

Async JS Practice Question Set 2

Instructions:

- Specific instructions have been written for each question present in this set.
- Please follow ES6 norms for writing your functions.
- You can make use of basic methods such as `.length`, `toLowerCase()`, `toUpperCase()` if needed.
- An example has been provided for fetch call related questions for your understanding.

1. Use this URL - `https://example.com/post/comments` to make a fake fetch call and list out all the emails of users on the DOM in ordered list. A fakeFetch has been provided. Use HTML, CSS & JS template in REPL or Vanilla template in CodeSandbox for this question.

```
const fakeFetch = (url) => {  
  return new Promise((resolve, reject) => {  
    setTimeout(() => {  
      if (url === "https://example.com/post/comments") {  
        resolve({  
          status: 200,  
          message: "Success",  
          data: [  
            { email: "xyz@gmail.com", commentBody: "Well done keep it up" },  
            {  
              email: "hello@outlook.com",  
              commentBody: "Good going a lot to learn from you awesome"  
            },  
            {  
              email: "ram99@gmail.com",  
              commentBody: "Anyone from Vadodara here"  
            }  
          ]  
        });  
      } else {  
        reject({  
          status: 404,  
          message: "No Commentes found."  
        });  
      }  
    }, 2000);  
  });  
};
```

// your code here

// Output on the DOM should be:

// 1. xyz@gmail.com

// 2. hello@outlook.com

// 3. ram99@gmail.com

COPY

2. Use this URL - `https://example.com/winner-team` to make a fake fetch call to the get names of winner group members and show a nice message on the DOM congratulating all of them. Use HTML, CSS & JS template in REPL or Vanilla template in CodeSandbox for this question.

```
const fakeFetch = (url) => {
  return new Promise((resolve, reject) => {
    setTimeout(() => {
      if (url === "https://example.com/winner-team") {
        resolve({
          status: 200,
          data: {
            message: "Success",
            data: ["Jhon", "Raju", "Anjali", "Sakshi"]
          }
        });
      } else {
        reject({
          status: 404,
          message: "No Users found."
        });
      }
    }, 2000);
  });
};
```

// your code here

// Output on the DOM should be:

// Congratulation to the members of winning team Jhon, Raju, Anjali, Sakshi, great work k

COPY

3. Use this URL - <https://example.com/login> to make a fake fetch call and show the status like it is authenticated or not with a message in the DOM. Use HTML, CSS & JS template in REPL or Vanilla template in CodeSandbox for this question.

```
const fakeFetch = (url) => {
  return new Promise((resolve, reject) => {
    setTimeout(() => {
      if (url === "https://example.com/login") {
        resolve({
          status: 200,
          data: {
            auth: true
          }
        });
      } else {
        reject({
          status: 404,
          message: "Status not found"
        });
      }
    }, 2000);
  });
};
```

// your code here

// Output on the DOM should be:

// Verified

COPY

4. Use this URL - <https://example.com/order/status/OR00A12> to make a fake fetch call and show the order status using the user name on the DOM. Use HTML, CSS & JS template in REPL or

Vanilla template in CodeSandbox for this question.

```
const fakeFetch = (url) => {
  return new Promise((resolve, reject) => {
    setTimeout(() => {
      if (url === "https://example.com/order/status/OR00A12") {
        resolve({
          status: 200,
          data: {
            order: {
              orderId: "OR00A12",
              status: "delivery pending",
              userName: "Kajal Kumari"
            }
          }
        });
      } else {
        reject({
          status: 404,
          message: "No order found"
        });
      }
    }, 2000);
  });
};
```

// your code here

// Output on the DOM should be:

// Hello Kajal Kumari your order status is delivery pending.

[COPY](#)

5. Use this URL - <https://example.com/photo> to make a fake fetch call and show an image on the DOM using the photo link received in the response. Use HTML, CSS & JS template in REPL or Vanilla template in CodeSandbox for this question.

```
const fakeFetch = (url) => {
  return new Promise((resolve, reject) => {
    setTimeout(() => {
      if (url === "https://example.com/photo") {
        resolve({
          status: 200,
          data: {
            photo: {
              link: "https://source.unsplash.com/featured/300x201",
              title: "Random Image"
            }
          }
        });
      } else {
        reject({
          status: 404,
          message: "No order found"
        });
      }
    }, 2000);
  });
};
```

// your code here

// Output: an image on the DOM

COPY

6. Use this URL - <https://example.com/api/productlist> to make a fake fetch call and print the total price of all the products. A fakeFetch has been provided. Use HTML, CSS & JS template in REPL or Vanilla template in CodeSandbox for this question.

```
const fakeFetch = (url) => {
  return new Promise((resolve, reject) => {
    setTimeout(() => {
      if (url === "https://example.com/api/productlist") {
        resolve({
          status: 200,
          message: "Success",
          data: [
            { itemName: "Shoes", price: 100, quantity: 2 },
            { itemName: "Hat", price: 350, quantity: 1 },
            { itemName: "Tshirt", price: 410, quantity: 5 }
          ]
        });
      } else {
        reject({
          status: 404,
          message: "Items list not found."
        });
      }
    }, 2000);
  });
};
```

// Your Code here

*// Output on the DOM should be:
// Total: INR 2600*

COPY

7. Use this URL - <https://example.com/api/users> to make a fake fetch call and handle errors if any. Show a proper message to the user on the DOM (Oops! Unexpected Error. Please try again.), as per the status received from the server. The error should be displayed in red colour. A fakeFetch has been provided. Use HTML, CSS & JS template in REPL or Vanilla template in CodeSandbox for this question.

```
const fakeFetch = (url) => {
  return new Promise((resolve, reject) => {
    setTimeout(() => {
      if (url === "https://example.com/api/users") {
        reject({
          status: 500,
          message: "Internal Server Error"
        });
      } else {
        resolve({
          status: 200,
          data: {
            message: "Success"
          }
        });
      }
    }, 2000);
  });
};
```

```

});
};

// Your Code here

// Output on the DOM should be (in red color):
// Ops. Unexpected Error. Please try again.

```

COPY

8. Use this URL - <https://example.com/api/allbooks> to make a fake fetch call and handle errors if any. Show a proper message to the user on the DOM (Bad Request! Requested size too large.), as per the status received from the server. A fakeFetch has been provided. Use HTML, CSS & JS template in REPL or Vanilla template in CodeSandbox for this question.

```

const fakeFetch = (url) => {
  return new Promise((resolve, reject) => {
    setTimeout(() => {
      if (url === "https://example.com/api/allbooks") {
        reject({
          status: 400,
          message: "Bad Request"
        });
      } else {
        resolve({
          status: 200,
          data: {
            message: "Success"
          }
        });
      }
    }, 2000);
  });
};

// Your Code here

// Output on the DOM should be:
// Bad Request! Requested size too Large.

```

COPY

9. Use this URL - <https://example.com/welcome> to make a fake fetch call and display a welcome message to the user on the DOM. Welcome message (if provided) should be used else the default message of welcome to the server should be shown. Message should be only shown if the user is logged in. A fakeFetch has been provided. Use HTML, CSS & JS template in REPL or Vanilla template in CodeSandbox for this question.

```

const fakeFetch = (url) => {
  return new Promise((resolve, reject) => {
    setTimeout(() => {
      if (!(url === "https://example.com/welcome")) {
        reject({
          status: 511,
          message: "Network auth required"
        });
      } else {
        resolve({
          status: 200,
          data: {

```

```

        logged: true
      }
    });
  }
}, 2000);
});
};

```

// your code here

// Output: As per the response from server

COPY

10. Use this URL - <https://example.com/getImage> to make a fake fetch call which takes a url and dimensions for the photo to be displayed. Dimensions should be passed in the format [width, height]. Show the image provided in the response on DOM. Use HTML, CSS & JS template in REPL or Vanilla template in CodeSandbox for this question.

Dimensions can be 200/300/100. For example you can pass [200, 200] or [200, 300], etc.

```

const fakeFetch = (url, dimensions) => {
  return new Promise((resolve, reject) => {
    setTimeout(() => {
      if (url === "https://example.com/getImage") {
        resolve({
          status: 200,
          data: {
            image: {
              link: `https://picsum.photos/${dimensions[0]}/${dimensions[1]}`,
              title: "Random Image"
            }
          }
        });
      } else {
        reject({
          status: 404,
          message: "No photo of such dimension found"
        });
      }
    }, 2000);
  });
}

```

// your code here

// Output on the DOM should be an image.

COPY

- Well Done completed all the sets of practice set.