HTML

1. 必考：你是如何理解 HTML 语义化的？
   i. 举例法
   HTML 语义化就是使用正确的标签（总结）段落就写 p 标签，标题就写 h1 标签，文章就写article标签，视频就写video标签，等等。
   ii. 阐述法
   首先讲以前的后台开发人员使用table布局，然后讲美工人员使用div+css布局，最后讲专业的前端会使用正确的标签进行页面开发。
2. meta viewport 是做什么用的，怎么写？
   举例法
   ```
   <meta name="viewport" content="width=device-width, initial-scale=1, maximum-scale=1, minimum-scale=1">
   ```
   然后逐个解释每个单词的意思。
3. 你用过哪些 HTML 5 标签？
   举例法
   平时如果只用div写页面你就完了，把你平时用到的html5标签列举出来即可，但是要注意如果这个标签的用法比较复杂，你要先看一下MDN的文档再说这个标签；如果你说出一个标签，却不知道它有哪些API，那么你就会被扣分
4. H5 是什么？
   阐述法
   搜一下知乎就知道了，H5表示移动端页面，反正不是HTML5。

CSS

1. 必考：两种盒模型分别说一下。
   先说两种盒模型分别怎么写，具体到代码。然后说你平时喜欢用border box，因为更好用。
2. 必考：如何垂直居中？
   背代码 https://jscode.me/t/topic/1936
3. 必考：flex 怎么用，常用属性有哪些？
   看 MDN，背代码。
4. 必考：BFC 是什么？
   背 BFC 触发条件，MDN 写了。
   但是不用全部背下来，面试官只知道其中几个：
   i. 浮动元素（元素的 float 不是 none）
   ii. 绝对定位元素（元素的 position 为 absolute 或 fixed）
   iii. 行内块元素
   iv. overflow 值不为 visible 的块元素
   v. 弹性元素（display为 flex 或 inline-flex元素的直接子元素）
5. CSS 选择器优先级
   i. 背人云亦云的答案（错答案、已过时）：https://www.cnblogs.com/xugang/archive/2010/09/24/1833760.html
   ii. 看面试官脸色行事
   iii. 方方给的三句话
      a. 越具体优先级越高
      b. 同样优先级写在后面的覆盖写在前面的
      c. !important 优先级最高，但是要少用
6. 清除浮动说一下
   背代码

```
.clearfix:after{
    content: '';
    display: block; /*或者 table*/
    clear: both;
}
.clearfix{
    zoom: 1; /* IE 兼容*/
}
```

原生 JS

1. 必考：ES 6 语法知道哪些，分别怎么用？
   举例法
   let const 箭头函数 Promise 展开操作符 默认参数 import export，见方方整理的列表
2. 必考 Promise、Promise.all、Promise.race 分别怎么用？
   i. 背代码 Promise 用法

```
function fn(){
    return new Promise((resolve, reject)=>{
        成功时调用 resolve(数据)
        失败时调用 reject(错误)
    })
}
fn().then(success, fail).then(success2, fail2)
```

   ii. 背代码 Promise.all 用法

```
Promise.all([promise1, promise2]).then(success1, fail1)
```

   promise1和promise2都成功才会调用success1
   iii. 背代码 Promise.race 用法

```
Promise.race([promise1, promise2]).then(success1, fail1)
```

   promise1和promise2只要有一个成功就会调用success1；
   promise1和promise2只要有一个失败就会调用fail1；
   总之，谁第一个成功或失败，就认为是race的成功或失败。
3. 必考：手写函数防抖和函数节流
   i. 背代码

```
// 节流（一段时间执行一次之后，就不执行第二次）
function throttle(fn, delay){
    let canUse = true
    return function(){
        if(canUse){
            fn.apply(this, arguments)
            canUse = false
            setTimeout(()=>canUse = true, delay)
        }
    }
}

const throttled = throttle(()=>console.log('hi'))
throttled()
throttled()
```

   注意，有些地方认为节流函数不是立刻执行的，而是在冷却时间末尾执行的（相当于施法有吟唱时间），那样说也是对的。
   ii. 背代码

```
// 防抖（一段时间会等，然后带着一起做了）
function debounce(fn, delay){
    let timerId = null
    return function(){
        const context = this
        if(timerId){window.clearTimeout(timerId)}
        timerId = setTimeout(()=>{
            fn.apply(context, arguments)
            timerId = null
        },delay)
    }
}
const debounced = debounce(()=>console.log('hi'))
debounced()
debounced()
```

4. 必考：手写AJAX
   i. 背代码，完整版

```
var request = new XMLHttpRequest()
request.open('GET', '/a/b/c?name=ff', true);
request.onreadystatechange = function () {
  if(request.readyState === 4 && request.status === 200) {
    console.log(request.responseText);
  }};
request.send();
```

   ii. 背代码，简化版

```
var request = new XMLHttpRequest()
request.open('GET', '/a/b/c?name=ff', true)
request.onload = ()=> console.log(request.responseText)
request.send()
```

5. 必考：这段代码里的 this 是什么？
   i. 背代码
      a. fn()
      this => window/global
      b. obj.fn()
      this => obj
      c. fn.call(xx)
      this => xx
      d. fn.apply(xx)
      this => xx
      e. fn.bind(xx)
      this => xx
      f. new Fn()
      this => 新的对象
      g. fn = ()=> {}
      this => 外面的 this
   ii. 看调用
   《this 的值到底是什么？一次说清楚》
6. 必考：闭包/立即执行函数是什么？
   i. 闭包 https://zhuanlan.zhihu.com/p/22486908
   ii. 立即执行函数 https://zhuanlan.zhihu.com/p/22465092
7. 必考：什么是 JSONP，什么是 CORS，什么是跨域？
   饥人谷系统班全都有讲，没有报名的同学自己搜文章看
   i. JSONP https://zhuanlan.zhihu.com/p/22600501
   ii. CORS https://developer.mozilla.org/zh-CN/docs/Web/HTTP/Access_control_CORS
8. 常考：async/await 怎么用，如何捕获异常？
   i. 阮一峰的书讲了
   ii. 方方的视频课讲了 最后一节。
9. 常考：如何实现深拷贝？
   背代码，要点：
   i. 递归
   ii. 判断类型
   iii. 检查环（也叫循环引用）
   iv. 需要忽略原型
10. 常考：如何用正则实现 trim()？
   背代码

```
String.prototype.trim = function(){
    return this.replace(/^\s+|\s+$/g, '')
}
//或者
function trim(string){
    return string.replace(/^\s+|\s+$/g, '')
}
```

11. 常考：不用 class 如何实现继承？用 class 又如何实现？
   i. 背代码，不用 class 这样实现

```
function Animal(color){
    this.color = color
}
Animal.prototype.move = function(){} // 动物可以动
function Dog(color, name){
    Animal.call(this, color) // 或者 Animal.apply(this, arguments)
    this.name = name
}
// 下面三行实现 Dog.prototype.__proto__ = Animal.prototype
function temp(){}
temp.prototype = Animal.prototype
Dog.prototype = new temp()

Dog.prototype.constuctor = Dog // 这行看不懂就算了，面试官也不问
Dog.prototype.say = function(){ console.log('汪')}

var dog = new Dog('黄色','阿黄')
```

   ii. 背代码，用 class 就简单了

```
class Animal{
    constructor(color){
        this.color = color
    }
    move(){}
}
class Dog extends Animal{
    constructor(color, name){
        super(color)
        this.name = name
    }
    say(){}
}
```

12. 常考：如何实现数组去重？
   i. 计数排序变形，背代码
   ii. 使用 Set（面试已经禁止这种了，因为太简单）
   iii. 使用 WeakMap
13. 放弃：== 相关题目（反着答）
   不要背，记不住，太复杂且没有规律
14. 送命题：手写一个 Promise
   提前写一遍，放在博客里，参考 https://juejin.im/post/5aafe3edf265da238f125c0a

DOM

1. 必考：事件委托
   i. 错误版（但是可能能过）

```
ul.addEventListener('click', function(e){
    if(e.target.tagName.toLowerCase() === 'li'){
        fn() // 执行某个函数
    }
})
```

   bug 在于，如果用户点击的是 li 里面的 span，就没法触发 fn，这显然不对。
   ii. 高级版

```
function delegate(element, eventType, selector, fn) {
    element.addEventListener(eventType, e => {
      let el = e.target
      while (!el.matches(selector)) {
        if (element === el) {
          el = null
          break
        }
        el = el.parentNode
      }
      el && fn.call(el, e, el)
    })
    return element
  }
```

   思路是点击 span 后，递归遍历 span 的祖先元素看其中有没有 ul 里面的 li。
2. 曾考：用 mouse 事件写一个可拖曳的 div
   参考代码：https://jsbin.com/munuzureya/edit?html,js,output

HTTP

3. 必考：HTTP 状态码知道哪些？分别什么意思？
   - 2xx 表示成功
   - 3xx 表示需要进一步操作
   - 4xx 表示浏览器方面出错
   - 5xx 表示服务器方面出错
   - 完整参考 http://www.runoob.com/http/http-status-codes.html
4. 大公司必考：HTTP 缓存有哪几种？
   - 需要详细的了解 ETag、CacheControl、Expires 的异同
   - 参考 https://imweb.io/topic/5795dcb6fb312541492eda8c
   - 答题要点：
      a. ETag 是通过对比浏览器和服务器资源的特征值（如MD5）来决定是否要发送文件内容，如果一样就只发送 304（not modified）
      b. Expires 是设置过期时间（绝对时间），但是如果用户的本地时间错乱了，可能会有问题
      c. CacheControl: max-age=3600 是设置过期时长（相对时间），跟本地时间无关。
5. 必考：GET 和 POST 的区别
   i. 错解，但是能过面试
      - GET在浏览器回退时是无害的，而POST会再次提交请求。
      - GET产生的URL地址可以被加入收藏栏，而POST不可以。
      - GET请求会被浏览器主动cache，而POST不会，除非手动设置。
      - GET请求只能进行url编码，而POST支持多种编码方式。
      - GET请求参数会被完整保留在浏览器历史记录里，而POST中的参数不会被保留。
      - GET请求在URL中传送的参数是有长度限制的，而POST么有。
      - 对参数的数据类型，GET只接受ASCII字符，而POST没有限制。
      - GET比POST更不安全，因为参数直接暴露在URL上，所以不能用来传递敏感信息。
      - GET参数通过URL传递，POST放在Request body中。
   ii. 正解
   就一个区别：语义——GET 用于获取资源，POST 用于提交资源。
   iii. 想装逼请参考 https://zhuanlan.zhihu.com/p/22536382
6. Cookie V.S. LocalStorage V.S. SessionStorage V.S. Session
   - Cookie V.S. LocalStorage
      a. 主要区别是 Cookie 会被发送到服务器，而 LocalStorage 不会
      b. Cookie 一般最大 4k，LocalStorage 可以用 5Mb 甚至 10Mb（各浏览器不同）
   - LocalStorage V.S. SessionStorage
      a. LocalStorage 一般不会自动过期（除非用户手动清除），而 SessionStorage 在回话结束时过期（如关闭浏览器）
   - Cookie V.S. Session
      a. Cookie 存在浏览器的文件里，Session 存在服务器的文件里
      b. Session 是基于 Cookie 实现的，具体做法就是把 SessionID 存在 Cookie 里

框架 Vue

1. 必考：watch 和 computed 和 methods 区别是什么？
   i. 思路：先翻译单词，再阐述作用，最后强行找不同。
   ii. 要点：
      a. computed 和 methods 相比，最大区别是 computed 有缓存：如果 computed 属性依赖的属性没有变化，那么 computed 属性就不会重新计算。methods 则是看到一次计算一次。
      b. watch 和 computed 相比，computed 是计算出一个属性（废话），而 watch 则可能是做别的事情（如上报数据）
2. 必考：Vue 有哪些生命周期钩子函数？分别有什么用？
   i. 钩子在文档全都有，看红色的字。
   ii. 把名字翻译一遍就是满分
   iii. 要特别说明哪个钩子里请求数据，答案是 mounted
3. 必考：Vue 如何实现组件间通信？
   i. 父子组件：使用 v-on 通过事件通信
   ii. 爷孙组件：使用两次 v-on 通过爷爷爸爸通信，爸爸儿子通信实现爷孙通信
   iii. 任意组件：使用 eventBus = new Vue() 来通信，eventBus.$on 和 eventBus.$emit 是主要API
   iv. 任意组件：使用 Vuex 通信
4. 必考：Vue 数据响应式怎么做到的？
   i. 答案在文档《深入响应式原理》
   ii. 要点
      a. 使用 Object.defineProperty 把这些属性全部转为 getter/setter
      b. Vue 不能检测到对象属性的添加或删除，解决方法是手动调用 Vue.set 或者 this.$set
5. 必考：Vue.set 是做什么用的？
   见上一题
6. Vuex 你怎么用的？
   i. 背下文档第一句：Vuex 是一个专为 Vue.js 应用程序开发的状态管理模式
   ii. 说出核心概念的名字和作用：State/Getter/Mutation/Action/Module
7. VueRouter 你怎么用的？
   i. 背下文档第一句：Vue Router 是 Vue.js 官方的路由管理器。
   ii. 说出核心概念的名字和作用：History 模式/导航守卫/路由懒加载
   iii. 说出常用 API：router-link/router-view/this.$router.push/this.$router.replace/this.$route.params

```
this.$router.push('/user-admin')
this.$route.params
```

8. 路由守卫是什么？
   看官方文档的例子，背里面的关键的话

框架 React

1. 必考：受控组件 V.S. 非受控组件

```
<FInput value={x} onChange={fn}/> 受控组件
<FInput defaultValue={x} ref={input}/> 非受控组件
```

   区别受控组件的状态由开发者维护，非受控组件的状态由组件自身维护（不受开发者控制）
2. 必考：React 有哪些生命周期函数？分别有什么用？（Ajax 请求放在哪个阶段？）
   答题思路跟 Vue 的一样
   i. 钩子在文档里，蓝色框框里面的都是生命周期钩子
   ii. 把名字翻译一遍就是满分
   iii. 要特别说明哪个钩子里请求数据，答案是 componentDidMount
3. 必考：React 如何实现组件间通信？
   i. 父子靠 props 传函数
   ii. 爷孙可以穿两次 props
   iii. 任意组件用 Redux（也可以自己写一个 eventBus）
4. 必考：shouldComponentUpdate 有什么用？
   i. 要点：用于在没有必要更新 UI 的时候返回 false，以提高渲染性能
   ii. 参考：http://taobaofed.org/blog/2016/08/12/optimized-react-components/
5. 必考：虚拟 DOM 是什么？
   i. 要点：虚拟 DOM 就是用来模拟 DOM 的一个对象，这个对象拥有一些重要属性，并且更新 UI 主要就是通过对比（DIFF）旧的虚拟 DOM 树 和新的虚拟 DOM 树的区别完成的。
   ii. 参考：http://www.alloyteam.com/2015/10/react-virtual-analysis-of-the-dom/
6. 必考：什么是高阶组件？
   i. 要点：文档原话——高阶组件就是一个函数，且该函数接受一个组件作为参数，并返回一个新的组件。
   ii. 举例：React-Redux 里 connect 就是一个高阶组件，比如 `connect(mapState)(MyComponent)` 接受组件 MyComponent，返回一个具有状态的新 MyComponent 组件。
7. React diff 的原理是什么？
   看你记忆力了：https://imweb.io/topic/579e33d693d9938132cc8d94
8. 必考 Redux 是什么？
   i. 背下文档第一句：Redux 是 JavaScript 状态容器，提供可预测化的状态管理。重点是『状态管理』。
   ii. 说出核心概念的名字和作用：Action/Reducer/Store/单向数据流
   iii. 说出常用 API：store.dispatch(action)/store.getState()
9. connect 的原理是什么？
   react-redux 库提供的一个 API，connect 的作用是让你把组件和store连接起来，产生一个新的组件（connect 是高阶组件）
   参考：https://segmentfault.com/a/1190000017064759

TypeScript

1. never 类型是什么？
   不应该出现的类型 尤雨溪的答案：https://www.zhihu.com/question/354601204/answer/888551021
2. TypeScript 比起 JavaScript 有什么优点？
   提供了类型约束，因此更可控、更容易重构、更适合大型项目、更容易维护

Webpack

1. 必考：有哪些常见 loader 和 plugin，你用过哪些？
2. 英语题：loader 和 plugin 的区别是什么？
3. 必考：如何按需加载代码？
4. 必考：如何提高构建速度？
5. 转义出的文件过大怎么办？

上面五题请看这个不错的参考：https://zhuanlan.zhihu.com/p/44438844

安全

1. 必考：什么是 XSS？如何预防？
   比较复杂，看我的文章 https://zhuanlan.zhihu.com/p/22500730
2. 必考：什么是 CSRF？如何预防？
   比较复杂，看若愚的文章 https://zhuanlan.zhihu.com/p/22521378

开放题目

1. 必考：你遇到最难的问题是怎样的？
   要点：一波三折。参考 https://www.zhihu.com/question/35323603
2. 你在团队的突出贡献是什么？
   把小事说大。
3. 最近在关注什么新技术
   书、博客、推特、知乎，不要说 CSDN、百度。
4. 有没有看什么源码，看了后有什么记忆深刻的地方，有什么收获
   看过源码说源码，推荐看 underscore.js 的源码
   没看过源码就说同事的代码，代码烂就说哪里烂，代码好就说哪里好
   收获：命名规范、设计模式

刁钻题目

1. 代码

```
[1,2,3].map(parseInt)
```

   答案

```
1
NaN
NaN
```

2. 代码

```
var a = {name: 'a'}
a.x = a = {}
问 a.x 是多少？
```

   答案

```
undefined
```

3. `(a ==1 && a== 2 && a==3)` 可能为 true 吗？
   i. 利用 == 会调用 valueOf() 的特性

```
var a = {
 value: 1,
 valueOf(){
  return this.value++
 }
}
a ==1 && a== 2 && a==3 // true
```

   ii. 利用 a 会读取 window.a 的特性

```
var value = 1;
Object.defineProperty(window, 'a', {
    get(){
        return value++;
    }
})
a ==1 && a== 2 && a==3 // true
// 或者
a ===1 && a=== 2 && a===3 // true
```

超纲题

1. JS 垃圾回收机制
   i. 看图讲解 https://javascript.info/garbage-collection
      a. 什么是垃圾
      b. 如何捡垃圾（遍历和计数，只是不同的算法而已）
      c. 前端又有其特殊性（JS进程和DOM进程）
   ii. 更深入一些的讲解 http://newhtml.net/v8-garbage-collection/
2. Eventloop 说一下

```
setTimeout(function () {
    console.log(4);
}, 0);
new Promise(function (resolve) {
    console.log(1);
    resolve();
    console.log(2);
}).then(function () {
    console.log(5);
});

console.log(3);
```

```
1
2
3
5
4
```

   i. 肤浅理解：『一会儿』和『尽快』异步任务
   ii. 详细理解：Eventloop 是个啥？
   iii. 浏览器有 Eventloop 吗？
   iv. 每个 API 对应哪个任务队列？
      a. setTimeout
      b. setImmediate（浏览器没有）
      c. process.nextTick（浏览器没有）
      d. MutationObserver（Node 没有）
      e. promise.then
      f. await
   v. 这种题目尽量说思路，因为你不可能通过眼睛看出结果（必须画图）

```
async function async1() {
    console.log(1);
    await async2();
    console.log(2);
}
async function async2() {
    console.log(3)
}

async1();

new Promise(function (resolve) {
    console.log(4);
    resolve();
}).then(function () {
    console.log(5);
});
```

```
1
3
4
2
5
```

   注意：这一题的答案不唯一，在 Node.js 和 Chrome 的结果不一样，甚至在 Chrome 上也是时而这个答案，时而那个答案。所以还是说思路最重要。

个性化题目

- PWA
- echarts.js / d3.js
- three.js
- flutter
- SSR

做个 hello world 基本就能应付面试了，如果怕应付不了，就再做个复杂点的。