Example 1

```
[
  {
    "name": {
      "first": "Laurence",
      "last": "Svekis"
    },
    "age": 40,
    "location": {
      "city": "Toronto",
      "country": "Canada"
    }
  },
  {
    "name": {
      "first": "Lisa",
      "last": "Suekis"
    },
    "age": 30,
    "location": {
      "city": "New York",
      "country": "USA"
    }
  },
  {
    "name": {
      "first": "Johyn",
      "last": "Sekis"
    },
    "age": 50,
    "location": {
      "city": "New York",
      "country": "USA"
    }
  }
]
<!DOCTYPE html>
<html>
<head>
    <title>JavaScript JSON</title>
    <style>
```

```
        @import url('https://fonts.googleapis.com/css2?family=Potta+One&display=swap');
        body{
            font-family: 'Potta One', cursive;
        }
        .output{
            font-size: 1.2em;
            width:80%;
            margin:auto;
            padding:10px;
        }

    </style>
</head>
<body>
    <h1>JSON</h1>
    <input type="text" class="val">
    <button class="btn">Click</button>
    <div class="output"></div>
    <script src="app1.js"></script>
</body>
</html>
const url = 'https://www.discoveryvip.com/shared/test1.json';
const localUrl = 'people.json';
const btn = document.querySelector('.btn');
const output = document.querySelector('.output');
const inputVal = document.querySelector('.val');
let attemptCounter = false;
inputVal.style.display = 'none';
btn.textContent = 'Load JSON data';
btn.addEventListener('click',(e)=>{
    getData(url);
})


function getData(urlPath){
    fetch(urlPath).then(rep => {
        return rep.json()
    }).then((json)=>{
        maker(json);
    }).catch(err=>{
        if(!attemptCounter){
            getData(localUrl);
        }
        attemptCounter = true;
```

```
        console.log(err);
    })
}

function maker(data){
    output.innerHTML = '<h1>JSON Data</h1>';
    data.forEach((el,index) => {
        console.log(index%2);
        const bg = index%2 == 0 ? '#eee' : '#fff';
        //${JSON.stringify(el)}
        const div = document.createElement('div');
        div.style.backgroundColor = bg;
        div.innerHTML += `<div>${el.name.first} ${el.name.last}</div>`;
        div.innerHTML += `<div>${el.location.city} ${el.location.country}</div>`;
        div.innerHTML += `<div>${el.age} </div>`;
        output.append(div);
    });
}
```

Example 2

```
<!DOCTYPE html>
<html>
<head>
    <title>JavaScript JSON</title>
    <style>
        @import url('https://fonts.googleapis.com/css2?family=Potta+One&display=swap');
        .box{
            padding:10px;
            margin:auto;
            width:80%;
            border: 1px solid #ddd;
            border-radius: 25px;
        }
        .box > div{
            padding:10px;
            font-size: 1.2em;
        }
    </style>
```

```
</head>
<body>
   <h1>JSON</h1>
   <input type="text" class="val">
   <button class="btn">Click</button>
   <div class="output"></div>
   <script src="app2.js"></script>
</body>
</html>
const url =
'https://en.wikipedia.org/w/api.php?action=query&format=json&list=search&origin=*&srsearch=;'
;
const btn = document.querySelector('.btn');
const output = document.querySelector('.output');
const inputVal = document.querySelector('.val');
let attemptCounter = false;
inputVal.value = 'hello';
btn.textContent = 'Load JSON data';
btn.addEventListener('click',(e)=>{
   let searchTerm = inputVal.value || 'JavaScript';
   let tempURL = url + searchTerm;
   console.log(tempURL);
   fetch(tempURL).then((rep)=>{ return rep.json()})
   .then((data)=>{
      console.log(data);
      output.innerHTML = '<div>Results for ' + searchTerm + '</div>';
      output.innerHTML += `Total Results : ${data.query.searchinfo.totalhits}<br>`;
      maker(data.query.search);
   })
})

function maker(data){
   console.log(data);

   data.forEach(el=> {
      console.log(el);
      const div = document.createElement('div');
      div.innerHTML += `<h3><a href="https://en.wikipedia.org/wiki?curid=${el.pageid}"
target="_blank">${el.title}</a></h3>`;
      div.innerHTML += `<div>Page ID ${el.pageid} | Size ${el.size} | WordCount ${el.wordcount}
</div>`;
      div.classList.add('box');
      div.innerHTML += el.snippet;
      output.append(div);
```

```
    });
}
```

Example 3

```
const url = 'http://api.wikimapia.org/?key=example&function=place.getnearest&format=json';
const btn = document.querySelector('.btn');
const output = document.querySelector('.output');
const inputVal = document.querySelector('.val');
const inputVal2 = document.createElement('input');
inputVal2.setAttribute('type','text');
inputVal2.value = '2.29451'; //lon
document.body.prepend(inputVal2);
const h1 = document.querySelector('h1');
document.body.prepend(h1);
inputVal.value = '48.858252'; //lat
btn.textContent = 'Search Map Lon Lat';
btn.addEventListener('click',(e)=>{
    let lon = inputVal2.value;
    let lat = inputVal.value;

    let tempURL = `${url}&lat=${lat}&lon=${lon}`;
    console.log(tempURL);
    fetch(tempURL).then((res)=>res.json())
    .then((data)=>{
        console.log(data);
        output.innerHTML = '';
        //JSON.stringify(data);
        maker(data.places);
    })
    .catch((err)=>{
        console.log(err);
    })
})

function maker(data){
    data.forEach(el => {
        console.log(el);
        const div = document.createElement('div');
        div.classList.add('box');
        div.innerHTML = `<div>Title ${el.title}<br>${el.urlhtml}</div>`;
        output.append(div);
    });
```

```
}
```

```
<!DOCTYPE html>
<html>
<head>
   <title>JavaScript JSON</title>
   <style>
      .box{
         border: 1px solid black;
         border-radius: 25px;
         padding:20px;
         margin:10px auto;
         width:80%;

      }
      .box:hover{
         opacity: 0.8;
      }
   </style>
</head>
<body>
   <h1>JSON</h1>
   <input type="text" class="val">
   <button class="btn">Click</button>
   <div class="output"></div>
   <script src="app3.js"></script>
</body>
</html>
```

Example 4

```
const btn = document.querySelector('.btn');
const urls = [{
      'url': 'https://www.discoveryvip.com/shared/books2.json',
      'arr': 'books',
      'title': 'Books List'
   },
   {
      'url': 'https://www.discoveryvip.com/shared/people.json',
      'arr': 'people',
      'title': 'Friends List'
```

```
    },
    {
        'url': 'https://www.discoveryvip.com/shared/coin.json',
        'arr': 'data',
        'title': 'BitCoin Currency'
    }
];
const h1 = document.querySelector('h1');
h1.innerHTML = '';
const output = document.querySelector('.output');
const inputVal = document.querySelector('.val');
inputVal.value = 'test';
btn.textContent = 'Click Me';
btn.addEventListener('click', (e) => {
    //console.log('ready');
    const temp = urls[2];
    //console.log(temp);
    myURL(urls[0]);
})


urls.forEach((ele) => {
    const btn1 = document.createElement('button');
    btn1.classList.add('btn');
    h1.append(btn1);
    btn1.textContent = ele.title;
    btn1.addEventListener('click', (e) => {
        myURL(ele);
    })
})



function myURL(myObj) {
    let url = myObj.url;
    fetch(url)
        .then(rep => rep.text())
        .then((data) => {
            //let val = data.replace(/\s/g,'');
            const json = JSON.parse(data);
            output.innerHTML = url + '<br>';
            maker(json[myObj.arr]);
            //console.log(json);
            //console.log(json);
```

```
        })
        .catch((err) => {
            console.log(err);
        })
}


function maker(arr) {
    console.log(arr.length);

    arr.forEach(el => {
        //console.log(el);
        const div = document.createElement('div');
        div.classList.add('box');
        output.append(div);
        const entries = Object.entries(el);
        console.log(entries);
        div.innerHTML = 'Properties : ' + entries.length;
        for (const obj of entries) {
            console.log(obj);
            div.innerHTML += `<br>${obj[0]} : ${obj[1]}`;
        }
    });
}



<!DOCTYPE html>
<html>
<head>
    <title>JavaScript JSON</title>
    <style>
        .box{
            padding:10px;
            border: 1px solid #ddd;
            width:80%;
            margin:10px auto;
        }
        .btn{
            padding : 10px;
            font-size: 1em;
            margin:5px;
            border-radius: 15px;
        }
```

```
    </style>
</head>
<body>
    <h1>JSON</h1>
    <input type="text" class="val">
    <button class="btn">Click</button>
    <div class="output"></div>
    <script src="app4.js"></script>
</body>
</html>
```


```
{
"data": [
{
"id": 1,
"name": "Bitcoin",
"symbol": "BTC",
"slug": "bitcoin",
"is_active": 1,
"first_historical_data": "2013-04-28T18:47:21.000Z",
"last_historical_data": "2019-04-05T20:44:01.000Z",
"platform": null
},
{
"id": 825,
"name": "Tether",
"symbol": "USDT",
"slug": "tether",
"is_active": 1,
"first_historical_data": "2015-02-25T13:34:26.000Z",
"last_historical_data": "2019-04-05T20:44:01.000Z",
"platform": {
"id": 83,
"name": "Omni",
"symbol": "OMNI",
"slug": "omni",
"token_address": "31"
}
},
{
"id": 1839,
"name": "Binance Coin",
"symbol": "BNB",
"slug": "binance-coin",
```

```
"is_active": 1,
"first_historical_data": "2017-07-25T04:30:05.000Z",
"last_historical_data": "2019-04-05T20:44:02.000Z",
"platform": {
"id": 1027,
"name": "Ethereum",
"symbol": "ETH",
"slug": "ethereum",
"token_address": "0xB8c77482e45F1F44dE1745F52C74426C631bDD52"
}
}
],
"status": {
"timestamp": "2018-06-02T22:51:28.209Z",
"error_code": 0,
"error_message": "",
"elapsed": 10,
"credit_count": 1
}
}
{
"books": [{
"title": "Learn to Code",
"author": "John Smith",
"isbn": "324-23243"
}, {
"title": "The Adventures JSON",
"author": "Jason Jones",
"isbn": "3324-2-444"
}, {
"title": "New Objects",
"author": "Jane Doe",
"isbn": "2343-234-2433"
}]
 }
{
  "people": [
    {
      "first": "Laurence"
      , "last": "Svekis"
    }
    , {
      "first": "Laurence"
      , "last": "Svekis"
```

```
    }
    , {
      "first": "Laurence"
      , "last": "Svekis"
    }
    , {
      "first": "Laurence"
      , "last": "Svekis"
    }
  ]
}
```

Example 5

```
<!DOCTYPE html>
<html>

<head>
    <title>JavaScript JSON</title>
    <style>
        @import url('https://fonts.googleapis.com/css2?family=Source+Sans+Pro&display=swap');

        body {
            font-family: 'Source Sans Pro', sans-serif;
        }

        * {
            box-sizing: border-box;
        }

        .box {
            padding: 10px;
            display: inline-block;
            width: 25%;
            text-align: center;
            font-size: 0.7em;

        }

        .box:hover {
```

```
            cursor: pointer;
        }

        .box>div {
            border: 1px solid #ddd;
            padding: 10px;
            overflow: hidden;
        }

        .box img {
            width: 100%;
        }
    </style>
</head>

<body>
    <h1>JSON</h1>
    <input type="text" class="val">
    <button class="btn">Click</button>
    <div class="output"></div>
    <script src="app5.js"></script>
</body>

</html>
const btn = document.querySelector('.btn');
const h1 = document.querySelector('h1');
h1.style.width = '80%';
h1.style.margin = 'auto';
h1.style.textAlign = 'center';
h1.style.border = '5px solid #ddd';
const output = document.querySelector('.output');
const inputVal = document.querySelector('.val');
//.https://cors-anywhere.herokuapp.com/
const url = 'https://randomuser.me/api/';
inputVal.value = '10';
inputVal.setAttribute('type', 'number');
btn.textContent = 'Click Me';
btn.addEventListener('click', (e) => {
    console.log('ready');
    let val = `?results=${inputVal.value}`;
    adder(url + val);
})

function adder(url) {
```

```
    console.log(url);
    fetch(url)
        .then((rep) => {
            return rep.json()
        })
        .then((data => {
            console.log(data);
            output.innerHTML = `<h3>Seed : ${data.info.seed}<br>Results :
${data.info.results}</h3>`;
            maker(data.results);
        }))
}


function maker(data) {
    console.log(data);
    data.forEach(el => {
        console.log(el);
        const loc = el.location;
        const div = eleMaker('div', output, '');
        div.classList.add('box');
        const temp = `${el.name.title} ${el.name.first} ${el.name.last}<br>${el.email}<br>Age :
${el.dob.age}`;
        const temp1 = `<img src="${el.picture.large}">`;
        const temp2 = `${loc.city} ${loc.state} ${loc.country}`;
        div.addEventListener('click', (e) => {
            h1.innerHTML = temp + '<div>' + temp1 + '</div>';
            window.scrollTo({
                top: 0
            });
        })


        eleMaker('div', div, temp);
        eleMaker('div', div, temp1);
        eleMaker('div', div, temp2);
    });
}

function eleMaker(eleTag, parent, contents) {
    const elem = document.createElement(eleTag);
    parent.append(elem);
    elem.innerHTML = contents;
    return elem;
```

```
}
```

Example 6

```
const btn = document.querySelector('.btn');
const h1 = document.querySelector('h1');
const output = document.querySelector('.output');
const inputVal = document.querySelector('.val');
const url1 = 'https://samples.openweathermap.org/';
const url2 = 'https://cors-anywhere.herokuapp.com/';
btn.textContent = 'Click Me';
btn.addEventListener('click', (e) => {
    //console.log('ready');
    getValues(url2 + url1);
})

function getValues(url){
    fetch(url)
    .then(rep => rep.json())
    .then(data => {
        maker(data.products.forecast_5days);
    })
}

function maker(data){
    //console.log(data.docs);
    data.samples.forEach((el)=>{
        getMore(url2 + el);
    })
}

function getMore(url){
    fetch(url)
    .then(rep => rep.json())
    .then(data => {
        adder(data);
    })
}

function adder(data){
    console.log(data.list);
```

```
    const city = data.city;
    const div = document.createElement('div');
    output.append(div);
    div.innerHTML = `${city.name} ${city.country} <br>
    ${city.coord.lat} ${city.coord.lon}`;
}
```

```
<!DOCTYPE html>
<html>

<head>
    <title>JavaScript JSON</title>
    <style>
    </style>
</head>

<body>
    <h1>JSON</h1>
    <input type="text" class="val">
    <button class="btn">Click</button>
    <div class="output"></div>
    <script src="app6.js"></script>
</body>

</html>
```

This API enables cross-origin requests to anywhere.

Usage:

/ Shows help
/iscorsneeded This is the only resource on this host which is served without CORS headers.
/<url> Create a request to <url>, and includes CORS headers in the response.

If the protocol is omitted, it defaults to http (https if port 443 is specified).

Cookies are disabled and stripped from requests.

Redirects are automatically followed. For debugging purposes, each followed redirect results in the addition of a X-CORS-Redirect-n header, where n starts at 1. These headers are not

accessible by the XMLHttpRequest API.
After 5 redirects, redirects are not followed any more. The redirect response is sent back to the browser, which can choose to follow the redirect (handled automatically by the browser).

The requested URL is available in the X-Request-URL response header.
The final URL, after following all redirects, is available in the X-Final-URL response header.

To prevent the use of the proxy for casual browsing, the API requires either the Origin or the X-Requested-With header to be set. To avoid unnecessary preflight (OPTIONS) requests, it's recommended to not manually set these headers in your code.

Demo : https://robwu.nl/cors-anywhere.html
Source code : https://github.com/Rob--W/cors-anywhere/
Documentation : https://github.com/Rob--W/cors-anywhere/#documentation

https://developer.mozilla.org/en-US/docs/Web/HTTP/CORS

Cross-Origin Resource Sharing (CORS) is an HTTP-header based mechanism that allows a server to indicate any other origins (domain, scheme, or port) than its own from which a browser should permit loading of resources. CORS also relies on a mechanism by which browsers make a “preflight” request to the server hosting the cross-origin resource, in order to check that the server will permit the actual request. In that preflight, the browser sends headers that indicate the HTTP method and headers that will be used in the actual request.

An example of a cross-origin request: the front-end JavaScript code served from https://domain-a.com uses XMLHttpRequest to make a request for https://domain-b.com/data.json.

For security reasons, browsers restrict cross-origin HTTP requests initiated from scripts. For example, XMLHttpRequest and the Fetch API follow the same-origin policy. This means that a web application using those APIs can only request resources from the same origin the application was loaded from unless the response from other origins includes the right CORS headers.

Example 7

```
<!DOCTYPE html>
<html>

<head>
```

```
    <title>JavaScript JSON</title>
    <style>
        .btns{
            padding:12px;
            margin:5px;
            cursor: pointer;
            font-size: 0.9em;
            border: 1px solid #ddd;
            border-radius: 25px;
            text-transform: capitalize;
        }
        .btns:hover{
            opacity: 0.8;
        }
        input, .btn{
            font-size: 1.5em;
            display: block;
            margin:auto;
            text-align: center;
            width:80%;
        }
        .output{
            padding:20px;
            margin:auto;
            width: 70%;
            font-size: 0.9em;
            border:1px solid #ddd;
        }

    </style>
</head>

<body>
    <h1>JSON</h1>
    <input type="text" class="val">
    <button class="btn">Click</button>
    <div class="output"></div>
    <script src="app7.js"></script>
</body>

</html>
const btn = document.querySelector('.btn');
const h1 = document.querySelector('h1');
const output = document.querySelector('.output');
```

```
const inputVal = document.querySelector('.val');
const url1 = 'https://api.chucknorris.io/jokes/';
btn.textContent = 'Search';

buildCats();

btn.addEventListener('click', (e) => {
    console.log('ready');
    const val1 = inputVal.value || 'test';
    const tempURL = url1 + 'search?query='+val1;
    getJokes(tempURL,val1);
})

function buildCats(){
    const urlTemp = url1 + 'categories';
    console.log(urlTemp);
    fetch(urlTemp).then(rep => rep.json())
    .then((data)=>{
        console.log(data);
        h1.innerHTML = '';
        data.forEach((cat)=>{
            const btnTemp = document.createElement('button');
            btnTemp.classList.add('btns');
            btnTemp.textContent = cat;
            h1.append(btnTemp);
            btnTemp.addEventListener('click',(e)=>{
                //https://api.chucknorris.io/jokes/random?category={category}
                console.log(cat);
                const tempURL = url1 + 'random?category=' + cat;
                fetch(tempURL).then(rep=>rep.json())
                .then((json)=>{
                    output.innerHTML = 'Category : ' + cat + '<hr>';
                    addJoke(json.value);
                })
            })
        })
    })
}


function getJokes(url,searchTerm){
    fetch(url)
    .then(rep => rep.json())
    .then((data)=>{
```

```
        output.innerHTML = `${searchTerm} found ${data.total}`;
        console.log(data);
        data.result.forEach((joke) => {
            console.log(joke);
            addJoke(joke.value);
        });
    })

}


function getJoke(url){
    fetch(url)
    .then(rep => rep.json())
    .then((data)=>{
        output.innerHTML = '';
        addJoke(data.value);
    })
}

function addJoke(val){
    output.innerHTML += val + '<br>';
}
```

Example 8

The for...in statement iterates over all enumerable properties of an object that are keyed by strings (ignoring ones keyed by Symbols), including inherited enumerable properties.

```
const object = { a: 1, b: 2, c: 3 };

for (const property in object) {
  console.log(`${property}: ${object[property]}`);
}

// expected output:
// "a: 1"
// "b: 2"
// "c: 3"
```

https://developer.mozilla.org/en-US/docs/Web/JavaScript/Reference/Statements/for...in

https://swapi.dev/

```
<!DOCTYPE html>
<html>

<head>
   <title>JavaScript JSON</title>
   <style>
      * {
         box-sizing: border-box;
      }

      .btnz {
         padding: 10px;
         border: 1px solid black;
         border-radius: 15px;
         font-size: 0.8em;
         margin: 5px;
         text-transform: uppercase;
         cursor: pointer;
      }

      .bigText {
         font-size: 1.5em;
         padding: 5px;
         font-weight: bold;
         text-transform: capitalize;
      }

      .btnz:hover {
         opacity: 0.8;
      }

      .output {
         text-align: center;
      }

      .myTitle {
         text-transform: capitalize;
         font-size: 3em;

      }
```

```
        .box {
            padding: 10px;
            text-align: center;
            width: 50%;
            margin: 5px auto;
            border: 1px solid #ddd;
            cursor: pointer;
            background-color: black;
            color: white;
        }

        .box:hover {
            background-color: red;
        }

        .pages {
            width: 80%;
            text-align: center;
            margin: auto;

        }

        .pages button {
            font-size: 1.2em;
            margin: 5px;
            padding: 10px;
            border-radius: 20px;

        }
    </style>
</head>

<body>
    <h1>JSON</h1>
    <input type="text" class="val">
    <button class="btn">Click</button>
    <div class="output"></div>
    <script src="app8.js"></script>
</body>

</html>
const btn1 = document.querySelector('.btn');
const h1 = document.querySelector('h1');
const output = document.querySelector('.output');
```

```
const inputVal = document.querySelector('.val');
inputVal.style.display = 'none';
btn1.style.display = 'none';
const mainURL = 'http://swapi.dev/api/';
let endPoint = '';
let endTitle = '';
window.addEventListener('DOMContentLoaded', (e) => {
  //console.log('DOM ready');
  fetch(mainURL).then((rep) => {
    return rep.json();
  }).then((json) => {
    //console.log(json);
    h1.innerHTML = '';
    for (const prop in json) {
      //console.log(`${prop} : ${json[prop]}`);
      const btn = document.createElement('button');
      btn.classList.add('btnz');
      btn.textContent = prop;
      endTitle = prop;
      h1.append(btn);
      btn.urlz = json[prop];
      btn.addEventListener('click', getData);
    }
  })
})

btn1.addEventListener('click', (e) => {
  console.log('ready');
})


function getData(e) {
  //console.log(e.target);
  const el = e.target;
  getJSON(el.urlz);
}


function getJSON(url) {
  endPoint = url;
  fetch(url)
    .then(rep => rep.json())
    .then(data => {
      console.log(data);
```

```
    buildPage(data);
  })
}


function buildPage(data) {
  console.log(data);

  output.innerHTML = `<h1 class="myTitle">${endTitle}</h1><small>${endPoint}</small>`;
  data.results.forEach(el => {
    const div = document.createElement('div');
    div.textContent = el.name || el.title;
    div.classList.add('box');
    div.urlz = el.url;
    div.addEventListener('click', showItem);
    output.append(div);
    console.log(el.name);
  });
  const pages = document.createElement('div');
  pages.classList.add('pages');
  output.append(pages);
  if (data.previous) {
    const btn2 = document.createElement('button');
    btn2.textContent = 'Previous';
    pages.append(btn2);
    btn2.urlz = data.previous;
    btn2.addEventListener('click', (e) => {
      console.log(data.previous);
      getJSON(data.previous);
    });
  }

  const total = Math.ceil(data.count / 10);
  for (let i = 0; i < total; i++) {
    const btn2 = document.createElement('button');
    btn2.textContent = i + 1;
    pages.append(btn2);

    let cleanURL = endPoint.split('?');
    console.log(cleanURL);
    let tempURL = cleanURL[0] + '?page=' + (i + 1);
    btn2.urlz = tempURL;
    btn2.addEventListener('click', (e) => {
      console.log(tempURL);
```

```
      getJSON(tempURL);
    });
  }
  if (data.next) {
    const btn2 = document.createElement('button');
    btn2.textContent = 'Next';
    pages.append(btn2);
    btn2.urlz = data.next;
    btn2.addEventListener('click', (e) => {
      console.log(data.next);
      getJSON(data.next);
    });
  }
}

function showItem(e) {
  const el = e.target;
  console.log(el.urlz);
  output.innerHTML = '';
  fetch(el.urlz).then(rep => rep.json())
    .then((data) => {
      //console.log(data);
      for (const prop in data) {
        console.log(`${prop}: ${data[prop]}`);
        let html = (typeof (data[prop]) == 'string') ? data[prop] : JSON.stringify(data[prop]);
        let propTemp = prop.replace('_', ' ');
        output.innerHTML += `<div><span class="bigText">${propTemp}</span>: ${html}</div>`;
      }

    })
    .catch((err) => {
      console.log(err);
      output.innerHTML = 'ERROR';
    })
}
```

Example 9

https://opentdb.com/api_config.php

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

```
<head>
    <title>JavaScript JSON</title>
    <style>
        @import url('https://fonts.googleapis.com/css2?family=Work+Sans&display=swap');
        *{box-sizing: border-box;}
        body{
            font-family: 'Work Sans', sans-serif;
        }
        .output , .box{
            width:80%;
            margin:auto;
            border:1px solid #ccc;
            background-color: #ddd;
            padding:10px;
        }
        .output > div{
            font-size: 2em;
            margin:auto;

        }
        .box{
            text-align: center;
        }
        .box input  , .box select , .box button{
            font-size: 1.2em;
            border: 1px solid #222;
            line-height: 28px;
            text-transform: capitalize;
            min-height: 35px;
            margin:5px;
        }
        .box input{
            font-size: 1em;
            width:80px;
        }
        h1 {
            text-align: center;
        }
        .output small{
            text-align: center;
            font-size: 0.6em;
        }
        .output button{
```

```
            text-align: center;
            padding:10px;
            cursor: pointer;
            border: 1px solid black;
            color:white;
            background-color: #333;
            margin:5px;
            font-size:0.5em;
            border-radius: 10px;

        }

    </style>
</head>

<body>
    <h1>JSON</h1>
    <input type="text" class="val">
    <button class="btn">Click</button>
    <div class="output"></div>
    <script src="app9.js"></script>
</body>

</html>
const btn1 = document.querySelector('.btn');
const h1 = document.querySelector('h1');
h1.textContent = "Trivia DataBase Game";
const output = document.querySelector('.output');
const output1 = genElement(document.body, 'div', 'Please Make you Selection<br>#
Questions');
output1.classList.add('box');
const inputVal = document.querySelector('.val');
output1.append(inputVal);
const sel1 = genElement(output1, 'select', '');
const sel2 = genElement(output1, 'select', '');
output1.append(btn1);
const baseURL = 'https://opentdb.com/api.php?';
const game = {
    que: [],
    question: 0,
    eles: [],
    score:0
};
const cats = [{
```

```
    "title": "General",
    "num": 9
}, {
    "title": "Sports",
    "num": 21
}, {
    "title": "Animals",
    "num": 27
}];
const dif = ['easy', 'medium', 'hard'];

//https://opentdb.com/api.php?amount=10&category=9
window.addEventListener('DOMContentLoaded', (e) => {
    console.log('DOM ready');
    genSelections();
    //testinsert();
    btn1.textContent = 'Start Game';
    inputVal.setAttribute('type', 'number');
    inputVal.value = 10;
})

function genSelections() {
    cats.forEach((cat) => {
        console.log(cat);
        const optEle = genElement(sel1, 'option', cat.title);
        optEle.value = cat.num;
    })
    dif.forEach((d) => {
        console.log(d);
        const optEle = genElement(sel2, 'option', d);
        optEle.value = d;
    })
}

btn1.addEventListener('click', (e) => {
    output1.style.display = 'none';
    h1.textContent = inputVal.value + ' question(s) selected';
    let tempURL =
`${baseURL}amount=${inputVal.value}&difficulty=${sel2.value}&category=${sel1.value}`;
    console.log(tempURL);
    popPage(tempURL);
})

function testinsert() {
```

```
    for (let x = 0; x < 500; x++) {
        let tempArr = [0, 0, 0];
        let ranIndex = Math.floor((Math.random() * (tempArr.length + 1)));
        tempArr.splice(ranIndex, 0, 1);
        output.innerHTML += JSON.stringify(tempArr) + ' : ' + ranIndex + '<br>';
    }


}

function popPage(url) {
    fetch(url)
        .then(res => res.json())
        .then(data => {
            game.que = data.results;
            outputPage();
        })
}


function outputPage() {

    console.log(game.question);
    if (game.question >= game.que.length) {
        output.innerHTML = `<div>Your Score was ${game.score} out of
${game.que.length}</div>`;
        game.score = 0;
        output1.style.display = 'block';
        game.question = 0;
    } else {
        output.innerHTML = '';
        let question = game.que[game.question];
        game.question++; //move to next question
        h1.textContent = `Question ${game.question} of ${game.que.length} - SCORE :
${game.score}`;
        console.log(question);
        let answers = question.incorrect_answers;
        let ranIndex = Math.floor(Math.random() * (answers.length + 1));
        console.log(ranIndex);
        answers.splice(ranIndex, 0, question.correct_answer);
        //answers.push(question.correct_answer);
        console.log(answers);
        const mainDiv = genElement(output, 'div', '');
        const que1 = genElement(mainDiv, 'div', question.question);
```

```
        game.eles.length = 0;
        const optsDiv = genElement(output, 'div', '');
        answers.forEach(opt => {
            const opt1 = genElement(optsDiv, 'button', opt);
            game.eles.push(opt1);
            if (opt == question.correct_answer) {
                opt1.bgC = 'green';

            } else {
                opt1.bgC = 'red';
            }
            opt1.addEventListener('click', (e) => {
                game.eles.forEach((btnv) => {
                    btnv.disabled = true;
                    btnv.style.backgroundColor = btnv.bgC;
                })
                const message = genElement(optsDiv, 'div', `You got it Incorrect!
<small>${question.correct_answer} was correct.</small><br>`);
                if (opt == question.correct_answer) {
                    console.log('correct');
                    message.innerHTML = `You Got it Correct! <small>${opt} was
correct.</small><br>`;
                    game.score++;

                    opt1.style.backgroundColor = 'green';
                } else {
                    console.log('wrong');
                    opt1.style.backgroundColor = 'red';
                }
                h1.textContent = `Question ${game.question} of ${game.que.length} - SCORE :
${game.score}`;
                nextQue(message);
                console.log(game);
            })
        });
        /*     game.que.forEach(el => {
                console.log(el);
            }); */
    }
}

function nextQue(parent) {
    const btn2 = genElement(parent, 'button', 'Next Question');
    btn2.addEventListener('click', outputPage);
```

```
}

function genElement(parent, eleType, html) {
    const temp = document.createElement(eleType);
    temp.innerHTML = html;
    parent.append(temp);
    return temp;
}
```

Example 10

https://api.stackexchange.com/docs

```
<!DOCTYPE html>
<html>

<head>
    <title>JavaScript JSON</title>
    <style>
        @import url('https://fonts.googleapis.com/css2?family=Oswald&display=swap');

        * {
            box-sizing: border-box;
        }

        body {
            font-family: 'Oswald', sans-serif;
        }

        .tag {
            padding: 5px;
            margin: 5px;
            font-style: italic;
            border: 1px solid #ddd;
            display: inline-block;
        }

        .tag::before {
            content: "-";
        }
```

```
.topTitle {
    font-size: 1.2em;
    font-weight: bold;
    margin-bottom: 15px;
    border-bottom: 3px solid black;
    padding-bottom: 10px;
}

.box {
    padding: 15px;
    border: 1px dotted #ddd;
    border-radius: 25px;
    cursor: pointer;
    margin-bottom: 5px;
}

.box:hover {
    background-color: #eee;
}

.ans {
    text-align: center;
    font-size: 0.8em;
}

.info,
.output {
    width: 80%;
    border: 1px solid #ddd;
    font-size: 1em;
    margin: auto;
}

.output1 {
    width: 80%;
    border: 1px solid #ddd;
    font-size: 1em;
    margin: auto;
    text-align: center;
}

input,
.btn {
    display: block;
```

```
            width: 100%;
            font-size: 1.2em;
            height: 45px;
            text-align: center;
        }
    </style>
</head>

<body>
    <h1>JSON</h1>
    <input type="text" class="val">
    <button class="btn">Click</button>
    <div class="output"></div>
    <script src="app10.js"></script>
</body>

</html>
const btn1 = document.querySelector('.btn');
const h1 = document.querySelector('h1');
h1.textContent = 'Stackexchange API tester';
const output1 = makeNode(document.body, 'div', '');
output1.classList.add('output1');

const output = document.querySelector('.output');
const inputVal = document.querySelector('.val');
inputVal.value = 'javascript';
output1.append(h1);
output1.append(inputVal);
output1.append(btn1);
output1.append(output);

const baseURL = 'https://api.stackexchange.com/';
window.addEventListener('DOMContentLoaded', (e) => {
    //console.log('DOM ready');
    pageLoad();
})

btn1.addEventListener('click', (e) => {
    //console.log('Click ready');
    const searchTerm = inputVal.value;
    const url = baseURL +
`2.2/search?order=desc&sort=activity&intitle=${searchTerm}&site=stackoverflow`;
    fetch(url)
        .then(res => res.json())
```

```
        .then(data => {
            outputItems(data.items);
        })
    ///2.2/search?order=desc&sort=activity&intitle=javascript&site=stackoverflow
})

function pageLoad() {
    const url = baseURL + '2.2/questions?order=desc&sort=activity&site=stackoverflow';

    //console.log(url);
    fetch(url)
        .then(rep => rep.json())
        .then((data) => {
            outputItems(data.items);
        })
        .catch((err) => {
            //console.log(err);
        })
}

function outputItems(data) {
    ////console.log(data);
    output.innerHTML = '';
    data.forEach(item => {
        outputPage(item);
    });
}

function outputPage(data) {
    //console.log(data);
    const main = makeNode(output, 'div', '');
    main.classList.add('box');
    const ele = makeNode(main, 'div', data.title);
    ele.classList.add('topTitle');
    const ansCount = makeNode(main, 'div', `Answers ${data.answer_count}`);
    ansCount.classList.add('ans')
    ele.qid = data.question_id;
    const quesID = makeNode(main, 'div', `QID ${ele.qid}`);
    if (data.question_id) {
        main.addEventListener('click', (e) => {
            getByID(data.question_id);
        });
    } else {
        main.style.backgroundColor = '#ddd';
```

```
    }
    data.tags.forEach((tag) => {
        const span = makeNode(main, 'span', tag);
        span.classList.add('tag');
    })
    //console.log(ele);
}

function makeNode(parent, typeEle, html) {
    const element = document.createElement(typeEle);
    element.innerHTML = html;
    return parent.appendChild(element);
}

function getByID(qid) {


    if (qid) {
        const url1 = baseURL + '2.2/questions/' + qid +
'?order=desc&sort=activity&site=stackoverflow';
        const url2 = baseURL + '2.2/questions/' + qid +
'/answers?order=desc&sort=activity&site=stackoverflow';
        let itemInfo = {};
        fetch(url1)
            .then(rep => rep.json())
            .then((data) => {
                itemInfo = data;
                return fetch(url2)
            })
            .then(res => res.json())
            .then((data) => {
                console.log(itemInfo.items[0]);
                console.log(data.items);
                buildPageData(itemInfo.items[0], data.items);
            })
            .catch((err) => {
                //console.log(err);
            })
        ////2.2/questions/55986738/answers?order=desc&sort=activity&site=stackoverflow
    } else {
        console.log('No ID');
    }
}
```

```
function buildPageData(que, ans) {
    console.log(que);
    console.log(ans);
    output.innerHTML = '';
    const title = makeNode(output, 'div', `<h2>${que.title}</h2>`);
    const qid = makeNode(output, 'div', `<div>Question ID : ${que.question_id}</div>`);
    const link = makeNode(output, 'a', `${que.link}`);
    link.setAttribute('href', que.link);
    link.setAttribute('target', '_blank');
    const total = makeNode(output, 'div', `<div>Answers  : ${que.answer_count}</div>`);
    const answerDiv = makeNode(output, 'div', '');
    answerDiv.classList.add('info');
    ans.forEach((answer, index) => {
        console.log(answer);
        const rating = answer.owner.accept_rate || '-';
        const html = `
        <hr>
        Answer # ${index+1}<br>
        Answer ID ${answer.answer_id}<br>
        Owner : ${answer.owner.display_name} (${rating})
        `;
        const div1 = makeNode(answerDiv, 'div', html);
    });
}
```

Example 11

```
<!DOCTYPE html>
<html>
<head>
    <title>JavaScript JSON</title>
    <style>
        @import url('https://fonts.googleapis.com/css2?family=Noto+Sans+JP&display=swap');
        * {
            box-sizing: border-box;
        }
        body{
            font-family: 'Noto Sans JP', sans-serif;
        }
        .box{
            width:80%;
            text-align: center;
```

```
            padding: 10px;
            margin:10px auto;
        }
        .box img{
            max-width: 100%;
            width:200px;
        }
        .box h2{
            text-transform: uppercase;
        }
        .pgs{
            padding:10px;
            border: 1px solid #ddd;
            border-radius: 10px;
            cursor: pointer;
            display: inline-block;
            width:50px;
            color:white;
            background-color: black;
            text-align: center;
        }
    </style>
</head>
<body>
    <h1>JSON</h1>
    <input type="text" class="val">
    <button class="btn">Click</button>
    <div class="output"></div>
    <script src="apps.js"></script>
</body>

</html>
const btn = document.querySelector('.btn');
const h1 = document.querySelector('h1');
const output = document.querySelector('.output');
const inputVal = document.querySelector('.val');
const page = {json:{},page:1,per:10,arr:[]};
const baseurl = 'https://restcountries.eu/rest/v2/';
btn.textContent = 'Load Pages';
inputVal.style.display = 'none';
h1.textContent = 'Load Country Info';
btn.addEventListener('click', (e) => {
    console.log('ready');
    const para = 'all';
```

```
    const url = baseurl + para;
    fetch(url)
        .then(rep => rep.json())
        .then(data => {
            createPages(data);
        })
})

function createPages(data){
    page.arr.length = 0;
    //let ptotal = data.length / page.per;
    //console.log(ptotal);
    for(let i=0;i<data.length;i+=page.per){
        let tempArr = data.slice(i,i+page.per);
        //console.log(tempArr);
        page.arr.push(tempArr);
    }
    //console.log(page);
    loadPages();

}

function loadPagination(){
    const main = createNode(output,'div','');
    for(let i=0;i<page.arr.length;i++){
        const pg = createNode(main,'div',i+1);
        pg.classList.add('pgs');
        if(page.page == i+1){
            pg.style.backgroundColor = 'red';
        }
        pg.addEventListener('click',(e)=>{
            console.log(i+1);
            page.page = i+1;
            loadPages();
        })
    }
}


function loadPages(){
    output.innerHTML = '';
    console.log(page.arr,page.page);
    page.arr[page.page-1].forEach(el => {
        pageEl(el);
```

```
    });
    loadPagination();
}

function pageEl(data){
    console.log(data);
    const main = createNode(output,'div','');
    main.classList.add('box');
    const title = createNode(main,'div',`<h2>${data.name}</h2>`);
    title.style.color = 'red';
    const flag = createNode(main,'img','');
    flag.setAttribute('src',data.flag);
    let html1 = `<div>Population : ${data.population}</div>`;
    html1 += `<div>Currency : ${data.currencies[0].name} ${data.currencies[0].symbol}</div>`;
    const stats = createNode(main,'div',html1);
}


function createNode(parent,elType,html){
    const ele = document.createElement(elType);
    parent.append(ele);
    ele.innerHTML = html;
    return ele;
}
```

Example 12

```
<!DOCTYPE html>
<html>
<head>
    <title>JavaScript JSON</title>
    <style>
        @import url('https://fonts.googleapis.com/css2?family=Noto+Sans+JP&display=swap');
        * {
            box-sizing: border-box;
        }
        body{
            font-family: 'Noto Sans JP', sans-serif;
        }
        .box{
            width:80%;
            text-align: center;
```

```
            padding: 10px;
            margin:10px auto;
            cursor: pointer;
        }
        .box img{
            max-width: 100%;
            width:200px;
        }
        .box:hover{
            background-color: blanchedalmond;
        }
        .box h2{
            text-transform: uppercase;
        }
        .info {
            width:80%;
            text-align: center;
            padding: 10px;
            font-size: 1.2em;
            margin:10px auto;
            border: 1px solid #ddd;
            border-radius: 10px;
        }

        .pgs{
            padding:10px;
            border: 1px solid #ddd;
            border-radius: 10px;
            cursor: pointer;
            display: inline-block;
            width:50px;
            color:white;
            background-color: black;
            text-align: center;
        }
    </style>
</head>
<body>
    <h1>JSON</h1>
    <input type="text" class="val">
    <button class="btn">Click</button>
    <div class="output"></div>
    <script src="apps2.js"></script>
</body>
```

```
</html>
const btn = document.querySelector('.btn');
const h1 = document.querySelector('h1');
const output = document.querySelector('.output');
const inputVal = document.querySelector('.val');
const page = {json:{},page:1,per:10,arr:[]};
const baseurl = 'https://restcountries.eu/rest/v2/';
btn.textContent = 'Search by Name';
h1.textContent = 'Search Country Info';
inputVal.value = 'united';
btn.addEventListener('click', (e) => {
    console.log('ready');
    const para = 'name/'+inputVal.value;
    const url = baseurl + para;
    fetch(url)
        .then(rep => rep.json())
        .then(data => {
            loadPages(data);
        })
})

function makeaPage(data){
    output.innerHTML = '';
    const main = createNode(output,'div','');
    main.classList.add('info');
    objOutput(data,main);
}


function objOutput(obj,parent){
    Object.keys(obj).forEach(key => {
        console.log(key); //object key
        console.log(obj[key]); //value
        console.log(typeof(obj[key])); //data type
        let val = obj[key];
        if(typeof(val) == 'object'){
            val = JSON.stringify(val);
        }
        createNode(parent,'div',val);
    })

}
```

```
function loadPages(data){
    output.innerHTML = '';
    console.log(data);

    data.forEach(el => {
        pageEl(el);
    });
}

function pageEl(data){
    console.log(data);
    const main = createNode(output,'div','');
    main.classList.add('box');
    main.addEventListener('click',(e)=>{
        makeaPage(data);
    })
    const title = createNode(main,'div',`<h2>${data.name}</h2>`);
    title.style.color = 'red';
    const title2 = createNode(main,'div',`${data.nativeName}`);
    createNode(main,'div',`${data.subregion}`);

    const flag = createNode(main,'img','');
    flag.setAttribute('src',data.flag);
    let html1 = `<div>Population : ${data.population}</div>`;
    html1 += `<div>Currency : ${data.currencies[0].name} ${data.currencies[0].symbol}</div>`;
    const stats = createNode(main,'div',html1);
}


function createNode(parent,elType,html){
    const ele = document.createElement(elType);
    parent.append(ele);
    ele.innerHTML = html;
    return ele;
}
```