一：基础

1.变量提升

JavaScript 引擎的工作方式是，先解析代码，获取所有被声明的变量，然后再一行一行地运行。这造成的结果，就是所有的变量的声明语句，都会被提升到代码的头部，这就叫做变量提升

三：编程风格

1.全局变量

建议避免使用全局变量。如果不得不使用，可以考虑用大写字母表示变量名，这样更容易看出这是全局变量，比如UPPER_CASE。

四：console对象与控制台

- **Elements**：查看网页的 HTML 源码和 CSS 代码。
- **Resources**：查看网页加载的各种资源文件（比如代码文件、字体文件 CSS 文件等），以及在硬盘上创建的各种内容（比如本地缓存、Cookie、Local Storage等）。
- **Network**：查看网页的 HTTP 通信情况。
- **Sources**：查看网页加载的脚本源码。
- **Timeline**：查看各种网页行为随时间变化的情况。
- **Performance**：查看网页的性能情况，比如 CPU 和内存消耗。
- **Console**：用来运行 JavaScript 命令。

2.console对象的静态方法

2.1 console.log()　　console.info()　　console.debug()　用于在控制台输出信息。它可以接受一个或多个参数，将它们连接起来输出。

2.2 console.warn()：输出信息时，在最前面加一个黄色三角，表示警告。

console.error()：输出信息时，在最前面加一个红包的叉，表示出错。

2.3 console.table()：对于某些复合类型的数据，可以将其转为表格显示。

```
var languages = [
  { name: "JavaScript", fileExtension: ".js" },
  { name: "TypeScript", fileExtension: ".ts" },
  { name: "CoffeeScript", fileExtension: ".coffee" }
];

console.table(languages);
```

2.4 console.count() : 用于计数，输出它被调用了多少次

2.5 console.dir() : 用来对一个对象进行检查，并以易于阅读和打印的格式显示。

2.6 console.dirxml() : 主要用于以目录树的形式，显示DOM节点。

2.7 console.assert() : 主要用于程序运行过程中，进行条件判断，如果不满足条件，就显示一个错误，但不会中断程序执行。这样就相当于提示用户，内部状态不正确。

2.8 console.time(), console.timeEnd() : 用于计时，可以算出一个操作花费的准确时间。time方法表示计时开始，timeEnd方法表示计时结束。它们的参数是计时器的名称。调用timeEnd方法之后，控制台会显示“计时器名称: 所耗费的时间”。

2.9 console.group()，console.groupEnd()，console.groupCollapsed()

console.group和console.groupEnd这两个方法用于将显示的信息分组。它只在输出大量信息时有用，分在一组的信息，可以用鼠标折叠/展开。

console.groupCollapsed方法与console.group方法很类似，唯一的区别是该组的内容，在第一次显示时是收起的（collapsed），而不是展开的。

2.10 console.trace() : 显示当前执行的代码在堆栈中的调用路径

console.clear() : 用于清除当前控制台的所有输出，将光标回置到第一行。