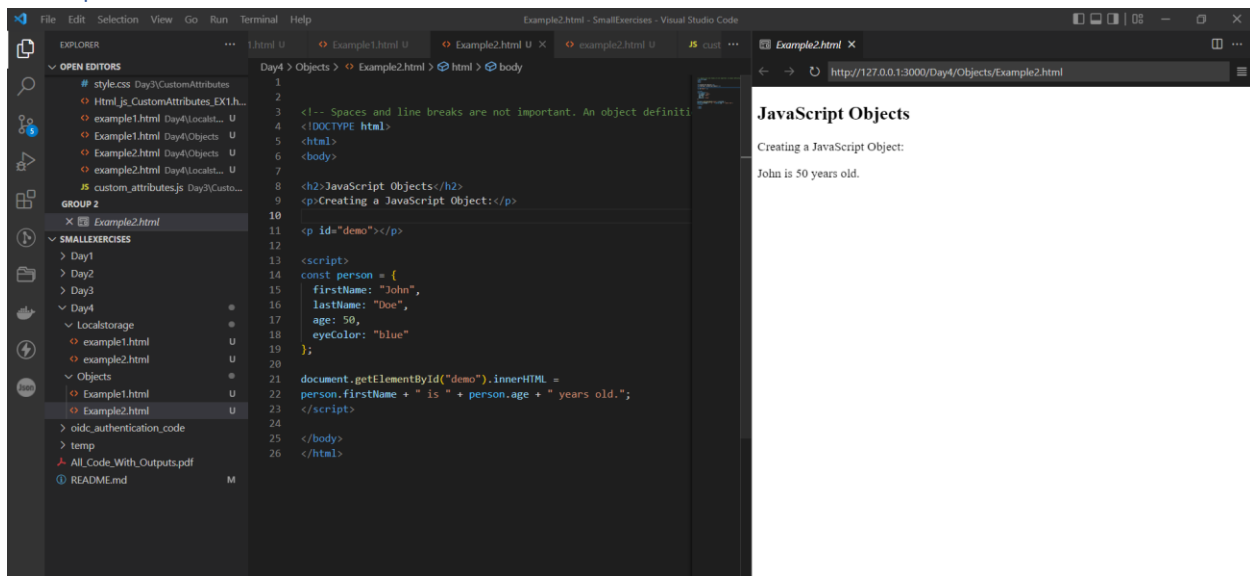
The following example create a new JavaScript object using `new Object()`, and then adds 4 properties:

```
const person = new Object();  
person.firstName = "John";  
person.lastName = "Doe";  
person.age = 50;  
person.eyeColor = "blue";
```

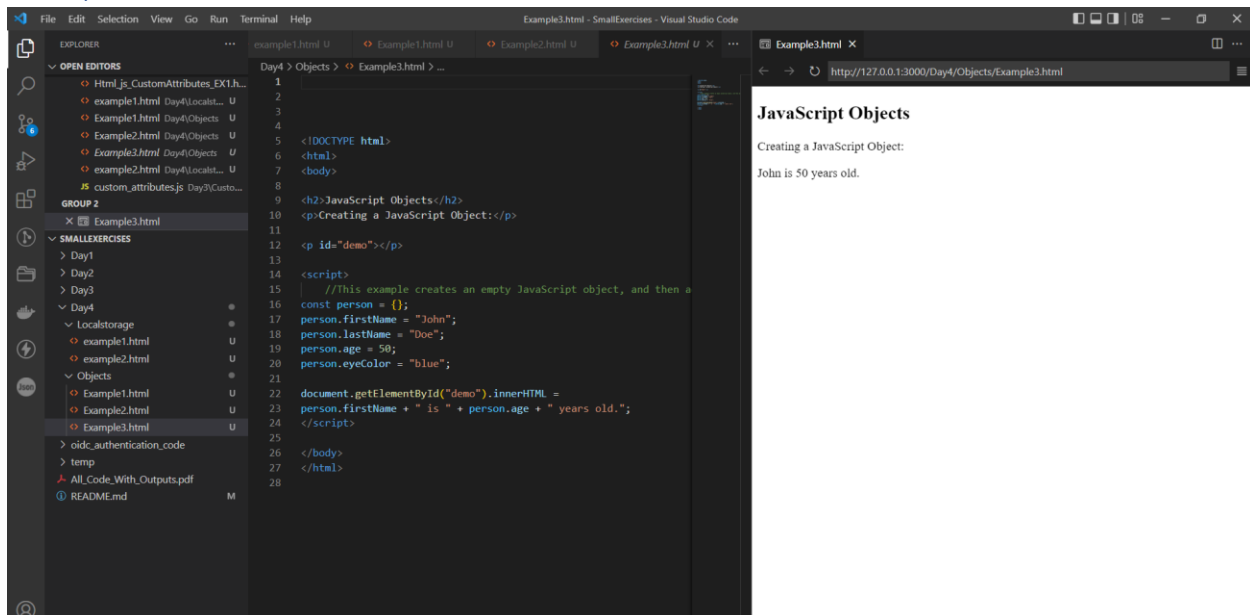
Example1:



## Example2:



## Example3:



## Example4:

