

Web前端笔试题

*注：编程题支持伪代码，但要写全逻辑与方法，着实不想写可以留着口述。

1、请阅读以下代码：

CSS：

```
#left{
    color: white !important;
}
#container #left{
    color: red;
}
#left {
    color: green !important;
}
.container #left{
    color: blue;
}
```

html：

```
<div class="container" id="container">
    <span id="left">left</span>
</div>
```

在html中，#left的最终color属性值为？

2、请阅读以下代码：

CSS：

```
*{
    padding: 0;
    margin: 0;
    box-sizing: content-box;
}
.a1{
    width: 100px;
    height: 100px;
    padding: 20px;
    margin: 20px;
    border: 1px solid #000;
}
.a2{
    box-sizing: border-box;
    width: 100px;
    height: 100px;
    padding: 20px;
    margin: 20px;
    border: 1px solid #000;
}
```

html：

```
<div class="container">
    <div class="a1"></div>
    <div class="a2"></div>
</div>
```

(1) a1、a2两个元素的实际宽高（不含margin）是多少？

(2) 父级元素container的实际宽高是多少？

3、请解释 :nth-of-type(n) 与 :nth-child(n) 的区别。

4、javascript 的 typeof 返回哪些基本数据类型？

5、下面这段代码想要循环延时输出结果0 1 2 3 4，请问输出结果是否正确，若不正确，请说明为什么，并写出输出正确结果的代码。

```
for (var i = 0; i < 5; ++i) {  
    setTimeout(function() {  
        console.log(i + ' ' )  
    }, 100 * i)  
}
```

6、请阅读以下代码（执行环境为浏览器）：

```
var name = 'html'  
var person = {  
    name: 'less',  
    sayName: function() {  
        console.log(`${name} -- `, `${this.name}`)  
    },  
    man: {  
        name: 'sass',  
        sayName: function() {  
            console.log(`${name} -- `, `${this.name}`)  
        }  
    },  
    woman: {  
        name: 'stylus',  
        sayName: () => {  
            console.log(`${name} -- `, `${this.name}`)  
            console.log(this)  
        }  
    }  
}  
person.sayName()  
person.man.sayName()  
person.woman.sayName()
```

(1) 最后三行分别会输出什么？

(2) person.woman.sayName.call(person)会输出什么？

7、简述纯html页面、Node.js应用、Vue单页应用、Vue ssr(服务器端渲染)在生产环境中如何部署。

8、以下代码用来访问redis：

```
var redis = require('redis')
var client = redis.createClient()
client.set('key', 'value', function(err, data) {
  if(err) {
    console.error(err.message)
    process.exit(1)
  }
  client.get('key', function(err, data) {
    if(err) {
      console.error(err.message)
      return
    }
    console.log(data)
    process.exit(0)
  })
})
```

(1)请用Promise的异步调用方式重写。

(2)在经历了多个异步回调后，如何拿到完整的堆栈信息(stack trace)？

9、在webpack中，如何实现打包多入口应用？

10、为了实现一个vue的filter，需要你写一个方法shortenNumber，其接收两个参数：一个单位集合的数组 arr 和一个代表进位基准的数字 base。该方法返回一个函数，返回的函数能按照给定的单位和进位将输入的参数进行格式化。

eg:
filter = shortenNumber(['B', 'KB', 'MB', 'GB'], 1024);
filter('32') == '32B';
filter('2100') == '2KB';
filter('5MB') == '5MB'; //遇到不合理输入时返回字符串
filter([1, 2, 3]) == '1,2,3';
filter('这题做起来基本就过了') == '这题做起来基本就过了';