

## ▽ Web 前置知识

- [《老生常谈的从 URL 输入到页面展现背后发生的事》](#) [编号: web\_01]

涉及面试题：

1. 从 URL 输入到页面展现背后发生了什么事？
2. 一次完整的 HTTP 事务是怎么一个过程？
3. 浏览器是如何渲染页面的？
4. 浏览器的内核有哪些？分别有什么代表的浏览器？
5. 刷新页面，JS 请求一般会有哪些地方有缓存处理？

- [《初次接触前端，我们要理解哪些名词？》](#)
- [《工欲善其事，必先利其器——软件安装、环境搭建》](#)
- [《做一次山大王，让操作系统乖得像只小绵羊——命令行入门》](#)
- [《用李涛高手之路前 3 节课开启我们光与色的大门》](#)
- [《Git 和 GitHub：① Git、GitHub 初认识》](#)
- [《Git 和 GitHub：② 提交代码+团队合作》](#)

## ▽ HTML

- [《① HTML 基础》](#) [编号: html\_01]

涉及面试题：

1. DOCTYPE 有什么作用？怎么写？
2. 列出常见的标签，并简单介绍这些标签用在什么场景？
3. 页面出现了乱码，是怎么回事？如何解决？
4. title 属性和 alt 属性分别有什么作用？
5. HTML 的注释怎样写？
6. HTML5 为什么只写 <!DOCTYPE html> ？
7. data- 属性的作用？
8. <img> 的 title 和 alt 有什么区别？
9. Web 标准以及 W3C 标准是什么？
10. DOCTYPE 作用？严格模式与混杂模式如何区分？它们有何意义？
11. HTML 全局属性 (Global Attribute) 有哪些？

- [《② HTML 元素、属性详解》](#) [编号: html\_02]

涉及面试题：

1. meta 有哪些常见的值？
2. meta viewport 是做什么用的，怎么写？
3. 列出常见的标签，并简单介绍这些标签用在什么场景？
4. 如何在 HTML 页面上展示 `<div></div>` 这几个字符？
5. 你是如何理解 HTML 语义化的？
6. 前端需要注意哪些 SEO？
7. 你对网页标准和 W3C 重要性的理解？

- [《③ HTML 表单详解》](#) [编号：html\_03]

涉及面试题：

1. POST 和 GET 方式提交数据有什么区别？
2. 在 input 里，name 有什么作用？
3. label 有什么作用？如何使用？
4. radio 如何分组？
5. placeholder 属性有什么作用？
6. `type=hidden` 隐藏域有什么作用？举例说明。
7. CSRF 攻击是什么？如何防范？
8. 网页验证码是干嘛的？是为了解决什么安全问题？
9. 常见 Web 安全及防护原理？

## ▼ CSS

- [《CSS 基础与选择器初识》](#) [编号：css\_01]

涉及面试题：

1. CSS 加载方式有几种？
2. @import 有什么作用？如何使用？
3. CSS 选择器常见的有几种？
4. id 选择器和 class 选择器的使用场景分别是什么？
5. @charset 有什么作用？
6. 简述 src 和 href 的区别？
7. 页面导入时，使用 link 和 @import 有什么区别？

- [《CSS 选择器详解》](#) [编号: css\_02]

涉及面试题:

1. 伪类选择器有哪些?
2. 伪元素和伪类的区别?

- [《CSS 结构和层叠》](#) [编号: css\_03]

涉及面试题:

1. 选择器的优先级是如何计算的?
2. 什么是 CSS 继承? 哪些属性能继承, 哪些不能?

- [《CSS 值和单位》](#) [编号: css\_04]

涉及面试题:

1. 你有没有使用过视网膜分辨率的图形? 当中使用什么技术?
2. px, em, rem, vw 有什么区别?

- [《CSS 给文本加样式: ① 字体属性》](#) [编号: css\_05]

涉及面试题:

简述字体图标的原理, 动手实现使用 iconfont 实现字体图标的 demo。

- [《CSS 给文本加样式: ② 文本属性》](#)
- [《CSS 基本视觉格式化: ① “块盒子”格式化》](#) [编号: css\_07]

涉及面试题:

1. 块级元素和行内元素分别有哪些? 空 (void) 元素有那些? 块级元素和行内元素有什么区别?
2. IE 盒模型和 W3C 盒模型有什么区别?
3. 在什么场景下会出现外边距合并? 如何合并? 如何不让相邻元素外边距合并? 给个父子外边距合并 demo。

- [《CSS 基本视觉格式化: ② “行内盒子”格式化》](#) [编号: css\_08]

涉及面试题:

1. line-height: 2; 和 line-height: 200%; 有什么区别?
2. 在什么场景下会出现外边距合并? 如何合并? 如何不让相邻元素外边距合并? 给个父子外边距合并 demo。
3. 行内元素的“边框”、“边界”等“框属性”是由 font-size 还是 line-height 控制?

4. `height=line-height` 可以用来垂直居中单行文本？代码怎么实现？
5. `inline-block` 有什么特性？
6. `inline-block` 在实际工作中有什么作用？
7. 怎么去除两个按钮中间的缝隙问题？
8. 一个页面有一排高度不一样的产品图，这时如果我们用 `inline-block`，怎样使他们“顶端”

- [《CSS 给盒子、背景、链接、列表、表单、表格等加样式》](#) [编号：css\_09]

涉及面试题：

1. 让一个元素“看不见”有几种方式？有什么区别？
2. 单行文本溢出加 ... 如何实现？
3. 如何在页面上实现一个圆形的可点击区域？

- [《让“盒子”动起来：① 浮动》](#) [编号：css\_10]

涉及面试题：

1. 浮动元素有什么特征？对父容器、其他浮动元素、普通元素、文字分别有什么影响？
2. 清除浮动指什么？如何清除浮动？两种以上方法。

- [《让“盒子”动起来：② “定位”和 BFC》](#) [编号：css\_11]

涉及面试题：

1. 有几种定位方式？分别是如何实现定位的？参考点是什么？使用场景是什么？
2. `z-index` 有什么作用？如何使用？
3. BFC 是什么？如何生成 BFC？BFC 有什么作用？举例说明。
4. 在什么场景下会出现外边距合并？如何合并？如何不让相邻元素外边距合并？给个父子外边距

- [《把“可以动的盒子”更优雅地展示：① “伪元素”妙用》](#) [编号：css\_12]

涉及面试题：

1. 如何使用伪元素来清除浮动？
2. 可以通过哪些方法优化 CSS3 Animation 渲染？

- [《把“可以动的盒子”更优雅地展示：② “居中”盒子》](#) [编号：css\_13]

涉及面试题：

1. 如何让块级元素水平居中？如何让行内元素水平居中？如何让 `inline-block` 元素水平居中
2. 垂直上下居中的办法？

- [《把“可以动的盒子”更优雅地展示：② 常用的“布局”》](#) [编号：css\_14]

涉及面试题：

响应式布局原理？

- [《让这些“展示”有更好的扩展性——媒体查询》](#) [编号：css\_15]

涉及面试题：

1. 列举你了解的 HTML5、CSS3 新特性？
2. Canvas 和 SVG 有什么区别？

- [《CSS 拓展：① 浏览器兼容》](#) [编号：css\_16]

涉及面试题：

1. 渐进增强和优雅降级分别是什么意思？
2. 什么是 CSS Hack？在哪个网站查看标签（属性）的浏览器兼容情况？
3. IE6、7 的 Hack 写法是？
4. 尽可能多的列举浏览器兼容的处理范例？
5. CSS Reset 是什么？CSS 预编译器是什么？后编译器（PostCSS）是什么？
6. CSS Reset 和 Normalize.css 有什么区别？
7. 尽可能多的写出浏览器兼容性问题？
8. 如何让 Chrome 浏览器显示小于 12px 的文字？
9. CSS 预处理器的比较：Less、Sass？
10. 常见兼容性问题？

- [《CSS 拓展：② CSS 编码规范》](#) [编号：css\_17]

涉及面试题：

1. 列举 CSS 编码规范？
2. 编码规范的作用是什么？列举 5 条以上编码规范。

- [《CSS 拓展：③ Emmet 常用语法介绍》](#)
- [《CSS 拓展：④ flex 布局》](#)

- 《CSS 拓展：⑤ CSS3 Transition》
- 《CSS 拓展：⑥ CSS3 变形》
- 《CSS 拓展：⑦ CSS3 Animation》

## 🔪🚀 HTML+CSS 实战：PC 端“简书”静态首页开发（难度：☆☆）

- 《PC 端“简书”静态首页：① 结构+header》[编号：html-css\_01]

涉及面试题：

1. 什么是盒模型？
2. CSS 的属性 `box-sizing` 有什么值？分别有什么作用？

- 《PC 端“简书”静态首页：② 侧边栏+推荐》
- 《PC 端“简书”静态首页：③ 轮播+按钮组+内容区》

## 🔪 JavaScript 初识

- 《① JS 速览——进入 JS 的世界》[编号：js\_01]

涉及面试题：

1. 简单介绍 JavaScript 的发展历史。ES3、ES5、ES6 分别指什么？
2. 说几条写 JavaScript 的基本规范？
3. JavaScript 代码中的“`use strict`”是什么意思？
4. 说说严格模式的限制？

- 《② 运算符、运算符优先级》[编号：js\_02]

涉及面试题：

1. NaN 是什么？有什么特别之处？
2. `==` 与 `===` 有什么区别？
3. `console.log(1+"2")` 和 `console.log(1-"2")` 的打印结果？
4. 为什么 `console.log(0.2+0.1==0.3)` 输出 `false` ？
5. 请用三元运算符（问号冒号表达式）改写以下代码：

```
if(a > 10) {  
    b = a  
}else {  
    b = a - 2  
}
```

6. 以下代码输出的结果是？

```
var a = 1;  
a+++a;  
typeof a+2;
```

7. 以下代码输出什么？

```
var d = a = 3, b = 4  
console.log(d)
```

8. 以下代码输出什么？

```
var d = (a = 3, b = 4)  
console.log(d)
```

9. 以下代码输出结果是？为什么？

```
var a = 1, b = 2, c = 3;  
var val = typeof a + b || c > 0  
console.log(val)  
var d = 5;  
var data = d == 5 && console.log('bb')  
console.log(data)  
var data2 = d = 0 || console.log('haha')  
console.log(data2)  
var x = !! "Hello" + (! "world", !! "from here!!");  
console.log(x)
```

10. 以下代码输出结果是？为什么？

```
var a = 1;  
var b = 3;  
console.log( a+++b );
```

11. 以下代码输出的结果是？为什么？

```
console.log(1+1);  
console.log("2"+"4");  
console.log(2+"4");  
console.log(+ "4");
```

- 《③ 变量、值、数据类型、数据类型转换》[编号: js\_03]

涉及面试题:

1. JavaScript 定义了几种数据类型? 哪些是原始类型? 哪些是复杂类型? null 是对象吗?
2. 对象类型和原始类型的不同之处? 函数参数是对象会发生什么问题?
3. 怎样判断“值”属于哪种类型? typeof 是否能正确判断类型? instanceof 呢? instanceof 有什么作用? 内部逻辑是如何实现的?
4. null, undefined 的区别?
5. 说一下 JS 中类型转换的规则?
6. 以下代码的输出? 为什么?  

```
console.log(a);  
var a = 1;  
console.log(b);
```
7. 以下代码输出什么?  

```
var a = typeof 3+4  
console.log(a)
```
8. 以下代码输出什么?  

```
var a = typeof typeof 4+4  
console.log(a)
```

- 《④ 流程控制语句》[编号: js\_04]

涉及面试题:

1. break 与 continue 有什么区别?
2. switch...case 语句中的 break 有什么作用?
3. for...of、for...in 和 forEach、map 的区别?
4. 写出如下知识点的代码范例:
  - ① if...else 的用法;
  - ② switch...case 的用法;
  - ③ while 的用法;
  - ④ do...while 的用法;
  - ⑤ for 遍历数组的用法;
  - ⑥ for...in 遍历对象的用法;
  - ⑦ break 和 continue 的用法。



5. 以下代码输出什么？

```
var a = 2
if(a = 1) {
  console.log("a 等于 1")
}else {
  console.log("a 不等于 1")
}
```

## ▽ JavaScript 入门

- 《① JS 函数——养成函数思维》[编号：js\_05]

涉及面试题：

1. 写一个函数，返回参数的平方和？

```
function sumOfSquares() {
  // 补全
}
var result = sumOfSquares(2, 3, 4)
var result2 = sumOfSquares(1, 3)
console.log(result) // 29
console.log(result2) // 10
```

2. 如下代码的输出？为什么？

```
sayName("world");
sayAge(10);
function sayName(name) {
  console.log("hello ", name);
}
var sayAge = function(age) {
  console.log(age);
};
```

3. 如下代码的输出？为什么？

```
var x = 10;
bar()
```

```
function bar() {  
  var x = 30;  
  function foo() {  
    console.log(x)  
  }  
  foo();  
}
```

4. 如下代码的输出? 为什么?

```
var x = 10  
bar()  
function foo() {  
  console.log(x)  
}  
function bar() {  
  var x = 30  
  foo()  
}
```

5. 如下代码的输出? 为什么?

```
var a = 1  
function fn1() {  
  function fn3() {  
    function fn2() {  
      console.log(a)  
    }  
    fn2()  
    var a = 4  
  }  
  var a = 2  
  return fn3  
}  
var fn = fn1()  
fn() // ?
```

6. 如下代码的输出? 为什么?

```
var a = 1
```

```
function fn1() {  
  function fn2() {  
    console.log(a)  
  }  
  function fn3() {  
    var a = 4  
    fn2()  
  }  
  var a = 2  
  return fn3  
}  
var fn = fn1()  
fn() // ?
```

7. 如下代码的输出? 为什么?

```
var a = 1  
function fn1() {  
  function fn3() {  
    var a = 4  
    fn2()  
  }  
  var a = 2  
  return fn3  
}  
function fn2() {  
  console.log(a)  
}  
var fn = fn1()  
fn() // ?
```

8. 如下代码的输出? 为什么?

```
var a = 1  
var c = {name: "oli", age: 2}  
function f1(n) {  
  ++n  
}  
function f2(obj) {
```

```
    ++obj.age
  }
  f1(a)
  f2(c)
  f1(c.age)
  console.log(a)
  console.log(c)
```

9. 如下代码的输出？为什么？

```
var obj1 = {a:1, b:2};
var obj2 = {a:1, b:2};
console.log(obj1 == obj2);
console.log(obj1 = obj2);
console.log(obj1 == obj2);
```

- 《② JS 数组——让数据排排坐》[编号：js\_06]

涉及面试题：

1. 写一个函数 `squireArr`，其参数是一个数组，作用是把数组中的每一项变为原值的平方。

```
var arr = [3, 4, 6]
function squireArr(arr) {
  // 补全
}
squireArr(arr)
console.log(arr) // [9, 16, 36]
```

2. 写一个函数 `squireArr`，其参数是一个数组，返回一个新的数组，新数组中的每一项是原数组对应值的平方，原数组不变。

```
var arr = [3, 4, 6]
function squireArr(arr) {
  // 补全
}
var arr2 = squireArr(arr)
console.log(arr) // [3, 4, 6]
console.log(arr2) // [9, 16, 36]
```

3. 遍历 `company` 对象，输出里面每一项的值。

```
var company = {  
  name: "qdywxs",  
  age: 3,  
  sex: "男"  
}
```

4. 遍历数组，打印数组里的每一项的平方。

```
var arr = [3, 4, 5]
```

- 《③ JS 对象——理解对象》[编号：js\_07]

涉及面试题：

1. 介绍 JS 有哪些内置对象？
2. 以下代码输出什么？

```
var name = "sex"  
var company = {  
  name: "qdywxs",  
  age: 3,  
  sex: "男"  
}  
console.log(company[name])
```

3. 以下代码输出什么？

```
var name = "sex"  
var company = {  
  name: "qdywxs",  
  age: 3,  
  sex: "男"  
}  
console.log(company.name)
```

- 《④ 了解 DOM——与网页交互》
- 《⑤ JS 事件——异步编码》

## ▽ JavaScript 基础

- 《JS 函数：① 把函数视为“值”》

- 《JS 函数：② 嵌套函数、作用域和闭包》[编号：js\_11]

涉及面试题：

1. 闭包是什么？闭包的作用是什么？闭包有哪些使用场景？
2. 使用递归完成 1 到 100 的累加？
3. 谈谈垃圾回收机制的方式及内存管理？
4. 谈谈你对 JS 执行上下文栈和作用域链的理解？
5. 如下代码输出多少？如果想输出 3，那如何改造代码？

```
var fnArr = [];  
for(var i=0; i<10; i++) {  
  fnArr[i] = function() {  
    return i  
  };  
}  
console.log(fnArr[3]())
```

6. 封装一个 Car 对象。

```
var Car = (function() {  
  var speed = 0;  
  // 补充  
  return {  
    setSpeed: setSpeed,  
    getSpeed: getSpeed,  
    speedUp: speedUp,  
    speedDown: speedDown  
  }  
})();  
Car.setSpeed(30)  
Car.getSpeed() // 30  
Car.speedUp()  
Car.getSpeed() // 31  
Car.speedDown()  
Car.getSpeed() // 30
```

7. 如下代码输出多少？如何连续输出 0, 1, 2, 3, 4？

```
for(var i=0; i<5; i++) {  
  setTimeout(function() {
```

```
    console.log("delay:" + i)
  }, 0)
}
```

8. 如下代码输出多少?

```
function makeCounter() {
  var count = 0
  return function() {
    return count++
  };
}
var counter = makeCounter()
var counter2 = makeCounter();
console.log(counter()) // 0
console.log(counter()) // 1
console.log(counter2()) // ?
console.log(counter2()) // ?
```

- 《JS 数组：① ES3 数组方法》[编号: js\_12]

涉及面试题:

1. 数组的哪些 API 会改变原数组?
2. 写一个函数，操作数组，返回一个新数组，新数组中只包含正数。

```
function filterPositive(arr) {
  // 补全
}
var arr = [3, -1, 2, true]
filterPositive(arr)
console.log(filterPositive(arr)) // [3, 2]
```

3. 补全代码，实现数组按姓名、年纪、任意字段排序。

```
var users = [
  {name: "John", age: 20, company: "Baidu"},
  {name: "Pete", age: 18, company: "Alibaba"},
  {name: "Ann", age: 19, company: "Tencent"}
]
```

```
users.sort(byField("age"))
users.sort(byField("company"))
```

4. 用 splice 函数分别实现 push、pop、shift、unshift 方法。

如：

```
function push(arr, value) {
  arr.splice(arr.length, 0, value)
  return arr.length
}

var arr = [3, 4, 5]
arr.push(10) // arr 变成 [3, 4, 5, 10], 返回 4。
```

- 《JS 数组：② ES5 数组方法》[编号：js\_13]

涉及面试题：

1. for...of、for...in 和 forEach、map 的区别？
2. 数组的哪些 API 会改变原数组？
3. 如何消除一个数组里面重复的元素？
4. 判断一个变量是否是数组，有哪些办法？
5. ["1", "2", "3"].map(parseInt) 答案是多少？
6. 取数组的最大值（ES5、ES6）？
7. 实现一个 reduce 函数，作用和原生的 reduce 类似下面的例子。

Ex：

```
var sum = reduce([1, 2, 3], function(memo, num) {return memo + num;}, 0);
```

8. 怎样用原生 JS 将一个多维数组拍平？

```
var array = [1, [2], [3, [4, [5]]]]
function flat(arr) {
  // 补全
}

console.log(flat(array)) // [1, 2, 3, 4, 5]
```

- 《面向对象编程：① 对象构造函数》[编号：js\_14]

涉及面试题：

1. new 的原理是什么？通过 new 的方式创建对象和通过字面量创建有什么区别？
2. Object.create 有什么作用？



3. 怎样判断“值”属于哪种数据类型？`typeof` 是否能正确判断类型？`instanceof` 呢？`instance`
4. JavaScript 有哪些方法定义对象？
5. 如下代码中？`new` 一个函数本质上做了什么？

```
function Modal(msg) {  
    this.msg = msg  
}  
  
var modal = new Modal()
```

- 《面向对象编程：② 使用原型》[编号：js\_15]

涉及面试题：

1. JS 原型是什么？如何理解原型链？
2. JS 如何实现继承？
3. 实现一个函数 `clone` 可以对 JavaScript 中的五种主要数据类型（Number、string、
4. 对 String 做扩展，实现如下方式获取字符串中频率最高的字符：

```
var str = "ahbbccdeddddfg";  
var ch = str.getMostOften();  
console.log(ch); // d, 因为 d 出现了 5 次
```

5. 有如下代码，代码中并未添加 `toString` 方法，这个方法是从哪里来的？画出原型链图进行解

```
function People() {  
}  
  
var p = new People()  
p.toString()
```

6. 有如下代码，解释 Person、prototype、\_\_proto\_\_、p、constructor 之间的关联

```
function Person(name) {  
    this.name = name;  
}  
  
Person.prototype.sayName = function() {  
    console.log("My name is :" + this.name);  
}  
  
var p = new Person("Oli")  
p.sayName();
```

7. 下面两种写法有什么区别？

```
// 方法一:
function People(name, sex) {
  this.name = name;
  this.sex = sex;
  this.printName = function() {
    console.log(this.name);
  }
}
var p1 = new People("Oli", 2)
// 方法二:
function Person(name, sex) {
  this.name = name;
  this.sex = sex;
}
Person.prototype.printName = function() {
  console.log(this.name);
}
var p1 = new Person("Aman", 2);
```

8. 补全代码, 实现继承:

```
function Person(name, sex){
  // 补全
};
Person.prototype.getName = function() {
  // 补全
};
function Male(name, sex, age) {
  // 补全
};
// 补全
Male.prototype.getAge = function() {
  // 补全
};
var catcher = new Male("Oli", "男", 2);
catcher.getName();
```

9. 如下代码中 call 的作用是什么?

```
function Person(name, sex) {
  this.name = name;
  this.sex = sex;
}
function Male(name, sex, age) {
  Person.call(this, name, sex); // 这里的 call 有什么作用?
  this.age = age;
}
```

- 《JS 提供的对象：① 作为对象的“字符串”》[编号：js\_16]

涉及面试题：

1. 多行字符串的声明有哪几种常见写法？
2. 以下代码输出什么？

```
var str = "C:\Users\Document\node\index.js"
console.log(str.length)
```

如何声明 str 让 console.log(str) 输出 C:\Users\Document\node\index.js ?

- 《JS 提供的对象：② 正则表达式》[编号：js\_17]

涉及面试题：

1. 写一个函数 isValidUsername(str)，判断用户输入的是不是合法的用户名（长度 6-20 只能包括字母、数字、下划线）？
2. 写一个函数 isPhoneNum(str)，判断用户输入的是不是手机号？
3. 写一个函数 isEmail(str)，判断用户输入的是不是邮箱？
4. 写一个函数 trim(str)，去除字符串两边的空白字符？
5. \d, \w, \s, [a-zA-Z0-9], \b, ., \*, +, ?, x{3}, ^, \$ 分别是什么？
6. 什么是贪婪模式和非贪婪模式？

- 《JS 提供的对象：③ Date》[编号：js\_18]

涉及面试题：

写一个函数，参数为时间对象毫秒数的字符串格式，返回值为字符串。假设参数为时间对象毫秒数根据 t 的时间分别返回如下字符串：

- 刚刚（t 距当前时间不到 1 分钟时间间隔）
- 3 分钟前（t 距当前时间大于等于 1 分钟，小于 1 小时）

- 8 小时前 (t 距离当前时间大于等于 1 小时, 小于 24 小时)
- 3 天前 (t 距离当前时间大于等于 24 小时, 小于 30 天)
- 2 个月前 (t 距离当前时间大于等于 30 天小于 12 个月)
- 8 年前 (t 距离当前时间大于等于 12 个月)

```
function friendlyDate(time) {  
    // 补充  
}  
  
var str = friendlyDate("1556286683394") // x 分钟前 (以当前时间为准)  
var str2 = friendlyDate("1555521999999") // x 天前 (以当前时间为准)
```

- 《JS 提供的对象: ④ Math》[编号: js\_19]

涉及面试题:

1. 写一个函数, 返回从 min 到 max 之间的随机整数, 包括 min 不包括 max ?
2. 写一个函数, 生成一个随机颜色字符串, 合法的颜色为 #000000 ~ #ffffff。

```
function getRandColor() {  
    // 补全  
}  
  
var color = getRandColor()  
console.log(color) // #3e2f1b
```

3. 写一个函数, 生成一个长度为 n 的随机字符串, 字符串字符的取值范围包括 0 到 9, a 到

```
function getRandStr(len) {  
    // 补全  
}  
  
var str = getRandStr(10); // 0a3iJiRZap
```

4. 写一个函数, 生成一个随机 IP 地址, 一个合法的 IP 地址为 0.0.0.0 ~ 255.255.255

```
function getRandIP() {  
    // 补全  
}  
  
var ip = getRandIP()  
console.log(ip) // 10.234.121.45
```

- 《JS 提供的对象: ⑤ JSON》[编号: js\_20]

涉及面试题：

1. JSON 格式的数据需要遵循什么规则？
2. 使用 JSON 对象实现一个简单的深拷贝函数（deepCopy）？
3. XML 和 JSON 的区别？
4. eval 是做什么的？
5. 深拷贝和浅拷贝的区别？如何实现？

- 《浏览器提供的对象：① BOM》[编号：js\_21]

涉及面试题：

1. URL 如何编码解码？为什么要编码？
2. iframe 有那些缺点？
3. 补全如下函数，判断用户的浏览器类型。

```
function isAndroid() {  
    // 补全  
}  
function isIphone() {  
    // 补全  
}  
function isIpad() {  
    // 补全  
}  
function isIOS() {  
    // 补全  
}
```

- 《浏览器提供的对象：② DOM》[编号：js\_22]

涉及面试题：

1. 什么是 Virtual DOM, 为何要用 Virtual DOM?
2. 怎么添加、移除、复制、创建、和查找节点？
3. offsetWidth/offsetHeight, clientWidth/clientHeight 与 scrollWidth/scrollHeight 的区别是什么？
4. attribute 和 property 的区别是什么？
5. 写一个函数，批量操作 CSS。

```
function css(node, styleObj){  
    // 补全
```

```
}
css(document.body, {
  "color": "red",
  "background-color": "#ccc"
})
```

6. 补全代码，要求：当鼠标放置在 li 元素上，会在 img-preview 里展示当前 li 元素的 data-img 对应的图片。

```
<ul class="ct">
  <li data-img="1.png">鼠标放置查看图片 1</li>
  <li data-img="2.png">鼠标放置查看图片 2</li>
  <li data-img="3.png">鼠标放置查看图片 3</li>
</ul>
<div class="img-preview"></div>
<script>
  // 补全
</script>
```

- 《浏览器提供的对象：Ⓢ 定时器》[编号：js\_23]

涉及面试题：

1. setTimeout、setInterval、requestAnimationFrame 各有什么特点？
2. 实现一个节流函数？
3. setTimeout 倒计时为什么会出现误差？
4. 简单解释单线程、任务队列的概念？
5. 简述同步和异步的区别？
6. JS 延迟加载的方式有哪些？
7. 函数防抖节流的原理？
8. defer 和 async ？
9. 下面这段代码输出结果是？ 为什么？

```
var flag = true;
setTimeout(function() {
  flag = false;
}, 0)
while(flag) {}
console.log(flag);
```

10. 下面这段代码输出结果是？为什么？

```
var a = 1;
setTimeout(function() {
  a = 2;
  console.log(a);
}, 0);
var a ;
console.log(a);
a = 3;
console.log(a);
```

- 《JS 事件：① 事件流和 DOM2 事件处理程序》[编号：js\_24]

涉及面试题：

1. DOM 事件模型是什么？
2. 解释 DOM2 事件传播机制？

- 《JS 事件：② 事件对象和事件代理》[编号：js\_25]

涉及面试题：

1. 解释以下概念：事件传播机制、阻止传播、取消默认事件、事件代理？
2. 写一个 Demo，演示事件传播的过程，演示阻止传播的效果？
3. JS 的事件委托是什么，原理是什么？

- 《JS 事件：③ 常见事件使用》[编号：js\_26]

涉及面试题：

window.onload 和 document.onDOMContentLoaded 有什么区别？

## 🚀 原生 JS 实战：小 DEMO 系列（难度：☆☆）

- 《Tab 切换》
- 《模态框》
- 《JS 动画》
- 《图片懒加载》
- 《瀑布流布局》

## ▼ ES6+

- 《ES6 速学：① let、const 和解构赋值》[编号：es6\_01]

涉及面试题：

1. var、let 及 const 区别？
2. 使用解构，实现两个变量的值的交换？
3. 解构赋值？
4. 函数默认参数？
5. JavaScript 中什么是变量提升？什么是暂时性死区？

- 《ES6 速学：② 字符串、数组、函数、对象》[编号：es6\_02]

涉及面试题：

1. 箭头函数？
2. 箭头函数与普通函数有什么区别？
3. 反引号 ` 标识？
4. 使用 ES6 改下面的模板？

```
let iam = "我是";
let name = "Oli";
let str = "大家好, "+iam+name+", 请多指教。";
```
5. 属性简写、方法简写？
6. for...of 循环？
7. 字符串新增方法？

- 《ES6 速学：③ 彻底弄懂各种情况下的 this 指向》[编号：es6\_03]

涉及面试题：

1. 如何改变函数内部的 this 指针的指向？
2. 如何判断 this？箭头函数的 this 是什么？
3. call、apply 以及 bind 函数内部实现是怎么样？

- 《ES6 速学：④ 类、继承和模块化》[编号：es6\_04]



涉及面试题：

1. import 和 export?
2. ES6 中的 class 了解吗？ES6 中的 class 和 ES5 的类有什么区别？
3. 知道 ECMAScript6 怎么写 class 么？为什么会出现 class 这种东西？
4. 原型如何实现继承？Class 如何实现继承？Class 本质是什么？

- 《ES6 速学：⑤ Promise 对象》[编号：es6\_05]

涉及面试题：

1. Promise 有几种状态？Promise 的特点是什么，分别有什么优缺点？
2. Promise 构造函数是同步还是异步执行？then 呢？Promise 如何实现 then 处理？
3. Promise 和 setTimeout 的区别？
4. 如何实现 Promise.all() ？
5. 如何实现 Promise.prototype.finally() ？
6. all() 的用法？
7. 说说你对 Promise 的了解？

## 🔪🚀 原生 JS 实战：造轮子系列（难度：☆☆☆☆）

- 《实现一个简单的左右滑动手势库》
- 《Web 轮播：① “面向过程”写法》
- 《Web 轮播：② “面向对象”写法》
- 《Web 轮播：③ 给轮播加动画》

## 🔪 前后端交互

- 《交互的规则、标准：HTTP——① 五层网络模型和 HTTP 协议的发展历史》[编号：interaction\_01]

涉及面试题：

1. HTTPS 是如何实现加密的？
2. HTTP 和 HTTPS 的区别？
3. 如何实现页面每次打开时清除本页缓存？

- 《交互的规则、标准：HTTP——② HTTP 三次握手、URI、URL 和 URN》
- 《交互的规则、标准：HTTP——③ HTTP 报文》
- 《响应请求的“服务器”：用 Node.js 搭建简单“Web 服务器”》
- 《发出请求的“客户端”：① 认识 HTTP 客户端》

- 《发出请求的“客户端”：② 浏览器异步发送“HTTP 数据请求”——初识 AJAX》[编号：interaction\_06]

涉及面试题：

1. HTTP 状态码知道哪些？
2. 301 和 302 的区别是什么？
3. AJAX 是什么？有什么作用？
4. HTTP 的几种请求方法和区别？
5. AJAX 原理？

- 《发出请求的“客户端”：③ 浏览器异步发送“HTTP 数据请求”——XMLHttpRequest 对象详解》[编号：interaction\_07]

涉及面试题：

1. 把 GET 和 POST 类型的 AJAX 的用法手写一遍？
2. 封装一个 AJAX 函数？

- 《发出请求的“客户端”：④ 浏览器异步发送“HTTP 数据请求”——用 AJAX 实现简单新闻列表切换》
- 《发出请求的“客户端”：⑤ 同源策略和跨域》[编号：interaction\_09]

涉及面试题：

1. 什么是同源策略？
2. 什么是跨域？列举跨域有几种实现形式？
3. JSONP 的原理是什么？
4. JSON 和 JSONP 的区别？

- 《发出请求的“客户端”：⑥ 浏览器存储——Cookie 和 Session》[编号：interaction\_10]

涉及面试题：

1. Cookie、Session、localStorage 分别是什么？
2. Cookie、sessionStorage 和 localStorage 的区别？
3. 如何实现同一个浏览器多个标签页之间的通信？
4. HTML5 的离线储存怎么使用，工作原理能不能解释一下？
5. 浏览器是怎么对 HTML5 的离线储存资源进行管理和加载的呢？
6. Web 开发中会话跟踪的方法有哪些？
7. 使用 localStorage 封装一个 Storage 对象，达到如下效果：  
Storage.set("name", "前端一万小时") // 设置 name 字段存储的值为“前端一万小  
Storage.set("age", 2, 30);

```
Storage.set("people", ["Oli", "Aman", "Dante"], 60)
Storage.get("name") // "前端一万小时"
Storage.get("age") /*
    如果不超过 30 秒，返回数字类型的 2；如果超过 30 秒，返回
    并且 localStorage 里清除 age 字段。
    */
Storage.get("people") // 如果不超过 60 秒，返回数组； 如果超过 60 秒，返回
```

- 《发起请求的“客户端”：② 简析浏览器缓存》

## ▽ 前端拓展

- 《NPM 与 Node.js 入门：① Node.js》
- 《NPM 与 Node.js 入门：② NPM 初识》
- 《NPM 与 Node.js 入门：③ NPM 详解》
- 《NPM 与 Node.js 入门：④ 开发天气命令行应用》
- 《前端工程化：① “前端工程化”初识》[编号：engineering\_01]

涉及面试题：

1. 模块化开发怎么做？
2. Webpack 如何实现打包的？

- 《前端工程化：② JS 模块加载方案和“组件化”初探》[编号：engineering\_02]

涉及面试题：

谈谈你对 AMD、CMD 的理解？