字节前端面试

面试重点

自我介绍，是一场面试的核心，面试官很多问题都源自你的自我介绍中。所以自我介绍的时候一定要捡你会的说。

面试题

这次三轮面试总共做了四道面试题。

实现一个抓包请求

这块一开始没了解清楚面试官的要求，然后具体问了下，最终理解下来是需要实现一个并发限制功能。

```
function asyncPool(poolLimit, array, iteratorFn) {
  let i = 0;
  const ret = [];
  const executing = [];
  const enqueue = function () {
    if (i === array.length) {
      return Promise.resolve();
    }
    const item = array[i++];
    const p = Promise.resolve().then(() => iteratorFn(item, array));
    ret.push(p);
    const e = p.then(() => executing.splice(executing.indexOf(e), 1));
    executing.push(e);
    let r = Promise.resolve();
    if (executing.length >= poolLimit) {
      r = Promise.race(executing);
    }
    return r.then(() => enqueue());
  };
  return enqueue().then(() => Promise.all(ret));
}
```

实现一个防抖功能

so easy！这个应该是大家都会的吧。

```
function debounce(fn, wait) {
  let timeout = null;
  return function () {
    if (timeout !== null) clearTimeout(timeout);
    timeout = setTimeout(fn, wait);
  };
}
```

拓展：现在有一个搜索输入框，已加上防抖功能，但是恰巧后一个请求先回来怎么办?

提供一个简单的思路，给每个请求复制给一个id，然后每次请求回来的时候判断下id是不是最新的。

实现一个图片懒加载

这里就不实现了，提供一个思维方式，图片懒加载中，最重要的一个步骤就是，在页面滚动的时候判断图片即将进入窗口。

我们可以判断图片在屏幕外，然后取反，就可以了。

实现一个异步求和函数

提供一个异步add方法如下，需要实现一个await sum(...args)函数;

```
function asyncAdd(a, b, callback) {
  setTimeout(function () {
    callback(null, a + b);
  }, 1000);
}
```

实现方式如下：

```
async function sum(...args) {
  if (args.length > 1) {
    const result = await new Promise((resolve) => {
      asyncAdd(args[0], args[1], (err, result) => {
        if (!err) {
          resolve(result);
        }
      });
    });
    return sum(result, ...args.splice(2));
  }
  return args[0];
}
```

认真看的同学应该就能发现，当前版本存在一个优化点，计算时长可以缩短。优化版本如下：

```
function createAdd(a, b = 0) {
  return new Promise((resolve) => {
    asyncAdd(a, b, (err, result) => {
      if (!err) {
        resolve(result);
      }
    });
  });
}

async function sum(...args) {
  if (args.length > 1) {
    const result = [];
    for (let i = 0; i < args.length; i = i + 2) {
      result.push(createAdd(args[i], args[i + 1]));
    }
    return sum(...(await Promise.all(result)));
  }
  return args[0];
}
```

面试总结