Fakultas Ilmu Komputer

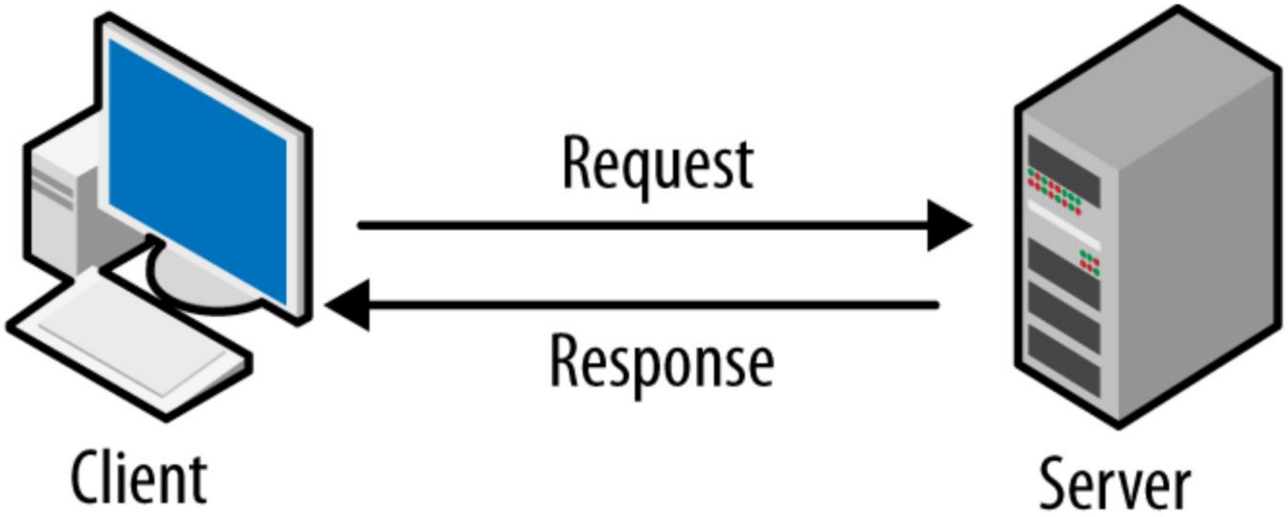
Learning is the process of acquiring new understanding, knowledge, behaviors, skills, values, attitudes, and preferences.

JavaScript Object Notation (JSON)

Welcome to *the* Front-end Web Development *course*

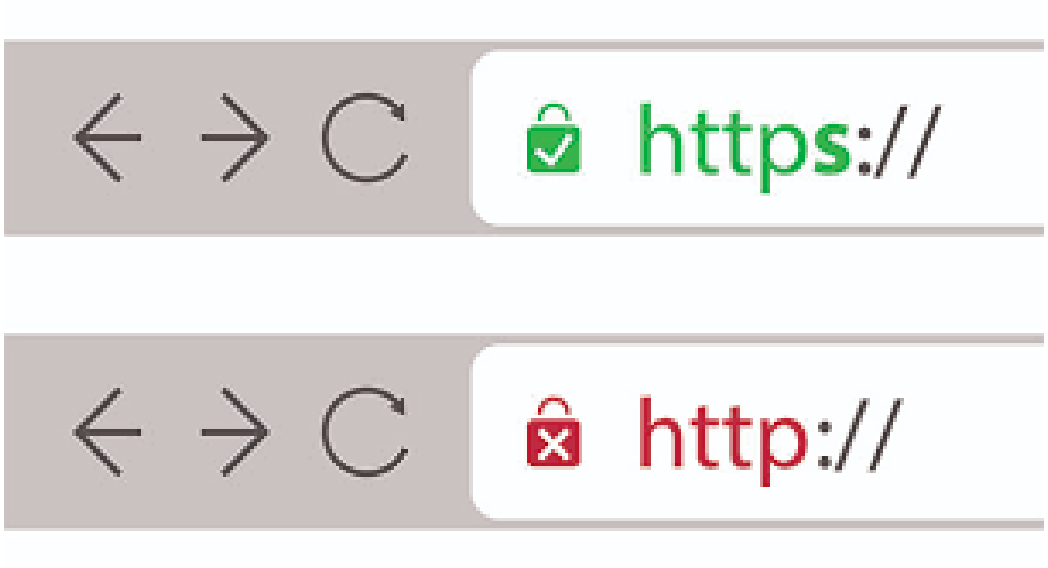
Client Server Communications

Client-server communication is a fundamental concept in computer networking and software architecture. Client-server communication refers to the interaction between two types of entities: clients and servers. **Clients request** and consume resources or services provided by servers.



A common use of JSON is to exchange data to/from a web server. When sending data to a web server, the data has to be a string.

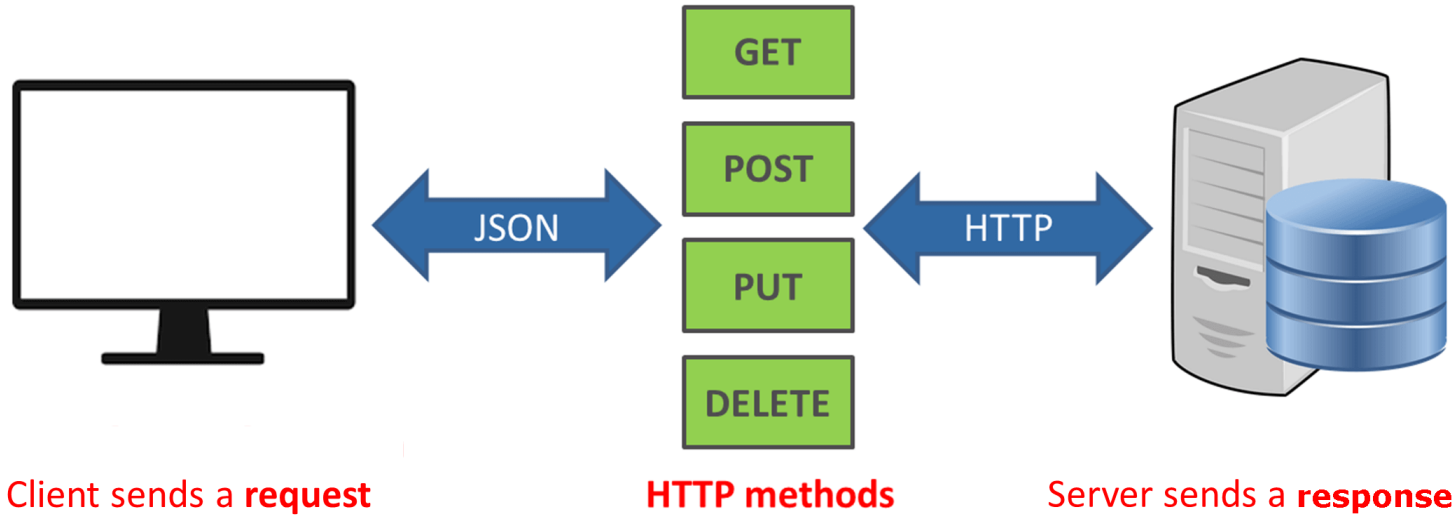
HTTP/HTTPS



Hypertext Transfer Protocol (**HTTP**)
Hypertext Transfer Protocol Secure (**HTTPs**)

Used for web browsing and communication between web browsers and web servers. HTTP is the standard protocol, while HTTPS adds security through encryption.

REST & APIs



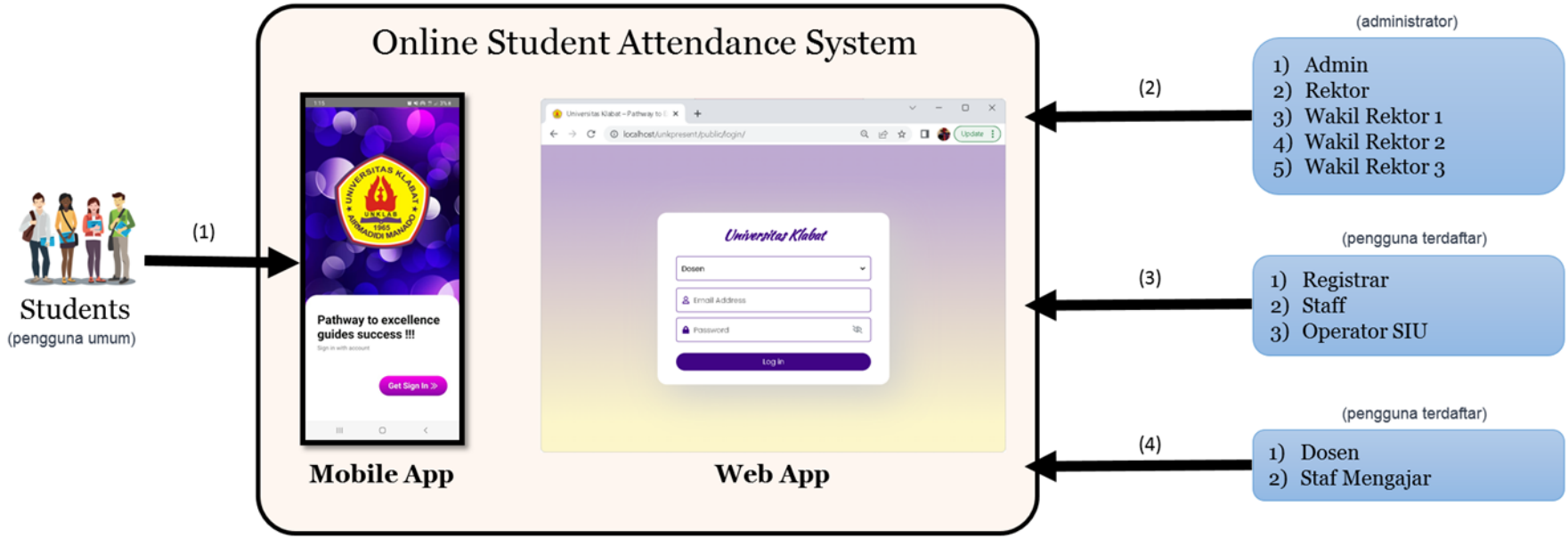
Representational State Transfer (REST) is an architectural style for designing networked applications. It involves using standard HTTP methods (GET, POST, PUT, DELETE) to interact with resources on servers.

RESTful APIs provide a structured way for clients to communicate with servers, typically **exchanging data** in **JSON** or **XML format**.

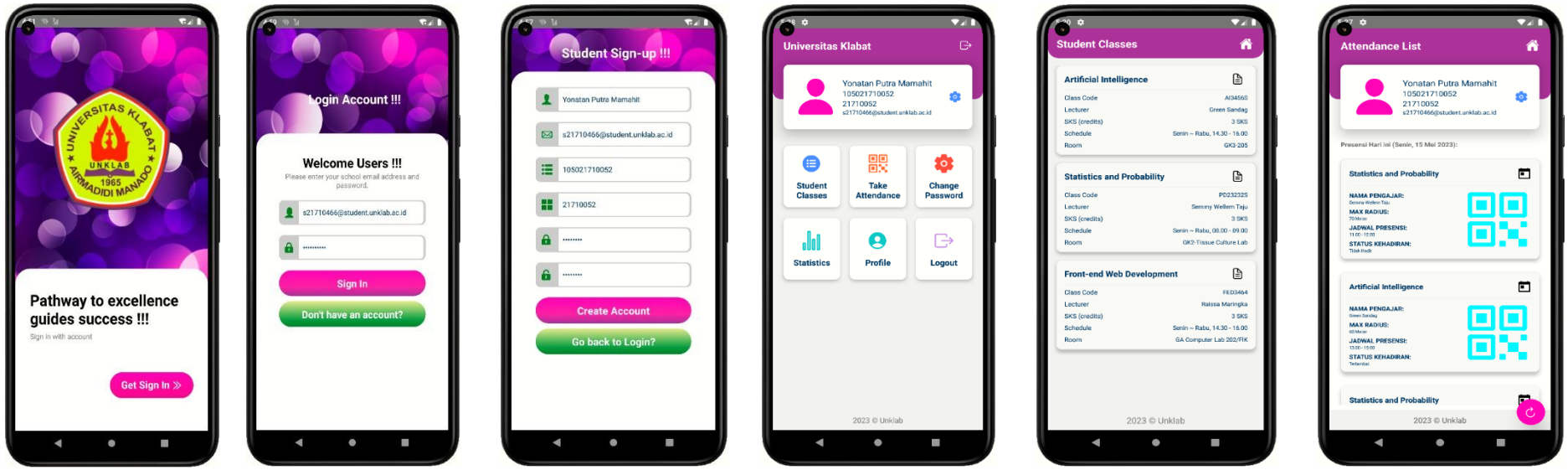
Apakah kalian sudah pernah mendengar istilah JSON? apabila kalian seorang developer, pastinya sudah tidak asing lagi dengan istilah JSON. JSON atau Javascript Object Notation biasanya sering digunakan untuk **pertukaran data antar aplikasi** dalam instansi tertentu. **Hasil atau response dalam Restful API** yang diberikan dalam aplikasi tersebut biasanya dalam bentuk JSON yang nantinya akan ditampilkan kepada *user* atau *pengguna*.

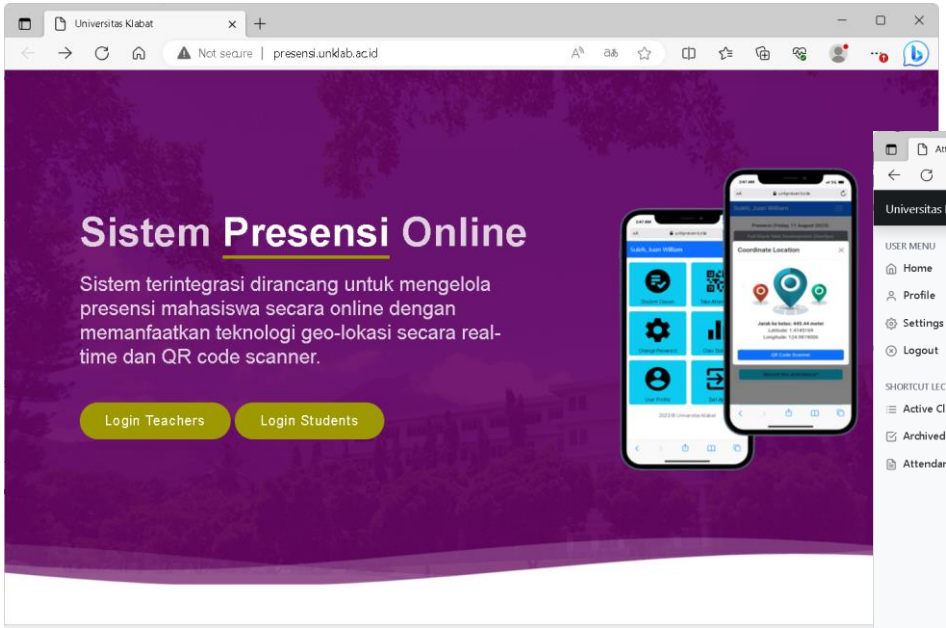
Untuk melakukan **pertukaran data** antar *aplikasi* maupun *server*, anda dapat menggunakan JSON (Javascript Object Notation).

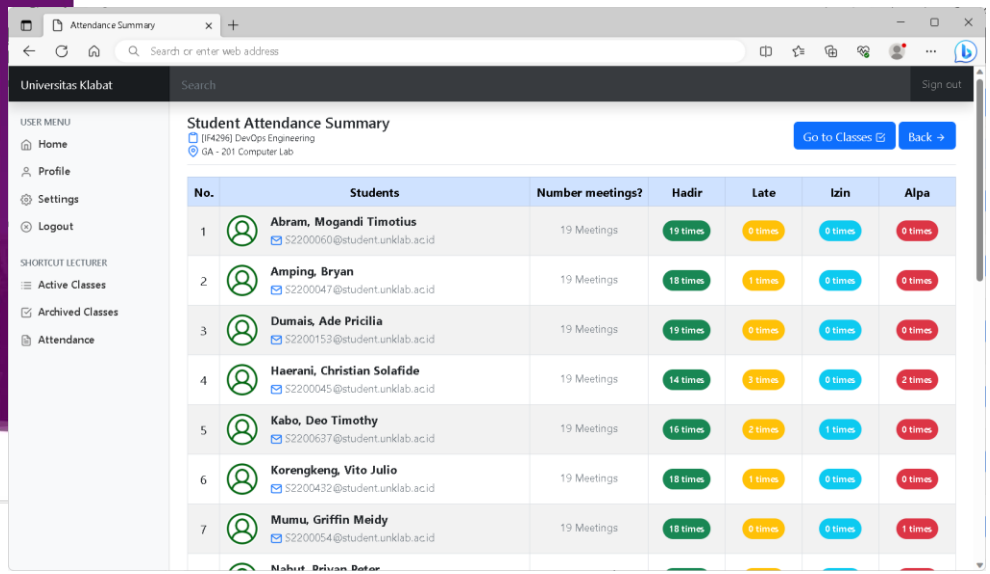
Format JSON dibuat berdasarkan bagian dari Bahasa Pemprograman JavaScript. **JSON merupakan format teks** yang tidak bergantung pada bahasa pemprograman apapun karena menggunakan gaya bahasa yang umum digunakan oleh programmer keluarga C termasuk C, C++, C#, Java, JavaScript, Perl, Python dll.

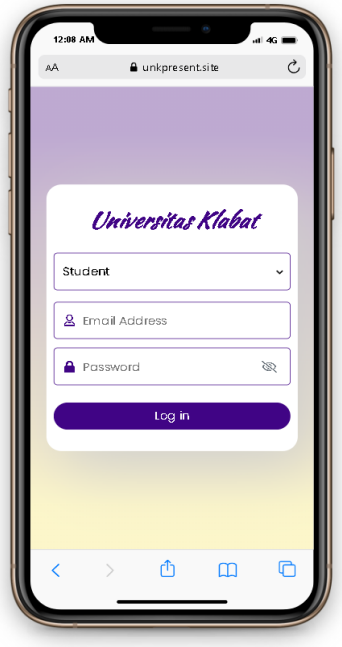


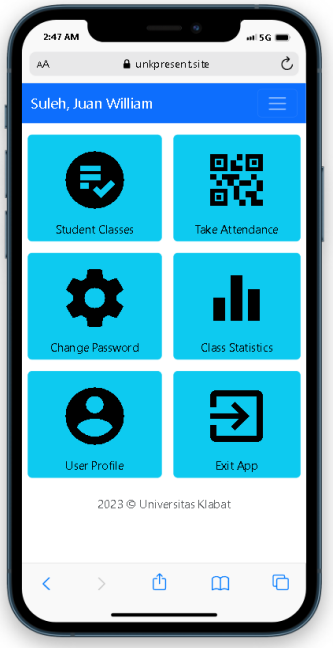
1) **Pengguna umum** hanya memiliki akses terbatas ke fitur dan informasi dalam sistem.
2) **Pengguna terdaftar** memiliki hak akses lebih seperti mengunggah atau mengedit.
3) **Administrator** memiliki hak akses penuh dan dapat mengatur dan mengelola sistem secara keseluruhan.



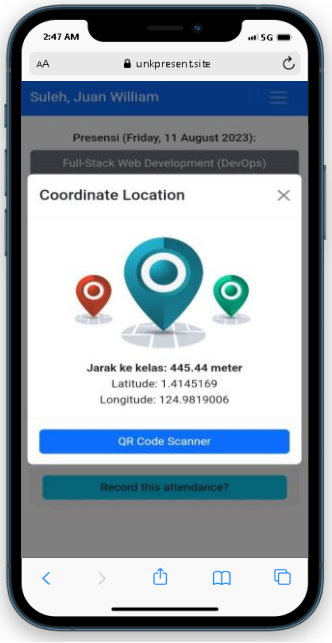

Universitas Klabat
Not secure | presensi.unklab.ac.id
Sistem Presensi Online
Sistem terintegrasi dirancang untuk mengelola presensi mahasiswa secara online dengan memanfaatkan teknologi geo-lokasi secara real-time dan QR code scanner.
Login Teachers
Login Students


Attendance Summary
Universitas Klabat
Search
Sign out
USER MENU
Home
Profile
Settings
Logout
SHORTCUT LECTURER
Active Classes
Archived Classes
Attendance
Student Attendance Summary
(IF4296) DevOps Engineering
GA - 201 Computer Lab
Go to Classes
Back
No. | Students | Number meetings? | Hadir | Late | Izin | Alpa
1 | Abram, Mogandi Timotius | S2200060@student.unklab.ac.id | 19 Meetings | 19 times | 0 times | 0 times | 0 times
2 | Amping, Bryan | S2200047@student.unklab.ac.id | 19 Meetings | 18 times | 1 times | 0 times | 0 times
3 | Dumais, Ade Pricilia | S2200153@student.unklab.ac.id | 19 Meetings | 19 times | 0 times | 0 times | 0 times
4 | Haerani, Christian Solafide | S2200045@student.unklab.ac.id | 19 Meetings | 14 times | 3 times | 0 times | 2 times
5 | Kabo, Deo Timothy | S2200637@student.unklab.ac.id | 19 Meetings | 16 times | 2 times | 1 times | 0 times
6 | Korengkeng, Vito Julio | S2200432@student.unklab.ac.id | 19 Meetings | 18 times | 1 times | 0 times | 0 times
7 | Mumu, Griffin Meidy | S2200054@student.unklab.ac.id | 19 Meetings | 18 times | 0 times | 0 times | 1 times


12:08 AM
unkpresent.site
Universitas Klabat
Student
Email Address
Password
Log in


2:47 AM
unkpresent.site
Suleh, Juan William
Student Classes
Take Attendance
Change Password
Class Statistics
User Profile
Exit App
2023 © Universitas Klabat


2:47 AM
unkpresent.site
Suleh, Juan William
Computer Vision
Code CSAR452
Lecturer Taju, Semmy Wellem
Credits 3 SKS
Schedule Senin-Selasa-Rabu-Kamis-Jumat, 15.10 - 17.00
Room GA - 201 Computer Lab
Max_Absent 4
Full-Stack Web Development (DevOps)
Code SENG433
Lecturer Taju, Semmy Wellem
Credits 3 SKS
Schedule Senin-Selasa-Rabu-Kamis-Jumat, 10.10 - 12.00
Room GA - 201 Computer Lab
Max_Absent 4


2:47 AM
unkpresent.site
Suleh, Juan William
Presensi (Friday, 11 August 2023):
Full-Stack Web Development (DevOps)
Coordinate Location
Jarak ke kelas: 445.44 meter
Latitude: 1.4145169
Longitude: 124.9819006
QR Code Scanner
Record this attendance?


2:47 AM
unkpresent.site
Full-Stack Web Development (DevOps)
Select Camera (4) camera2 2, facing back
Stop Scanning
2023 © Universitas Klabat

Machine Learning Using Json Data

```
sample_data = [{
  'Scaler': 'Standard',
  'Results': [ { 'max_depth': 2,
            'scores': { 'accuracy': '0.91' } },
          { 'max_depth': 5,
            'scores': { 'accuracy': '0.96' } },
      ]},
 {
  'Scaler': 'MinMax',
  'Results': [{ 'max_depth': 2,
            'scores': { 'accuracy': '0.93' }
          },
          { 'max_depth': 5,
            'scores': { 'accuracy': '0.97' } },
      ]}
]
```

```
pandas.json_normalize(
    sample_data,
    record_path=['Results'],
    meta=['Scaler'],
    meta_prefix='config_params_',
    record_prefix='random_forest_'
)
```

normalized_data

random_forest max_depth	random_forest scores.accuracy	config_params _Scaler
2	0.91	Standard
5	0.96	Standard
2	0.93	MinMax
5	0.97	MinMax

1 - JSON - Introduction

JSON - Introduction

- **JSON** stands for **J**ava**S**cript **O**bject **N**otation.
- **JSON** is a text format for storing and transporting data.
- **JSON** is "self-describing" and easy to understand.

What is JSON?

- JSON stands for JavaScript Object Notation.
- JSON is a lightweight data-interchange format.
- JSON is plain *text written* in JavaScript object notation.
- JSON is used to send data between computers.
- JSON is language independent. *

Why Use JSON?

- The JSON format is syntactically similar to the code for creating JavaScript objects. Because of this, a JavaScript program can easily convert JSON data into JavaScript objects.
- Since the format is text only, JSON data can easily be sent between computers, and used by any programming language.

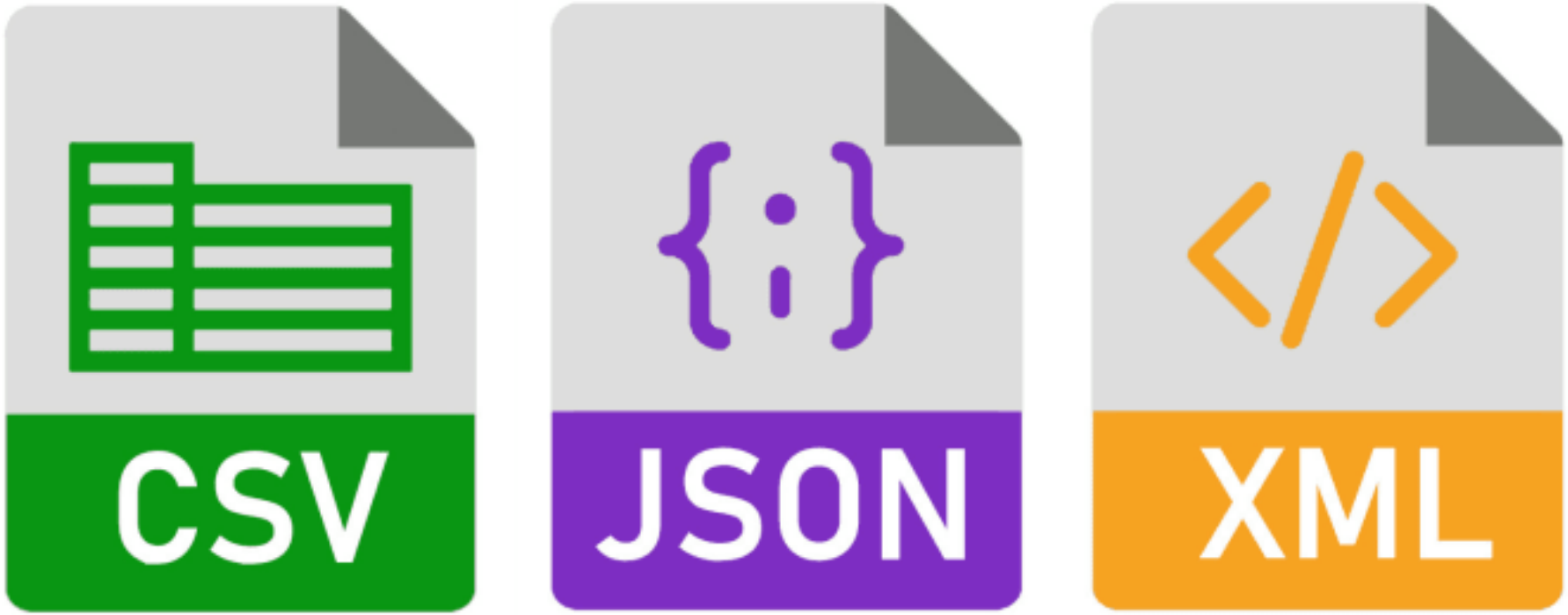
JavaScript has a built in function for **converting** JSON strings into JavaScript objects:
JSON.parse()

JavaScript also has a built in function for **converting** an object into a JSON string:
JSON.stringify()

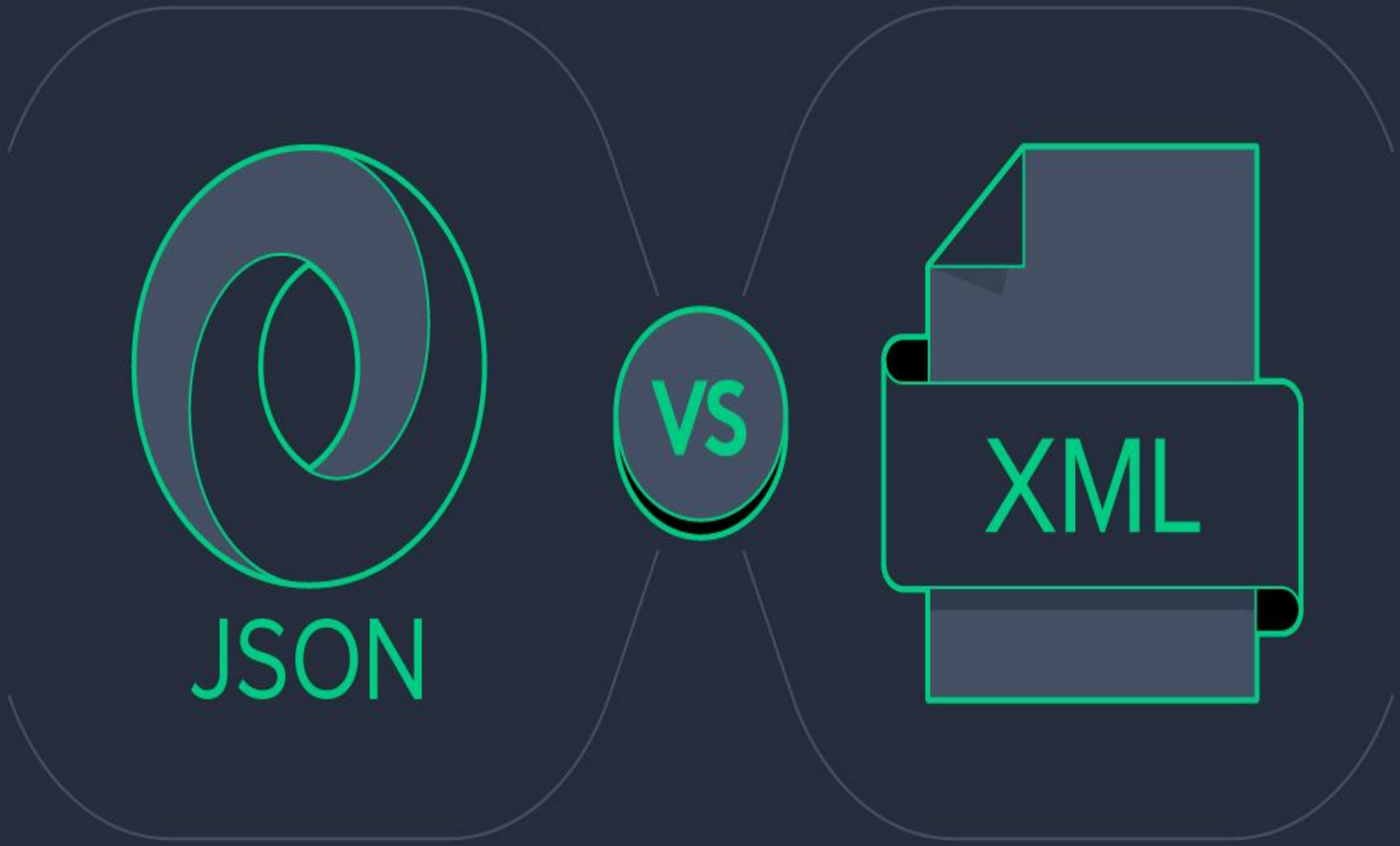
JSON Example

SampleRecords.json

```
[
    {
        "trackid": "AA-1234",
        "reported_dt": "12/31/2019 23:59:59",
        "longitude": -111.12500000,
        "latitude": 33.37500000
    },
    {
        "trackid": "BB-7890",
        "reported_dt": "12/31/2019 23:59:59",
        "longitude": -113.67500000,
        "latitude": 35.87500000
    },
    {
        "trackid": "CC-4545",
        "reported_dt": "12/31/2019 23:59:59",
        "longitude": -115.57500000,
        "latitude": 37.67500000
    }
]
```



- ✓ CSV file (comma separated value)
- ✓ JSON file (JavaScript Object Notation)
- ✓ XML file (Extensible Markup Language)


JSON
VS
XML

JSON *vs* XML

Both JSON and XML can be used to receive data from a web server.

JSON Example

```
{"employees":[
  { "firstName":"John", "lastName":"Doe" },
  { "firstName":"Anna", "lastName":"Smith" },
  { "firstName":"Peter", "lastName":"Jones" }
]}
```

XML Example

```
<employees>
  <employee>
    <firstName>John</firstName> <lastName>Doe</lastName>
  </employee>
  <employee>
    <firstName>Anna</firstName> <lastName>Smith</lastName>
  </employee>
  <employee>
    <firstName>Peter</firstName> <lastName>Jones</lastName>
  </employee>
</employees>
```

JSON *vs* XML

XML vs. JSON

```xml
<?xml version="1.0" encoding="UTF-8"?>
<endereco>
    <cep>31270901</cep>
    <city>Belo Horizonte</city>
    <neighborhood>Pampulha</neighborhood>
    <service>correios</service>
    <state>MG</state>
    <street>Av. Presidente Antônio Carlos, 6627</street>
</endereco>
```

```json
{
    "endereco": {
        "cep": "31270901",
        "city": "Belo Horizonte",
        "neighborhood": "Pampulha",
        "service": "correios",
        "state": "MG",
        "street": "Av. Presidente Antônio Carlos, 6627"
    }
}
```

JSON против XML	
JSON stands for JavaScript Object Notation and is based on JavaScript language.	XML is short for Extensive Markup Language and is derived from SGML.
It supports text and number data types including integer and strings, and arrays and objects.	It has no direct support for array.
It's a language-independent data-interchange format which supports only UTF-8 encoding.	It's an independent data format which supports different encodings.
It does not contain start and end tags.	It contains start and end tags.
It does not support native objects.	It gets support of objects via attributes and elements.
No support for Namespaces.	Namespaces are supported in XML.

JSON is Like XML Because

- Both JSON and XML are "self describing" (human readable)
- Both JSON and XML are hierarchical (values within values)
- Both JSON and XML can be parsed and used by lots of programming languages
- Both JSON and XML can be fetched with an XMLHttpRequest

JSON is Unlike XML Because

- JSON doesn't use end tag
- JSON is shorter
- JSON is quicker to read and write
- JSON can use arrays

Brief History *of* JSON

JSON pertamakali dipopulerkan oleh **Douglas Crockford**. Seorang *software engineer* yang juga terlibat dalam pengembangan bahasa pemrograman Javascript.

JSON tidak ditemukan oleh satu orang, dulu namanya belum JSON. Artinya kata “JSON” belum ada. Orang-oarang hanya mengenal Objek Javascript yang dikirim melalui jaringan. Sejak meledaknya teknologi AJAX pada tahun 2000. JSON mulai diperkenalkan dan pada tahun 2001, domain *json.org* mulai terkenal. Hingga saat ini JSON banyak digunakan di mana-mana.

Elemen JSON (Javascript Object Notation)

Elemen didalam JSON atau Javascript Object Notation tersusun dari dua struktur, yaitu kumpulan pasangan nilai/nama dan daftar nilai terurutkan. Kedua elemen tersebut yaitu berikut ini.

1) Kumpulan pasangan nama/nilai, pada bahasa pemrograman lain, pasangan nama / nilai ini sering disebut sebagai object (*objek*), record (*rekaman*), struct (*struktur*), dictionary (*kamus*), hash table (*tabel hash*), keyed list (*daftar kunci*) atau associative array.

2) Daftar nilai terurutkan (*an ordered list of value*), pada bahasa pemrograman lain, ordered list of value ini biasa disebut juga sebagai array (*larik*), vector (*vektor*), list (*daftar*), atau sequence (*urutan*).

Struktur data diatas disebut juga sebagai ***struktur data universal***, karena pada dasarnya semua bahasa pemrograman yang klasik maupun modern mendukung struktur tersebut baik dalam format yang sama ataupun berbeda. Berdasarkan struktur data ini, maka format data mudah dipertukarkan dengan bahasa-bahasa pemrograman yang lain. Selain menggunakan JSON, anda juga bisa menggunakan XML dalam pertukaran data.

Kelebihan Menggunakan JSON

- Kecepatan dalam penguraian yang merupakan proses pengenalan bagian terkecil dari suatu dokumen JSON/XML sehingga membuat kecepatan penguraian pada JSON melampaui XML.
- Kemampuan untuk menyimpan data dalam bentuk array yang memungkinkan transfer menjadi lebih mudah.
- Berdasar pada JavaScript membuat JSON memiliki sintaks yang kecil dan ringan sehingga lebih responsif terhadap request.
- Keunggulan dalam penanganan API untuk aplikasi web ataupun desktop.
- Adanya dukungan untuk bahasa pemograman lain seperti PostgreSQL dan JavaScript.

Kekurangan Menggunakan JSON

- Berbeda dengan XML yang memiliki sintaks yang menyerupai HTML, sintaks json distruktur dan diformat dengan gaya penulisan yang sulit dipahami.
- Bahasa JavaScript rentan terhadap hacking terutama pada website-website yang belum terpercaya.
- Tidak adanya penanganan error pada saat request.

Why JSON *is* Better Than XML ???

XML *is* much more difficult to parse than JSON. JSON *is* parsed into a ready-to-use JavaScript object.

For AJAX applications, JSON is faster and easier than XML:

Using XML

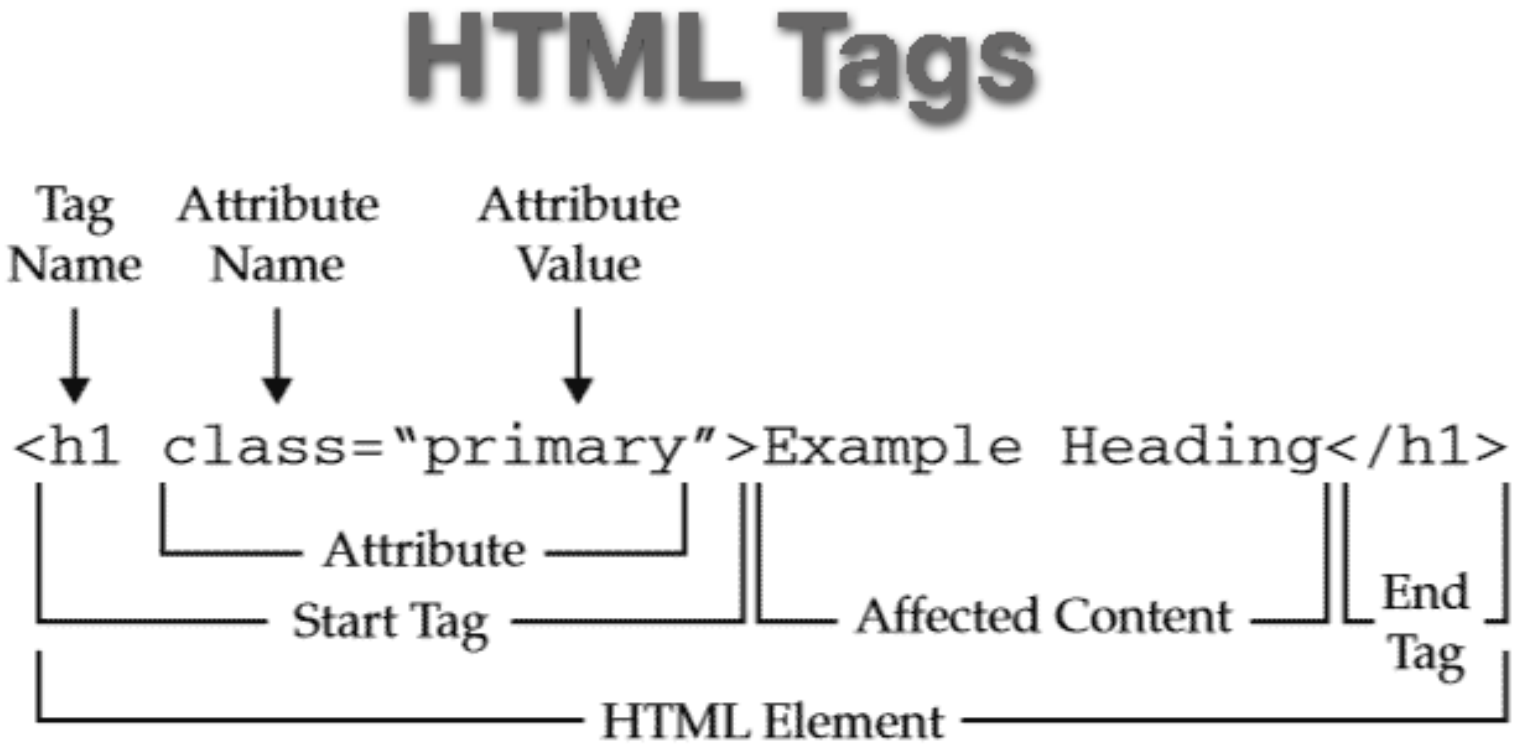
- Fetch an XML document
- Use the XML DOM to loop through the document
- Extract values and store in variables

Using JSON

- Fetch a JSON string
- JSON.Parse the JSON string

What *is* Tag Element *in* HTML?

- An HTML element is defined by a **START TAG**, some **CONTENT**, and an **END TAG**. An HTML element is defined by a starting tag. If the element contains other content, it ends with a closing tag.
- For example, ** ** is starting tag of a paragraph and ** ** is closing tag of the same paragraph but ** ** this is paragraph ** ** is a paragraph element.



What *is* Tag Element *in* HTML?

The **ID ATTRIBUTE** is used to point to a specific style declaration in a style sheet. It is also used by JavaScript to access and manipulate the element with the specific id.

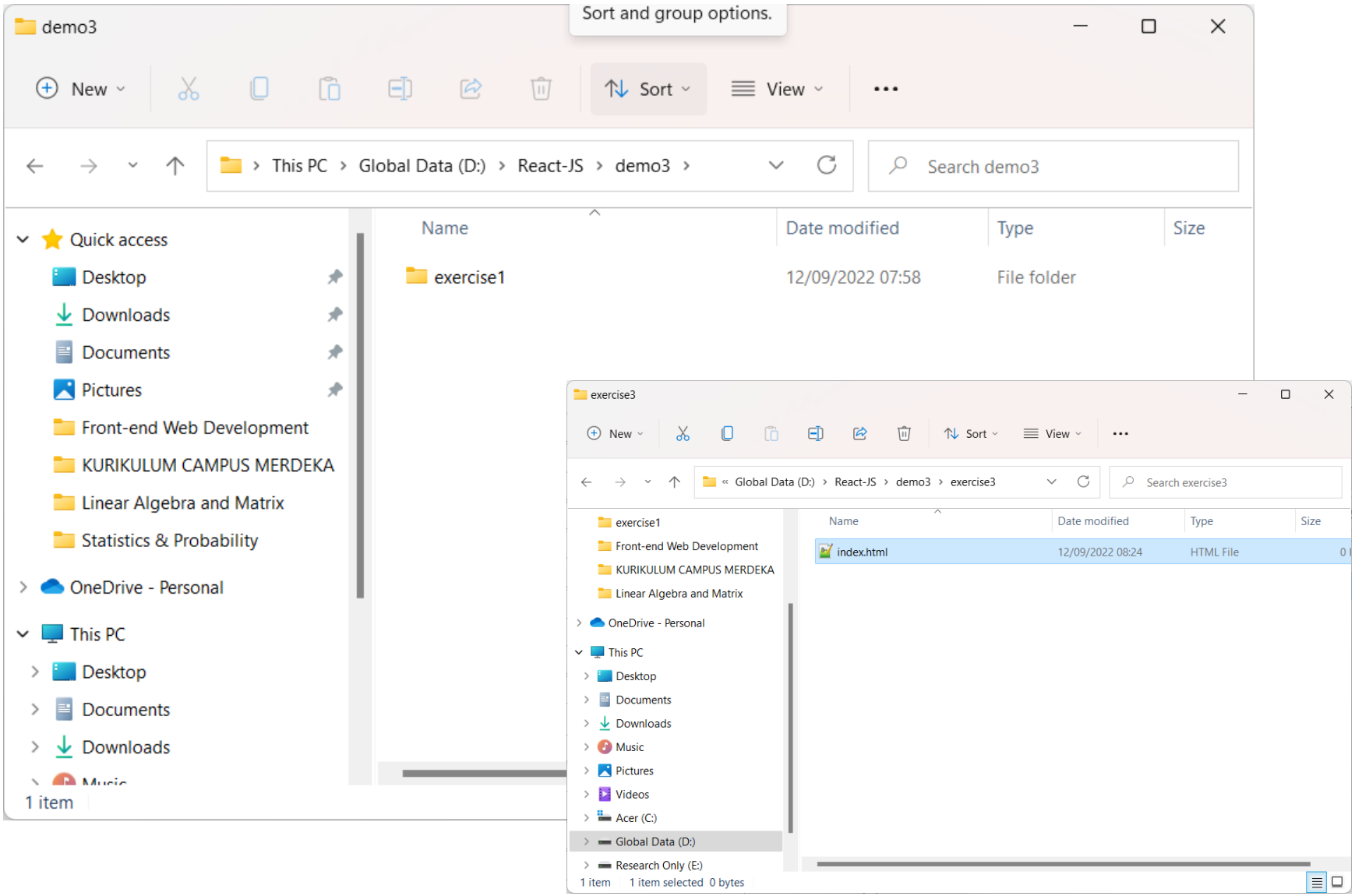
```
<!-- First define id attribute for HTML element -->
    <h1 id="firstElement">
        DOM Properties and Methods
    </h1>

// Declared & Assigned
        const h1Element = document.getElementById(
'firstElement');
        console.log(h1Element);
        console.log(h1Element.innerText);
```

Exercise *for* Students

Exercise #1

(*Parse data with JSON.parse() to become a JavaScript object*)

Create New Folder “demo3” > “exercise1”



Write HTML Code

D:\React-JS\demo3\exercise1\index.html - Notepad++

File Edit Search View Encoding Language Settings Tools Macro Run Plugins Window ?

index.html

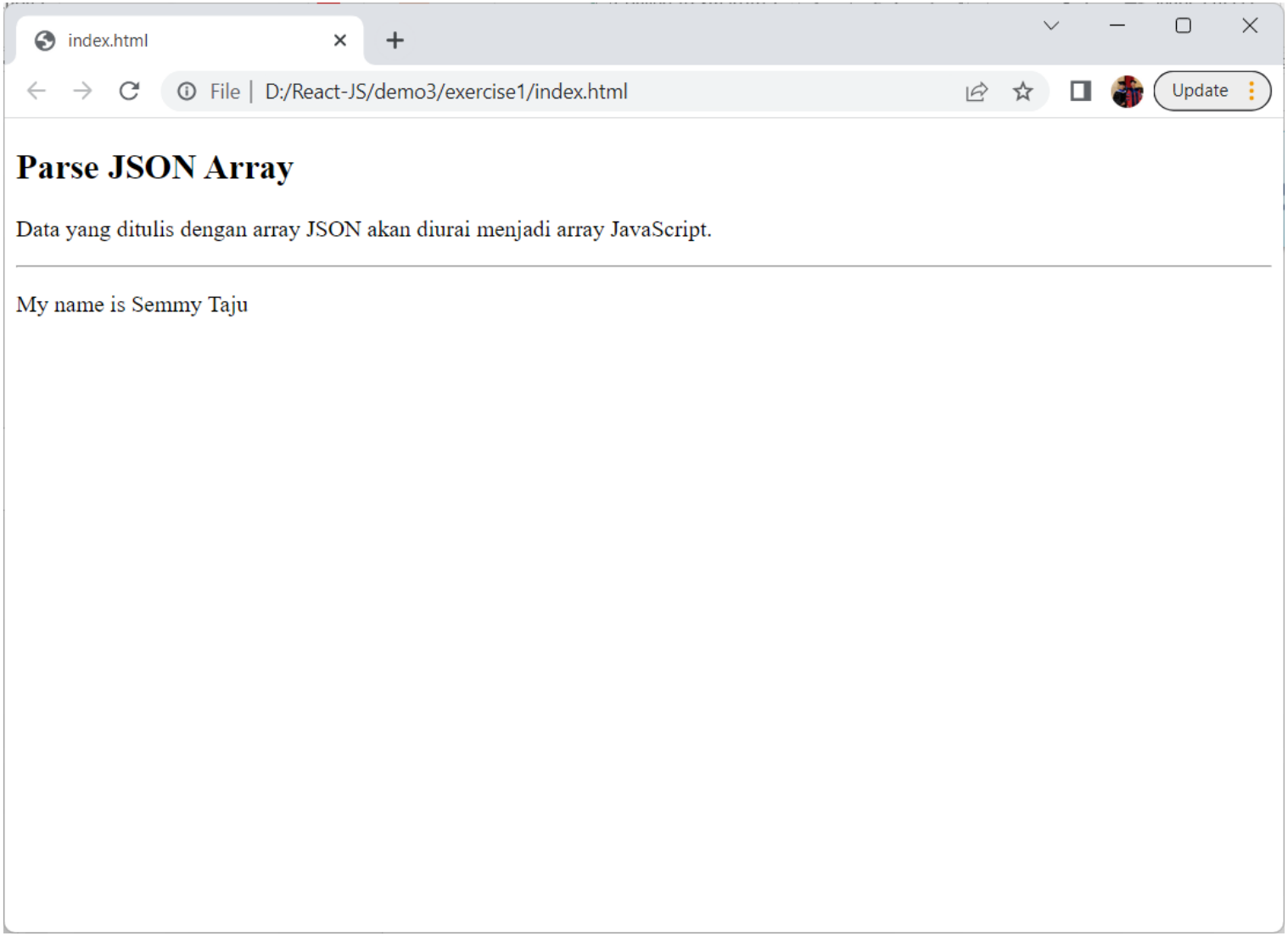
```html
<!DOCTYPE html>
<html>
<body>

<h2>Parse JSON Array</h2>
<p>Data yang ditulis dengan array JSON akan diurai menjadi array JavaScript.</p>
<hr>
<p id="demo3"></p>

<script>
    // array [redacted]
    const text = '[ "Semmy Taju", "Wandi Darea", "Recky Ronga", "Dilben Tulum"]';

    // object array javascript
    const myArr = JSON.parse(text);

    // print console
    console.log(myArr);

    // view to html
    document.getElementById("demo3").innerHTML = "My name is "+myArr[0];
</script>

</body>
</html>
```

Hyper Text Markup Language file | length : 511 lines : 26 | Ln : 1 Col : 1 Pos : 1 | Windows (CR LF) | UTF-8 | INS

Write JavaScript Code

```
10 <script>
11     // array [redacted]
12     const text = '[ "Semmy Taju", "Wandi Darea", "Recky Ronga", "Dilben Tulum"]';
13
14     // object array javascript
15     const myArr = JSON.parse(text);
16
17     // print console
18     console.log(myArr);
19
20     // view to html
21     document.getElementById("demo3").innerHTML = "My name is "+myArr[0];
22 </script>
```

Expected Output *in* Browser



Expected Output *in* Console

index.html

File | D:/React-JS/demo3/exercise1/index.html

Update

Parse JSON Array

Data yang ditulis dengan array JSON akan diurai menjadi array JavaScript.

My name is Semmy Taju

Elements Console Sources Network Performance

top Filter Default levels No Issues

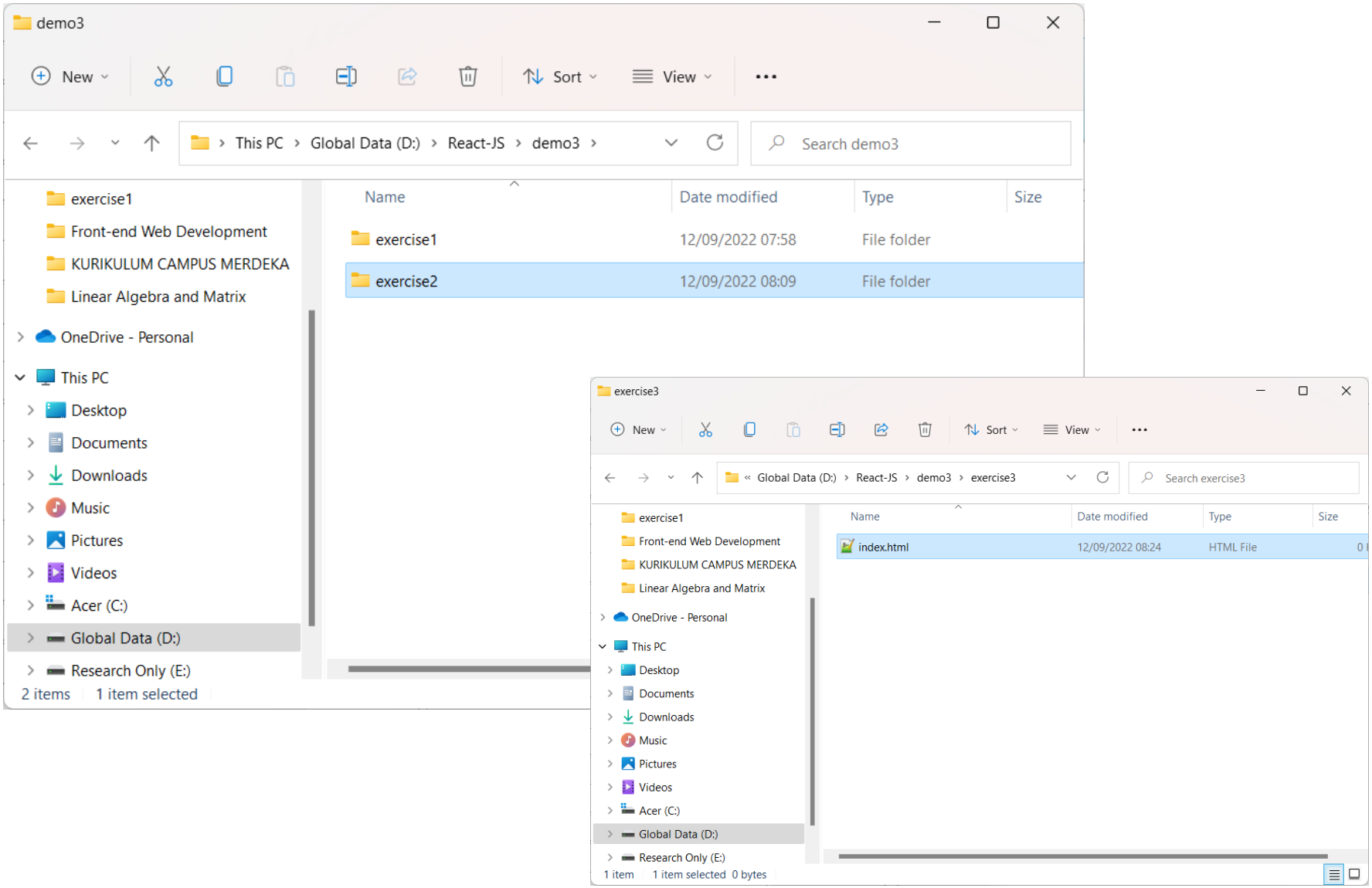
index.html:18

```
▼Array(4) ℹ
   0: "Semmy Taju"
   1: "Wandi Darea"
   2: "Recky Ronga"
   3: "Dilben Tulum"
   length: 4
  ▼[[Prototype]]: Array(0)
   ▶at: ƒ at()
   ▶concat: ƒ concat()
   ▶constructor: ƒ Array()
   ▶copyWithin: ƒ copyWithin()
   ▶entries: ƒ entries()
   ▶every: ƒ every()
   ▶fill: ƒ fill()
   ▶filter: ƒ filter()
   ▶find: ƒ find()
   ▶findIndex: ƒ findIndex()
   ▶findLast: ƒ findLast()
   ▶findLastIndex: ƒ findLastIndex()
   ▶flat: ƒ flat()
   ▶flatMap: ƒ flatMap()
   ▶forEach: ƒ forEach()
   ▶includes: ƒ includes()
   ▶indexOf: ƒ indexOf()
   ▶join: ƒ join()
   ▶keys: ƒ keys()
   ▶lastIndexOf: ƒ lastIndexOf()
    length: 0
   ▶map: ƒ map()
   ▶pop: ƒ pop()
   ▶push: ƒ push()
```

Exercise #2

(*Store and retrieve data from local storage*)

Create New Folder “exercise2”



Write HTML Code

D:\React-JS\demo3\exercise2\index.html - Notepad++

File Edit Search View Encoding Language Settings Tools Macro Run Plugins Window ?

index.html

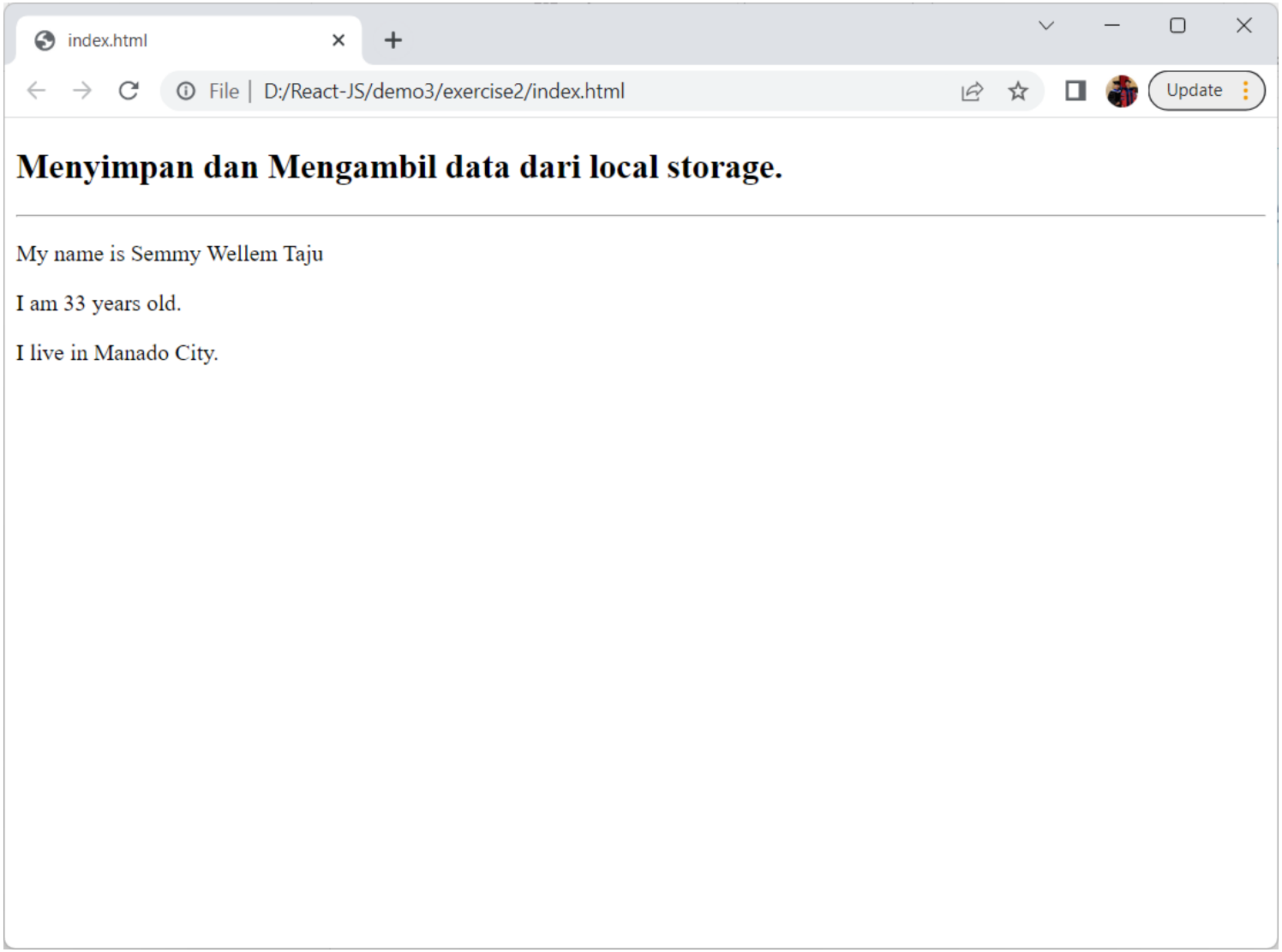
```html
<!DOCTYPE html>
<html>
<body>
<h2>Menyimpan dan Mengambil data dari local storage.</h2>
<hr>
<p id="txt-name"></p>
<p id="txt-age"></p>
<p id="txt-city"></p>

<script>
    // Store data:
    const myObj = { name: "Semmy Wellem Taju", age: 33, city: "Manado City" };
    const myJSON = JSON.stringify(myObj);
    localStorage.setItem("testJSON", myJSON);

    // print console
    console.log(myJSON);

    // Retrieve data:
    let text = localStorage.getItem("testJSON");
    let obj = JSON.parse(text);
    document.getElementById("txt-name").innerHTML = "My name is "+obj.name;
    document.getElementById("txt-age").innerHTML = "I am "+obj.age+" years old.";
    document.getElementById("txt-city").innerHTML = "I live in "+obj.city+".";
</script>
</body>
</html>
```

Hyper Text Markup Language file | length : 760 lines : 28 | Ln : 5 Col : 5 Pos : 97 | Windows (CR LF) | UTF-8 | INS

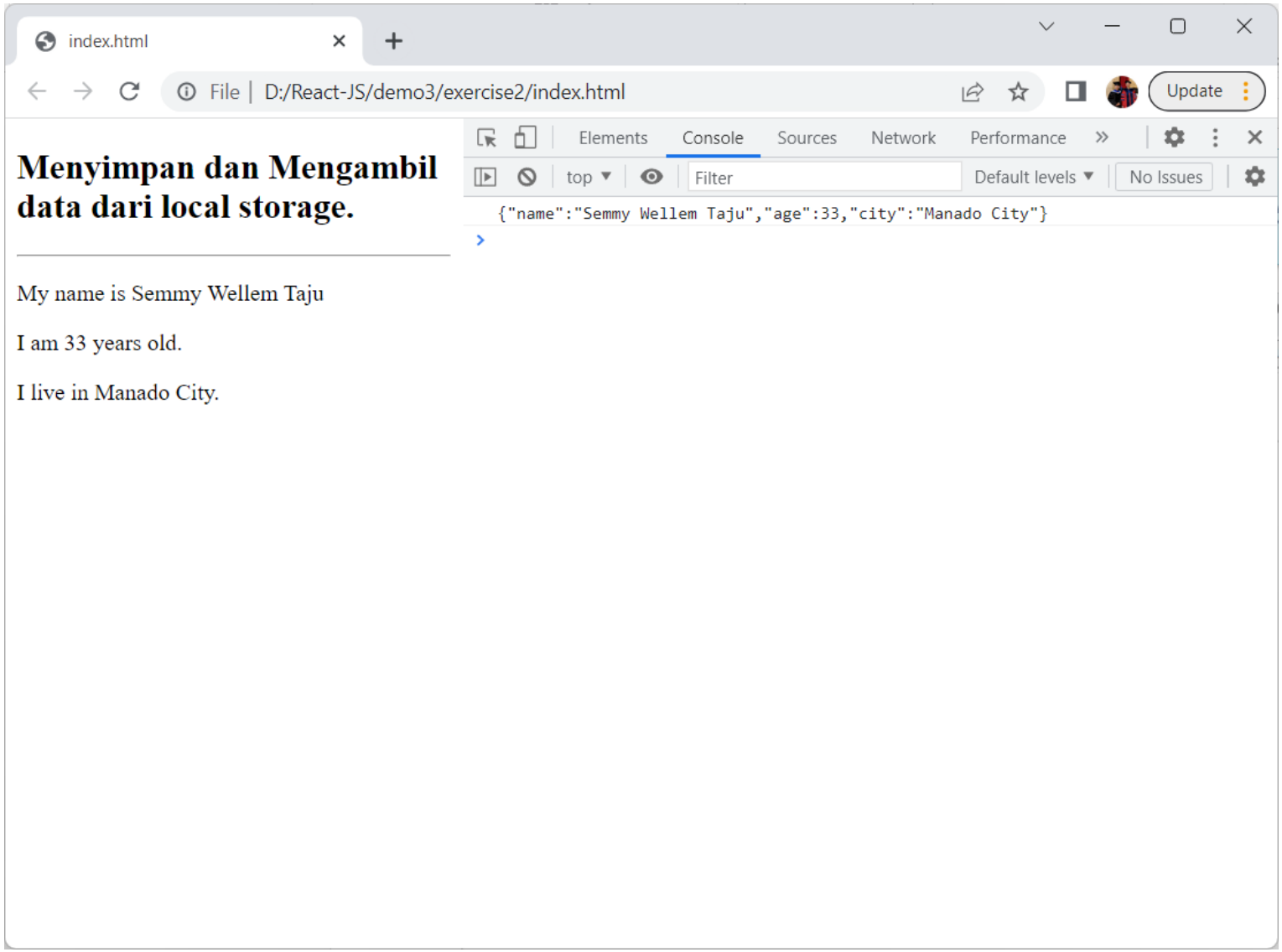
Write JavaScript Code

```
10 <script>
11     // Store data:
12     const myObj = { name: "Semmy Wellem Taju", age: 33, city: "Manado City" };
13     const myJSON = JSON.stringify(myObj);
14     localStorage.setItem("testJSON", myJSON);
15
16     // print console
17     console.log(myJSON);
18
19     // Retrieve data:
20     let text = localStorage.getItem("testJSON");
21     let obj = JSON.parse(text);
22     document.getElementById("txt-name").innerHTML = "My name is "+obj.name;
23     document.getElementById("txt-age").innerHTML = "I am "+obj.age+" years old.";
24     document.getElementById("txt-city").innerHTML = "I live in "+obj.city+".";
25 </script>
```

Expected Output *in* Browser



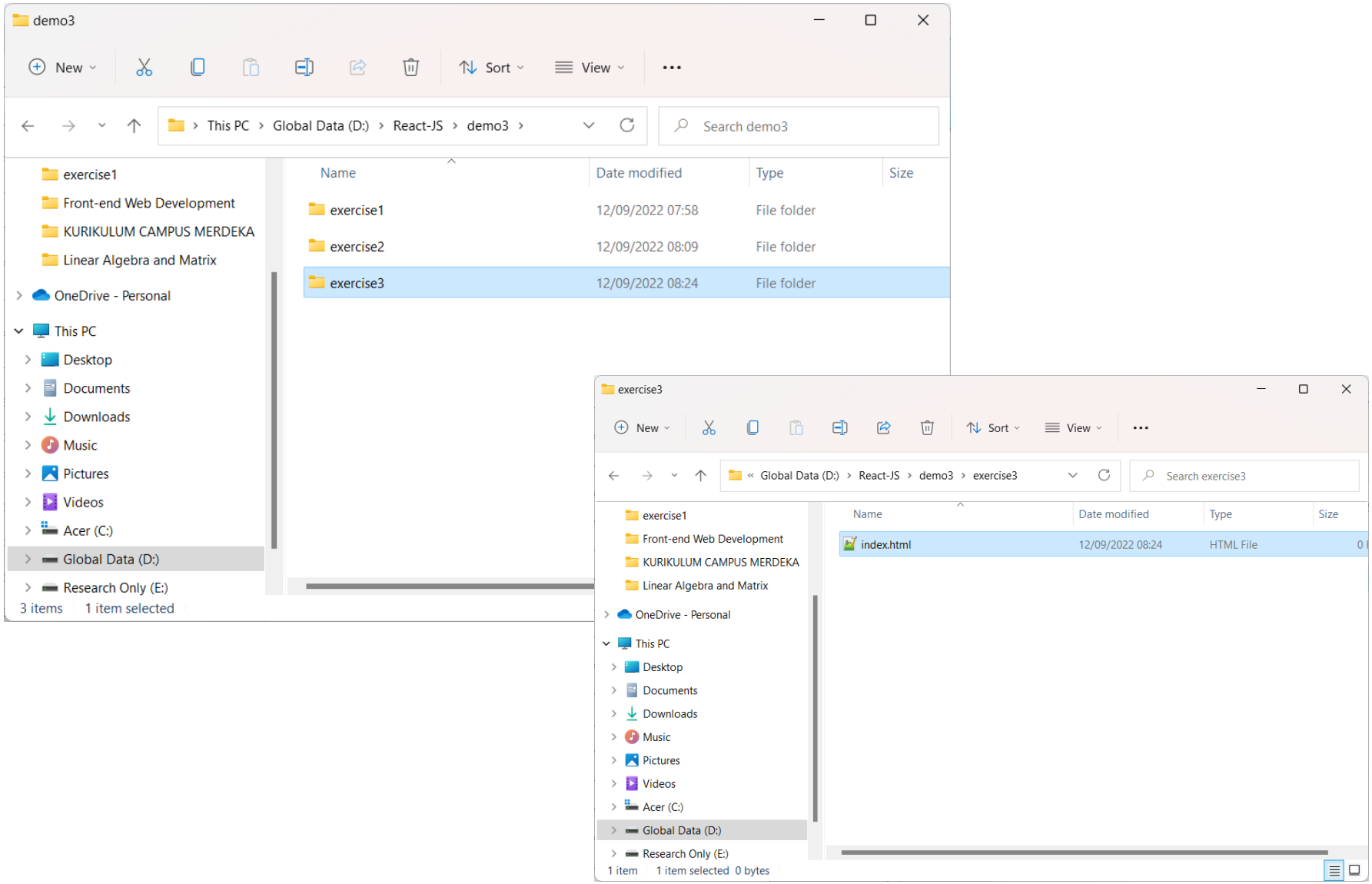
Expected Output *in* Console



Exercise #3

(*Display the Object in a Loop*)

Create New Folder “exercise3”



Write HTML Code

D:\React-JS\demo3\exercise3\index.html - Notepad++

File Edit Search View Encoding Language Settings Tools Macro Run Plugins Window ?

index.html

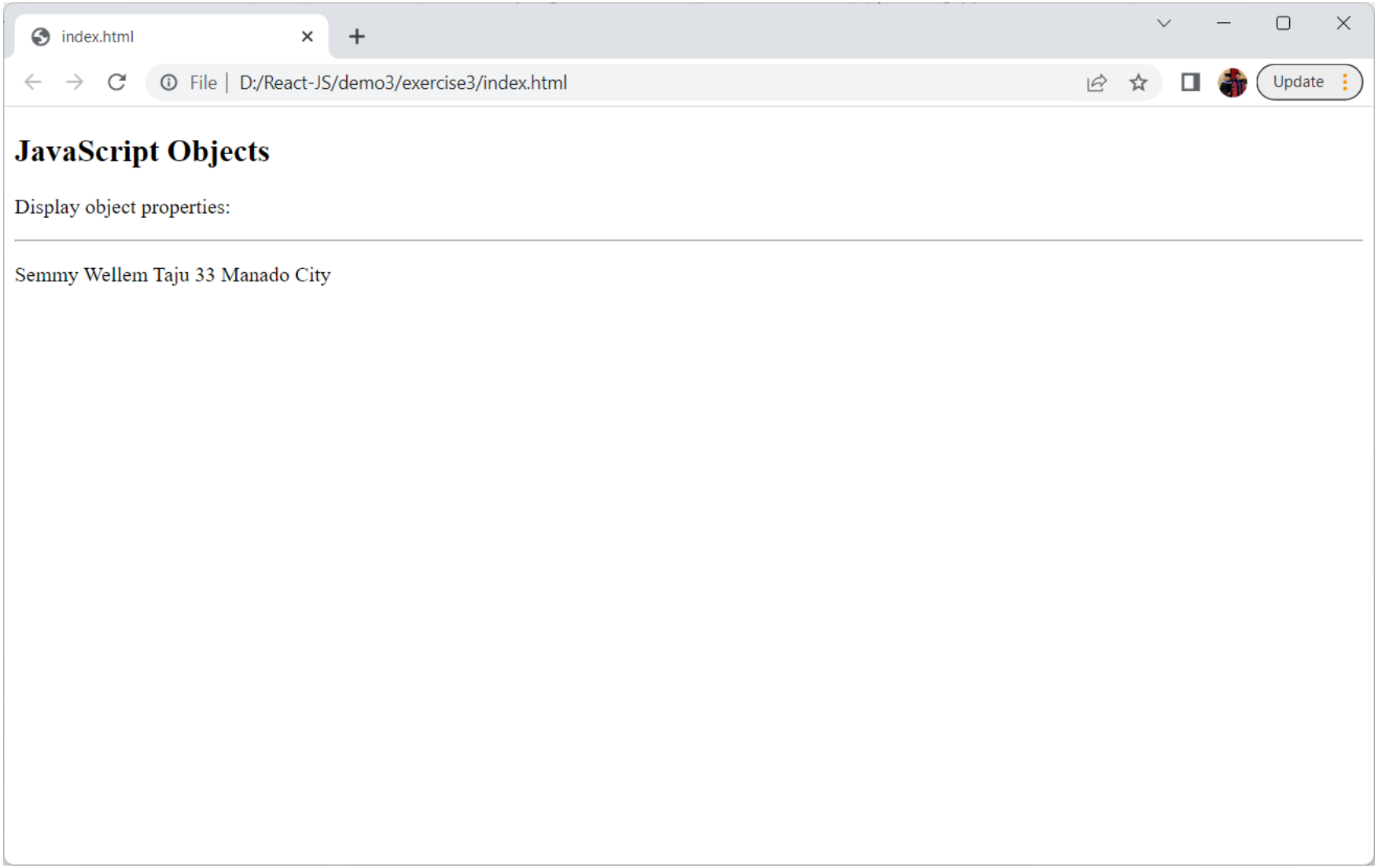
```html
<!DOCTYPE html>
<html>
<body>
<h2>JavaScript Objects</h2>
<p>Display object properties:</p>
<hr>
<p id="exercise3"></p>

<script>
    const person = {name: "Semmy Wellem Taju", age: 33, city: "Manado City"};

    let txt = "";
    for (let x in person) {
      txt += person[x] + " ";
    };

    document.getElementById("exercise3").innerHTML = txt;
</script>

</body>
</html>
```

Hyper Text Markup Language file | length : 379 lines : 21 | Ln : 1 Col : 1 Pos : 1 | Windows (CR LF) | UTF-8 | INS

Write JavaScript Code

```
9   <script>
10      const person = {name: "Semmy Wellem Taju", age: 33, city: "Manado City"};
11
12      let txt = "";
13      for (let x in person) {
14        txt += person[x] + " ";
15      };
16
17      document.getElementById("exercise3").innerHTML = txt;
18  </script>
```

Expected Output *in* Browser



END PRESENTATION

Thank you for your attention

Instructor: S – W – T