【前端23校招面经】瓜子二手车一面

1. 自我介绍
2. 讲`git merge`和`git rebase`的区别
3. 多人用`git`同时开发, 讲一下线上发布流程, 如何保证团队成员安全 merge 代码
4. 讲一下`CDN`回源策略
5. `https`协议
6. `Vue`和`React`区别
7. `cookie` / `localStorage` / `sessionStorage`区别
8. 浏览器缓存原理
9. http 状态码
10. 跨域方案
11. 手写`Promise.all`

```
function myAll(iterators) {
  const promises = Array.from(iterators);
  const values = [];
  return new Promise((resolve, reject) => {
    promises.forEach(p => {
      p.then((data, err) => {
        if (err) {
          reject(err);
        }
        values.push(data);
        if (values.length === promises.length) {
          resolve(values);
        }
      });
    });
  });
}
```

12. 算法题: 查找两个数组的交集元素

```
function intersection<T>(arr1: T[], arr2: T[]): T[] {
  const shortArr = arr1.length < arr2.length ? arr1 : arr2;
  const longArr = shortArr === arr1 ? [...arr2] : [...arr1]; // 防止后续 splice 操作
  const result = [];
  shortArr.forEach(item => {
    const index = longArr.findIndex(i => i === item);
    if (index !== -1) {
      result.push(item);
    }
    longArr.splice(index, 1);
  });
  return result;
}

console.log(intersection([1, 2, 2, 1], [2, 2]));
console.log(intersection([1, 2, 1], [2, 2]));
```

测试用例:

```
[1, 2, 2, 1], [2, 2] --> [2, 2]
[1, 2, 1], [2, 2] --> [2]
```

> 注意不能直接用 Set 莽干, 因为有些情况下不能去重