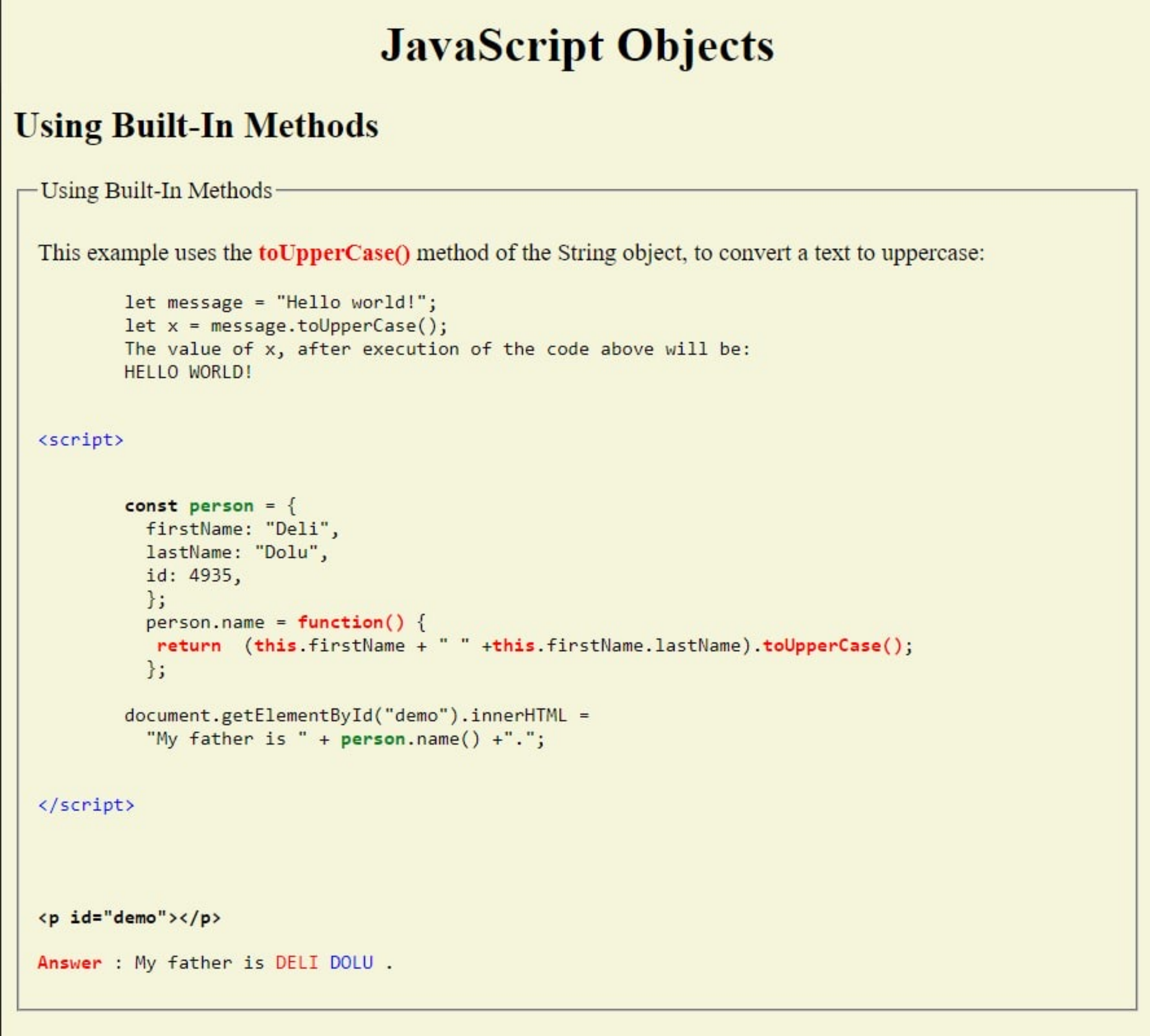


JavaScript Objects

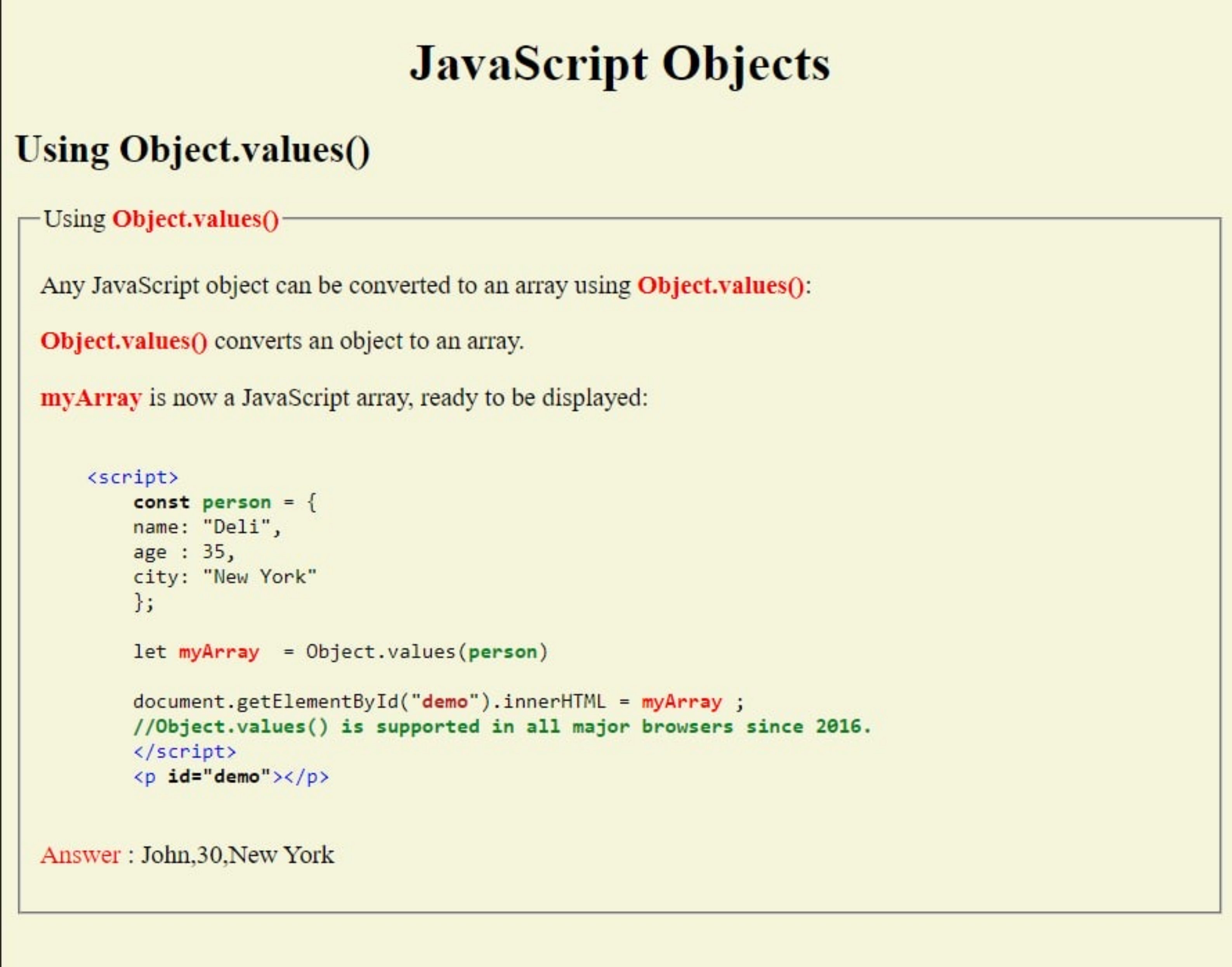
Using Built-In Methods

Using Built-In Methods

This example uses the **toUpperCase()** method of the String object, to convert a text to uppercase:

```
let message = "Hello world!";
let x = message.toUpperCase();
The value of x, after execution of the code above will be:
HELLO WORLD!
```

```
<script>


        const person = {
          firstName: "Deli",
          lastName: "Dolu",
          id: 4935,
          };
          person.name = function() {
           return  (this.firstName + " " +this.firstName.lastName).toUpperCase();
          };

        document.getElementById("demo").innerHTML =
          "My father is " + person.name() +".";


</script>




<p id="demo"></p>

Answer : My father is DELI DOLU .
```



JavaScript Objects

Using Object.values()

Using **Object.values()**

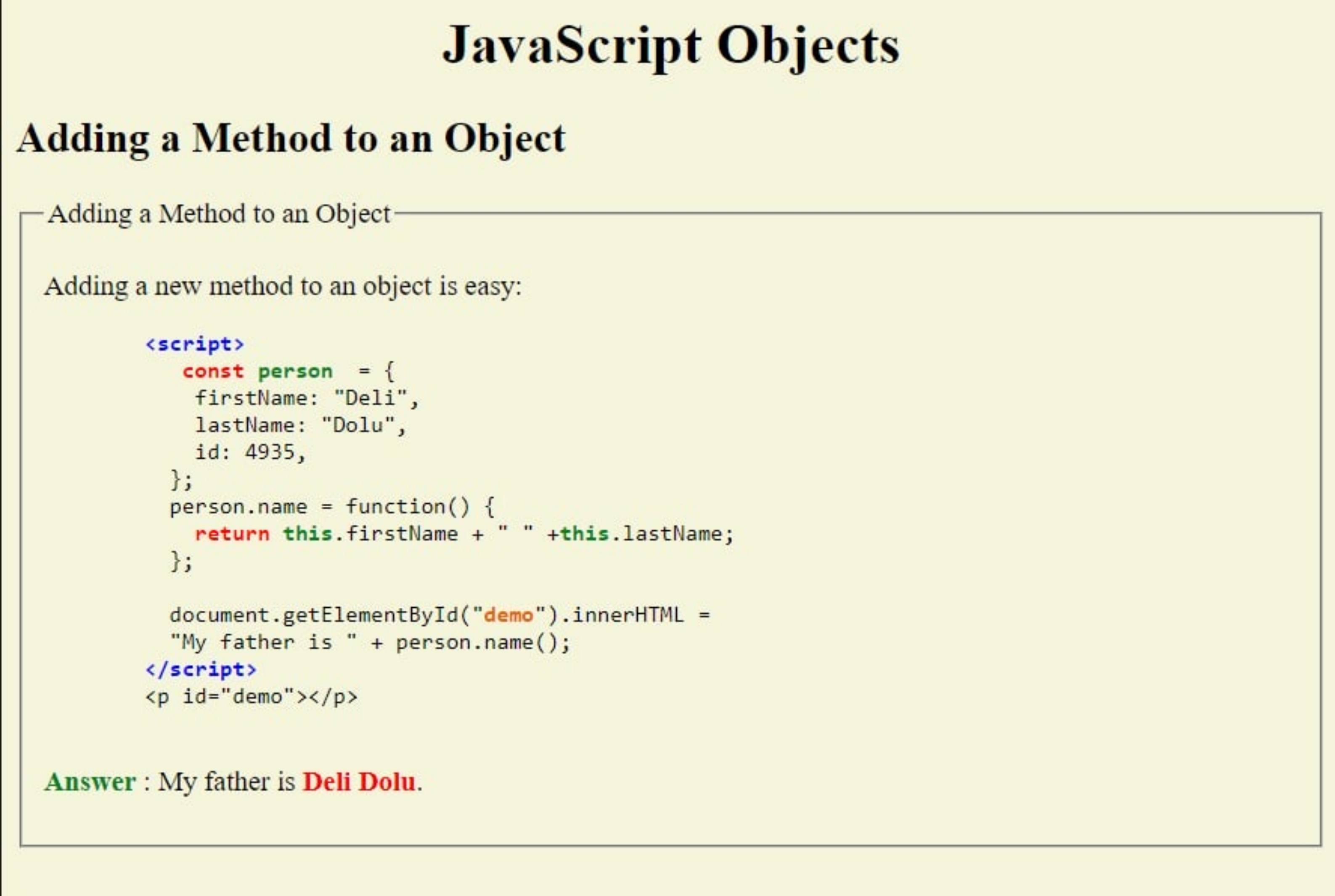
Any JavaScript object can be converted to an array using **Object.values()**:

Object.values() converts an object to an array.

myArray is now a JavaScript array, ready to be displayed:

```
<script>
    const person = {
    name: "Deli",
    age : 35,
    city: "New York"
    };

    let myArray  = Object.values(person)

    document.getElementById("demo").innerHTML = myArray ;
    //Object.values() is supported in all major browsers since 2016.
    </script>
    <p id="demo"></p>
```

Answer : John,30,New York



JavaScript Objects

Adding a Method to an Object

Adding a Method to an Object

Adding a new method to an object is easy:

```
<script>
   const person  = {
    firstName: "Deli",
    lastName: "Dolu",
    id: 4935,
  };
  person.name = function() {
    return this.firstName + " " +this.lastName;
  };

  document.getElementById("demo").innerHTML =
  "My father is " + person.name();
</script>
<p id="demo"></p>
```

Answer : My father is **Deli Dolu**.

JavaScript Objects

Adding a Method to an Object

Adding a Method to an Object

Adding a new method to an object is easy:

```
<script>
   const person  = {
    firstName: "Deli",
    lastName: "Dolu",
    id: 4935,
  };
  person.name = function() {
    return this.firstName + " " +this.lastName;
  };

  document.getElementById("demo").innerHTML =
  "My father is " + person.name();
</script>
<p id="demo"></p>
```

Answer : My father is **Deli Dolu**.

JavaScript Display Objects

Stringify Functions

Stringify Functions

JSON.stringify will not stringify functions:

```
    <script>
        const person= {
            name : "John",
            age  : function ()  {return 30;}
            };

        document.getElementById("demo").innerHTML = JSON.stringify(person);
        </script>
        <p  id="demo"></p>
```

Answer: {"name":"John"}

This can be "fixed" if you convert the functions into strings before stringifying.

Stringify Functions

```
    <script>
        const person= {
            name : "John",
            age  : function ()   {return 30;}
            };

            person.age = person.age.toString();

            let  myString = JSON.stringify(person);
            document.getElementById("demo").innerHTML =  myString;
        </script>
        <p  id="demo"></p>
```

Answer: {"name":"John","age":"function () {\n return 30;\n }"}

JavaScript Objects

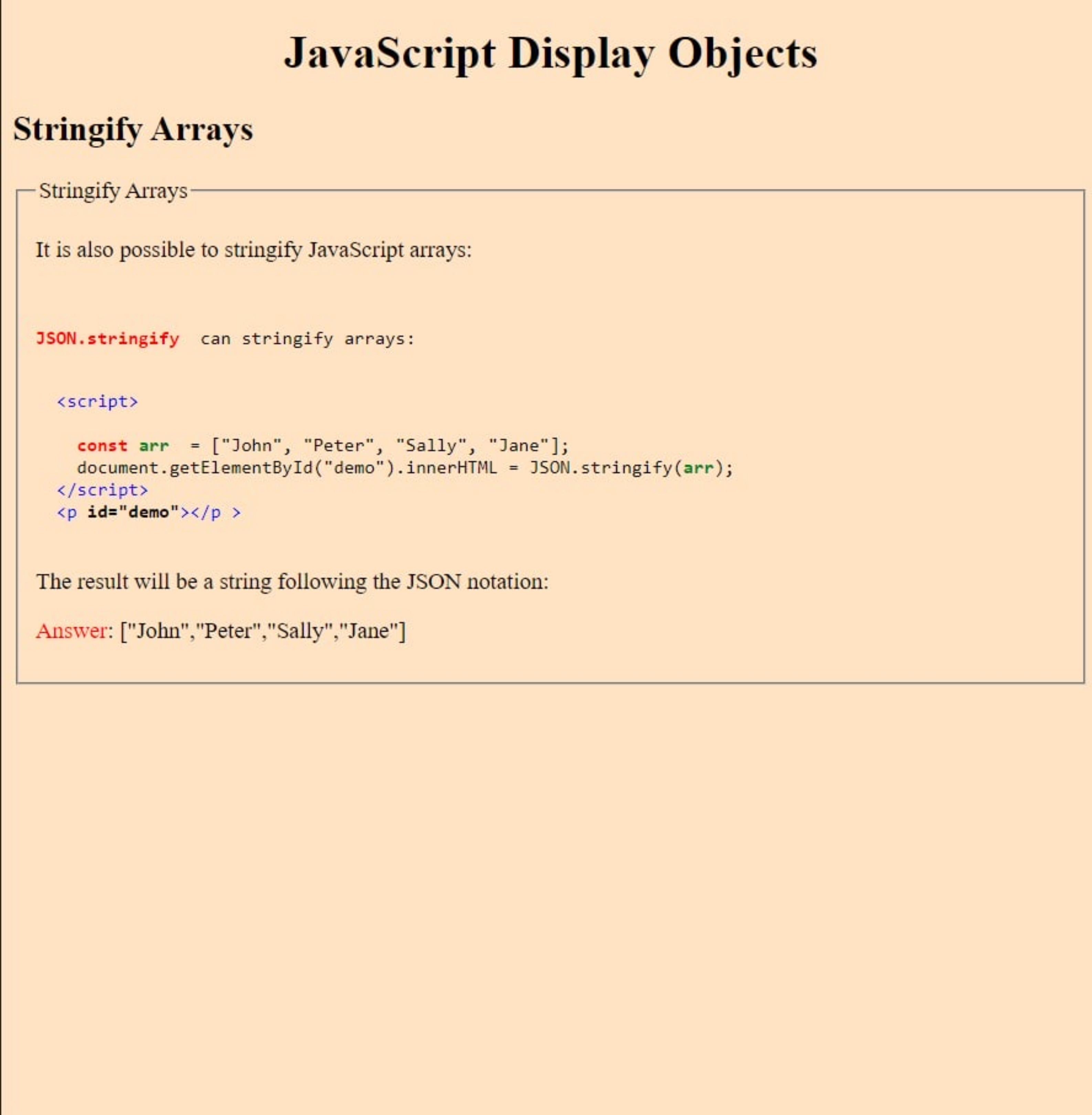
Using JSON.stringify()

Using JSON.stringify()

Any JavaScript object can be stringified (converted to a string) with the JavaScript function **JSON.stringify()** :

```
Display properties in JSON format:

  <script>
    const person = {
      name: "John",
      age: 30,
      city: "New York"
    };

    document.getElementById("demo").innerHTML = JSON.stringify(person);

      /*The result will be a string following the JSON notation:
      {"name":"John","age":50,"city":"New York"}*/

    JSON.stringify()  is included in JavaScript and supported in all major browsers.

  </script>
  <p id="demo"></p>
```

Answer : {"name":"John","age":30,"city":"New York"}



JavaScript Display Objects

Stringify Arrays

Stringify Arrays

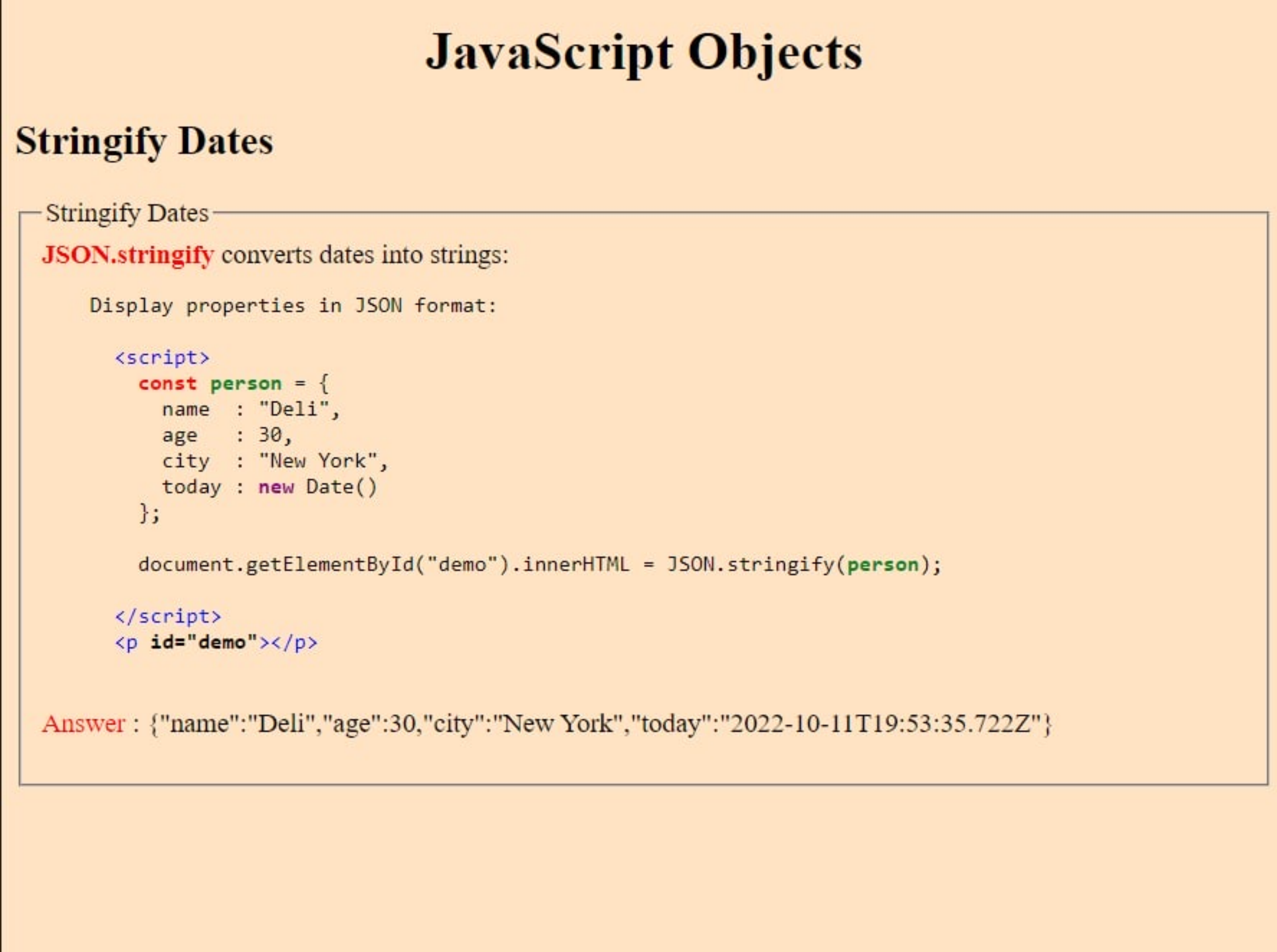
It is also possible to stringify JavaScript arrays:

```
JSON.stringify  can stringify arrays:
```

```
  <script>

    const arr  = ["John", "Peter", "Sally", "Jane"];
    document.getElementById("demo").innerHTML = JSON.stringify(arr);
  </script>
  <p id="demo"></p >
```

The result will be a string following the JSON notation:

Answer: ["John","Peter","Sally","Jane"]



JavaScript Objects

Stringify Dates

Stringify Dates

JSON.stringify converts dates into strings:

```
Display properties in JSON format:

  <script>
    const person = {
      name  : "Deli",
      age   : 30,
      city  : "New York",
      today : new Date()
    };

    document.getElementById("demo").innerHTML = JSON.stringify(person);

  </script>
  <p id="demo"></p>
```

Answer : {"name":"Deli","age":30,"city":"New York","today":"2022-10-11T19:53:35.722Z"}