Practically prepare for your JavaScript interview

JavaScript Revision

JavaScript-Concept Based Problems

Data Structures

Algorithms

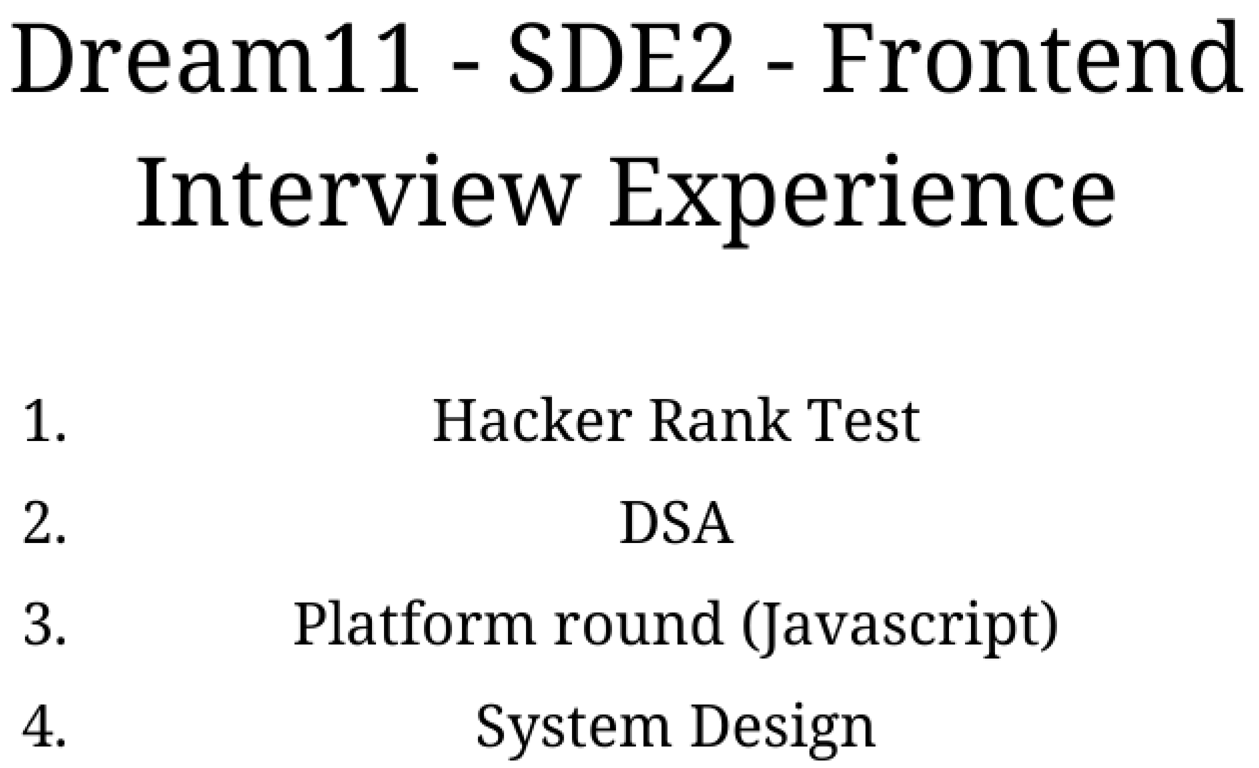
Machine Coding

Web Fundamentals

Dream11 – SDE2 – Frontend Interview Experience

Posted on March 29, 2021 | by Prashant Yadav

Posted in Interview, Tech



Dream11 - SDE2 - Frontend Interview Experience

1. Hacker Rank Test
2. DSA
3. Platform round (Javascript)
4. System Design

I had applied on Dream11 career section for SDE2 Frontend position in March 2021.

Round 1: Hacker rank test

3 questions to be solved in 90 minutes.

- Convert binary linked list to decimal.
- Coin change problem.
- Last stone weight

Round 2: DSA

- Largest contiguous subarray.
- Common nodes in two binary search tree.

Round 3: Platform round (Javascript)

1. Implement a function onlyTwice which stores two instances a function invocation and returns first on odd calls and second on even calls.

It was based on singleton design pattern.

Copy

```
const addTwoNumbers = (a, b) => a + b
const myFancyAdd = onlyTwice(addTwoNumbers)

console.log(myFancyAdd(2, 3)) // 5
console.log(myFancyAdd(1, 2)) // 3
console.log(myFancyAdd(3, 4)) // 5
console.log(myFancyAdd(3, 7)) // 3
```

My answer

Copy

```
const onlyTwice = (fn) => {
  let isOdd = true;
  let first = null;
  let second = null;

  return function(...args) {

    if(isOdd){
      if(!first){
        first = fn(...args);
      }

      isOdd = false;
      return first;
    }else{
      if(!second){
        second = fn(...args);
      }

      isOdd = true;
      return second;
    }
  }
}
```

2. Create throttle function.

3. Create a polyfill for promise which should handle following edge cases.

Copy

```
const prom = new MyPromise((resolve, reject)=>{
  setTimeout(() => {
    resolve("Done");
  }, 1000);
});


prom.then(function(data){
  console.log('1' + data) // Prints "1 Done" after 1 second
})

prom.then(function(data){
  console.log('2' + data) // Prints "2 Done" after 1 second
})

setTimeout(function(){
  prom.then(function(data){
    console.log('3' + data) // Prints "3 Done" after 2 seconds
  })
}, 2000)


const nwPromise = new Promise(()=>{
  xhr.send();

  xhr.onReadyStateChange = (data)=> {
    resolve(data)
  }

  xhr.abort()
})

nwPromise.cancel()
```

My Code

```
const MyPromise = function(fn){
  // Code here

  let result = null;
  let error = null;
  let thenCallBackFunction = [];
  let resolveState = 0;

  const resolve = (data) => {
    resolveState = 1;

    result = data;
    if(thenCallBackFunction.length > 0){
      for(let fn of thenCallBackFunction){
        fn(data);
      }

      thenCallBackFunction = [];
    }

    resolveState = 0;
  }

  const reject = (error) => {
    error = error;
  }

  this.then = (fn2) => {
    if(!result){
      thenCallBackFunction.push(fn2);
    }else{
      fn2(result);
    }
  }

  this.catch = (errorFn) => {
    errorFn(error);
  }

  this.cancel = (cancelFn) => {
    if(resolveState === 0){

    }
  }

  fn(resolve, reject);
}
```

I did not implemented the `.cancel()` as there was no time left.

Round 4: System design

Given the following API endpoints create a book reader.

```
/books (List of books with book IDs)
/book/:book_id/page_count (Returns page count for the requested book)
/book/:book_id/page/:page_number (Returns content for the requested page in RTE)
```

1. It will show the list of books
2. On click open the selected book
3. Scroll to the next page (no pagination)

Lots of cross question on

1. Handle latency
2. Debounce
3. Direct jump to random page
4. Caching pages
5. Pre-loading data
6. Buffer zone
7. Optimization.

I failed in the final in this round as the interviewer asked me that if I use debounce (2ms) to load the next page and consider that you are on a 2g network and there is delay of 10ms for response then how will you handle it. I got stuck here.

Also he asked me that suppose your RAM will be able to store only 20 pages at a time then how would you jump to 300th page from the 100th page, when will you make the api call to fetch the page, will you show blank page or not, your scroll should end at 300th page, etc, etc.

Recommended Posts:

How I got my first job after failing for more than 300 times.
Frontend Interview Process at Coursera
Frontend Interview Process at Atlassian
Fullstack interview experience at Nutanix
OLA-SDE2-Frontend Interview Experience
Frontend Interview Experience at Amazon Germany
In India more money is spent on import of semiconductors than on oil
Top 5 High-paying programming languages to learn in 2024
5 reasons to learn javascript in 2019
Reliance JIO interview experience

Prev
Next