1. For the given JSON iterate over all for loops (for, for in, for of, forEach)
2. Create your own resume data in JSON format
3. Read about the difference between window, screen and document in javascript
4. Codekata practice
5. **For the given JSON iterate over all for loops (for, for in, for of, forEach)**

**1.For in:**

This allows you to iterate over all property keys of an object. In each iteration of the loop a key is assigned to the variable, this loop contains for all object property.

**We should not use for in to iterate over an array when the index order is important.**

for in can be used for both array, string and objects/user defined data types (cannot be used for the Jasonobject)

**NOTE:** for.in represents the key of the array, object, string etc.

For (key in object)

{

//body of for in

}

**Eg:**

var sal={

mithun:100000,

ashi:70000,

kruthi:40000

}

for (var x in sal)

{

var salary= 'Rs.'+ sal[x]+'/-'

console.log(`${x}= ${salary}`)

}

for (var x in sal)

{

var salary= 'Rs.'+ sal[x]+'/-'

console.log(`monthly salary of ${x} is rs ${sal[x]}/-`)

}

**OutPut:**

mithun= Rs.100000/-

ashi= Rs.70000/-

kruthi= Rs.40000/-

monthly salary of mithun is rs 100000/-

monthly salary of ashi is rs 70000/-

monthly salary of kruthi is rs 40000/-

**Eg:**

var n='JOHN'

for ( var x in n)

{

console.log(n[x])

}

var student=['JOHN',28,'is my age']

for (var i in student)

{

console.log(student[i])

}

Output:

J

O

H

N

JOHN

28

is my age

**2.for of:**

The for..o f loop in JavaScript allows you to iterate over iterable objects (arrays, sets, maps, strings etc) and cannot iterate the object/user defined data types.

For (element of iratble)

{

//body of for of loop

}

**Eg:**

var stu=['mithun','ashwini','kruthi']

for (var ele of stu)

{

console.log(ele)

}

//used for string

var string='mithun'

for (var i of string)

console.log(i)

// used in sets

const set=new Set([1,2,3,4])

for(let i of set)

{

console.log(i)

}

//used to iterate the map

let map=new Map();

map.set('name','mithun')

map.set('age','23')

map.set('yop','2021')

for (let [key, value] of map)

{

console.log(key+': '+value)

console.log(`${key} is ${value}`)

}

**Output:**

mithun

ashwini

kruthi

m

i

t

h

u

n

1

2

3

4

name: mithun

age: 23

yop: 2021

|  |  |
| --- | --- |
| for...of | for...in |
| The for...of loop is used to iterate through the values of an iterable. | The for...in loop is used to iterate through the  keys of an object. |
| The for...of loop cannot be used to iterate over an object. | You can use for...in to iterate over an iterable such arrays and strings but you should avoid using for...in for iterables. |

**3.forEach:**

this method calls a function and iterates over the element of an array/set/map.

Array.forEach(function(item,index,array))

{

//for each body

}

**or**

Array.forEach(myFunction)

Function myFunction(item,index,array)

{

//for each body

}

**Eg:**

var stu=['mithu','ashu','kruthi']

stu.forEach(nam)

function nam(item)

{

console.log(item)

}

//to add a word/updating an array ele

stu.forEach(names)

function names(item,index,arr)

{

arr[index]='Hello '+item

console.log(arr[index])

}

console.log(stu)

//for each with arrow function

var stu=['ashu','kruthi','mithun']

stu.forEach(itm =>

{

console.log(itm);

})

//foreach with sets

var set= new Set([4,2,5,1])

set.forEach(x)

function x(item)

{

console.log(item)

}

var set= new Set([10,44,21,4,3])

set.forEach(x)

function x(item,index,arr)

{

console.log(item)

}

//for each for map

let map=new Map();

map.set('mane','mithun')

map.set('age','100')

map.forEach(mydata)

function mydata(value, key)

{

console.log(key+': '+value)

}

**Output:**

mithu

ashu

kruthi

Hello mithu

Hello ashu

Hello kruthi

[ 'Hello mithu', 'Hello ashu', 'Hello kruthi' ]

ashu

kruthi

Mithun

4

2

5

1

10

44

21

4

3

name: mithun

age: 100

**2. Create your own resume data in JSON format**

var resume=[

{

"name":’JOHN’,

"phone no":"123456789",

'mial id':'JOHN@gmail.com'

},

{

'career object':'to sucseed in an envirnoment of growth and excellence which provides me the self satisfaction to achive organiztional and personal goal'

},

{

"10th school name":'acpuc',

'percent':'88%',

'yop':'2015'

},

{

'12th college name':'jskpuc',

"percent":'69%',

'yop':'2017'

},

{

'BE college name':'IIT',

"percent":'82%',

'yop':'2021'

},

{

'skills':'',

'i':’c++ programming’,

'ii':'c programming',

'iii':'full stack development'

}

]

for (var i=0;i<resume.length;i++)

{

console.log(resume[i])

}

OutPut:

{ name: 'JOHN',

'phone no': '123456789',

'mial id': 'JOHN@gmail.com' }

{ 'career object':

'to sucseed in an envirnoment of growth and excellence which provides me the self satisfaction to achive organiztional and personal goal' }

{ '10th school name': 'acpuc', percent: '88%', yop: '2015' }

{ '12th college name': 'jskpuc', percent: '69%', yop: '2017' }

{ 'BE college name': 'IIT', percent: '82%', yop: '2021' }

{ skills: '',

i: 'c++ programming',

ii: 'c programming',

iii: 'full stack development' }

1. **Read about the difference between window, screen and document in javascript**

**Window:**

The JavaScript window object sits at the top of the JavaScript Object hierarchy and represents the browser window. The window is the first thing that gets loaded into the browser .

The window object represents the current browsing context. It holds things like window.location, window.history, window.screen, window.status, or the window.document .

window.window

**Screen:**

Screen is a small information object about physical screen dimensions . It can be used to display screen width, height, colorDepth, pixelDepth etc. It is not mandatory to write window prefix with screen object. It can be written without window prefix.

Properties:

screen.width

screen.height

screen.availWidth

screen.availHeight

screen.colorDepth

screen.pixelDepth

**Document:**

The Document interface represents any web page loaded in the browser and serves as an entry point into the web page's content, which is the DOM tree. When an HTML document is loaded into a web browser , it becomes a document object. It is the root node of the HTML document. The document actually gets loaded inside the window object and has properties available to it like title, URL, cookie, etc. HTML documents, served with the "text/html" content type, also implement the HTMLDocument interface, whereas XML and SVG documents implement the XMLDocument interface.