

# 四川文产院 2018 级《WEB 前端实训》试卷

---

答卷说明:

1. 本试卷满分 100 分, 120 分钟完卷。
2. 本次考试为开卷考试。
3. 本试卷适用于 2018 级移动专业 1、2 班

## 一、选择题 (30 分, 共 15 题, 每题 2 分, 答案包含单选、多选)

---

1. JavaScript 中声明变量的关键字包括: \_\_\_\_\_。

- a) var
- b) let
- c) const
- d) declare

2. 如下属于 JavaScript 原始数据类型的有: \_\_\_\_\_。

- a) Symbol
- b) Function
- c) Object
- d) Undefined

3. 在什么时候会出现 undefined?\_\_\_\_\_

- a) 变量被声明了，但没有赋值时，就等于 undefined。
- b) 调用函数时，应该提供的参数没有提供，该参数等于 undefined。
- c) 对象没有赋值的属性，该属性的值为 undefined。
- d) 函数没有返回值时，默认返回 undefined。

4. 以下数组方法中，哪一个是为了删除数组中的第一个元素?\_\_\_\_\_

- a) pop()
- b) push()
- c) shift()
- d) unshift()

5. 定义函数的方式包括: \_\_\_\_\_

- a) 函数声明
- b) 函数表达式
- c) 箭头函数
- d) Function() 构造函数

6. 间接调用函数的方法包括: \_\_\_\_\_

- a) call()
- b) apply()
- c) bind()
- d) Function()

7. 下述有关闭包的叙述，正确的有: \_\_\_\_\_。

- a) 如果内层函数中引用了自由变量，那么这个内层函数就是一个闭包。

- b) 如果一个外层函数返回了一个内存函数，那么内存函数就是一个闭包。
- c) 事件处理程序中的回调函数就是闭包的一个例子。
- d) 函数式编程中的柯里化，就是闭包的一个应用。

8. 下面有关纯函数的叙述，正确的有：\_\_\_\_\_。

- a) 纯函数的返回值只依赖于参数的值。
- b) 纯函数没有副作用。
- c) 给出相同的参数，纯函数总是返回相同的结果。
- d) 纯函数可以没有参数。

9. 面向对象编程的基本特征包括：\_\_\_\_\_。

- a) 方法重写
- b) 继承
- c) 封装
- d) 多态

10. 在 JavaScript 中创建对象的方式包括：\_\_\_\_\_。

- a) ES6 类
- b) `Object.create()`
- c) 构造函数
- d) 对象字面量

11. 在 JavaScript 中，执行上下文的创建分为两个阶段，这两个阶段是：\_\_\_\_\_。

- a) 解析阶段
- b) 编译阶段

- c) 创建阶段
- d) 执行阶段

12. 变量声明的三种方式中，变量可以提升的方式包括：\_\_\_\_\_。

- a) var 声明变量
- b) const 声明变量
- c) let 声明变量

13. 在 JavaScript 中，有关 this 值的说法，正确的包括：\_\_\_\_\_。

- a) 全局作用域下，this 的值就是全局变量。
- b) 箭头函数提供 this 绑定。
- c) 在函数中，this 的值根据函数调用的形式来确定。
- d) 构造函数中 this 的值为用 new 运算符调用构造函数返回的对象。

14. 下面有关 ES6 类的说法，正确的包括：\_\_\_\_\_。

- a) ES6 中的类本质上是 ES5 构造函数的一种语法糖。
- b) 有了 ES6 的类，JavaScript 就变成了一门基于类的编程语言。
- c) 即使有了 ES6 的类，我们也无法在 JavaScript 中实现 OOP 的三大基本特征。
- d) ES6 的类可以实现私有字段。

15. 下面有关 ES6 类的静态方法，正确的包括：\_\_\_\_\_。

- a) 静态方法是直接绑定到类的函数，而不是绑定类的具体实例上的函数。
- b) 公有静态方法可以在类体外调用，私有静态方法不能在类体外调用。
- c) 静态方法不能访问实例字段。
- d) 私有静态方法不能访问私有静态字段。

---

## 二、代码填空题（20 分，共 5 题，每题 4 分）

---

1.有如下代码：

```
var lastName = 'global_name';
const func = function(firstName) {
  return firstName + ' ' + this.lastName;
};

var person = {
  lastName: 'person_name';
};

let a = func.call(undefined, 'student');
console.log(a);
```

上述代码运行时，控制台的输出为：  。

---

2.有如下代码：

```
let a = 1;
let b = 2;
(() => {
  const temp = a;
  a = b;
  b = temp;
```

```
})();  
console.log(b);
```

上述代码运行后，控制台的输出为：  。

---

3.有如下代码：

```
function outerFunc() {  
  let outerVar = 'I am outside!';  
  
  function innerFunc() {  
    console.log(outerVar);  
  }  
  return innerFunc;  
}  
  
const myInnerFunc = outerFunc();  
myInnerFunc();
```

上述代码运行后，控制台的输出为：  。

---

4.有如下代码：

```
function outerFunc() {  
  let outerVar = 'I am outside!';
```

```
function innerFunc() {  
  console.log(outerVar);  
}  
return innerFunc;  
}  
  
const myInnerFunc = outerFunc();  
myInnerFunc();
```

上述代码运行后，控制台的输出为：  。

---

5.有如下代码：

```
const studentName = '小明';  
  
const getStudentName = () => {  
  console.log('hi');  
};  
  
setTimeout(getStudentName, 3000);  
console.log(studentName);
```

上述代码运行后，控制台的输出为：

---

### 三、代码题（50 分，共 5 题，每题 10 分）

---

1.斐波那契（Fibonacci）数组的计算是一项很耗时的操作，请使用 Memoization 技术对斐波那契数组的计算进行优化。

*提示：用函数的自定义属性实现。代码必须包含控制台测试代码，并写出用 Memoization 技术之前和之后的本机求斐波那契数组（ $n = 30$ ）所用时间。*

2.有如下 HTML 文件：

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <title>动态添加节点与事件委托</title>
</head>
<body>
  <h2>课程列表</h2>
  <ul id='list'>
    <li>Java</li>
    <li>Python</li>
    <li>JavaScript</li>
  </ul>
  <input type="text" id="courseName" />
  <button id='btnAdd'>+</button>
  <button id='btnDel'>-</button>
</body>
</html>
```

该 HTML 页面在浏览器中显示如下界面：



# 课程列表

- Java
- Python
- JavaScript

请编写代码实现如下功能：

- 在文本框中输入课程名称，点击  按钮，把课程名称添加到课程列表中，然后清除文本框中的输入；
- 点击  按钮，从课程列表中删除最后一个课程。
- 当鼠标悬停在某门课程上时，让该课程的颜色变为红色；移开后恢复为黑色。

提示：

1. 请直接把 *JavaScript*、*CSS* 都写在 *HTML* 文件中。
2. 需要用到事件委托技术。

3.有如下数组：

```
const oArray = [1, 2, 2, 3, null, null, undefined, undefined, 4, 4];
```

请编写 JavaScript 代码，去掉数组中重复的元素值。

提示：使用 Set 实现。

---

4.有如下数组:

```
const oArray = [1, 2, 12, 10, 33, 7, 9, 22];
```

请编写 JavaScript 代码，对数组中的元素按照数字值从小到大进行排序。

提示：使用回调函数以及数组的 `sort()` 方法实现。

---

5.有如下数组:

```
const oArray = [0, 12, 7, 8, 9, 24, 1];
```

请编写代码，对该数组中的偶数值进行汇总求和。

提示：使用回调函数和数组的 `filter()`、`reduce()` 方法实现。