**1 简述ECMAScript和JavaScript的关系**

**参考答案**

ECMAScript是欧洲计算机制造商协会(European Computer Manufacturers Association)通过ECMA-262标准化的脚本程序设计语言，它是一个Web行业标准。

JavaScript是美国网景公司(Netscape)在自己的Navigator浏览器中针对ECMAScript标准提供的一种实现；类似的实现还有美国微软公司(Microsoft)在自己的IE浏览器中实现的JScript脚本语言。

任何厂家都可以根据ECMAScript标准编写自己的浏览器脚本语言实现，在实现了ECMAScript所有的标准特性之外，还可以编写扩展自己的特殊功能——但这样的扩展功能往往就是造成浏览器兼容性问题的根源。

**2 可以在哪些地方编写JavaScript代码**

**参考答案**

可以在元素的事件处理方法中调用JavaScript代码；

可以<script>元素中调用JavaScript代码；

可以在外部.js文件中调用JavaScript代码，在HTML文件中引用此文件。

**3 描述下列JavaScript语句的作用**

1. document**.**write**(**'hello'**);**        //语句1
2. alert**(**'输入错误'**);**                //语句2
3. console**.**log**(**'hello world'**);**    //语句3

**参考答案**

语句1是在页面中输出一个字符串；

语句2 是弹出一个警告窗口；

语句3是在浏览器的控制台中输出一行提示消息。

**4 JavaScript 的数据类型分为哪些种类**

**参考答案**

JavaScript 的数据类型可以分为两类：原始类型和引用类型。

原始类型有 String、Number 、Boolean、undefined、null 五种。其中，字符串是使用一对单引号或者一对双引号括起来的任意文本；而数值类型都采用64 位浮点格式存储，不区分整数和小数；布尔（逻辑）只能有两个值：true 或 false。

Undefined表示一个变量声明过，但从未赋值。null 在程序中代表“无值”或者“无对象”，因此，可以通过将变量的值设置为 null 来清空变量。

引用类型指其他对象，如 Array、Date、Object 等。

**5 创建保存书籍相关数据需要的变量**

网上书城项目中，需要保存书籍对象所需要的相关数据(即书籍对象的属性)：如书名、简介、发行时间、页数、价格、是否特价等；创建必需的变量，保存这些数据。

**参考答案**

[代码](http://pdf7.tarena.com.cn/tts8_source/ttsPage/WEB/WEB_A_V02/JAVASCRIPTBASIC/DAY01/EXERCISE/01/index_answer.html)

### 1 录入学生数据

使用prompt( )函数获得用户输入，接收下列描述一个学生对象所需要的数据：姓名、性别、年龄、入学成绩、政治面貌、是否住校等；保存这些数据——注意数据的类型转换，然后再依次输出到控制台中。

**参考答案**

[代码](http://pdf7.tarena.com.cn/tts8_source/ttsPage/WEB/WEB_A_V02/JAVASCRIPTBASIC/DAY02/EXERCISE/01/index_answer.html)

### 2 总结JavaScript中的运算符有哪些类别

**参考答案**

算术运算符： + - \* / % ++ --

关系运算符： > < >= <= == === != !===

逻辑运算符： && || !

赋值运算符： = += -= \*= /= %=

字符连接运算符：+

条件运算符： ?:

### 3 说明下列代码的输出结果

1. **var** a **=** **10** **;**
2. **var** b **=** **10** **;**
3. console**.**log**(** a**++** **>** **++**b **);**
4. console**.**log**(**'a=' **+** a **);**
5. console**.**log**(**'b=' **+** b **);**

**参考答案**

运行结果为：

false

a=11;

b=11;

原因解释：

a++是“先取值再自加”，即整个表达式的值为a的原始值，a的值再自加；

++a是“先自加再取值”，即先执行自加，整个表达式的值为自加完成后的值。

### 4 下述哪些选项执行时不会出错

var userAge = 10;

A.console.log( userAge++ > 10 && userName=='tom' );

B.console.log( userAge++ >= 10 && userName=='tom' );

C.console.log( ++userAge < 10 || userName=='tom' );

D.console.log( ++userAge > 10 || userName=='tom' );

**参考答案**

这道题考察了三个知识点：

(1)i++和++i的区别

(2)变量未声明，直接使用会抛出ReferenceError

(3)&&和||运算符有“短路特性”，即只要第一个表达式能确定最终的结果，则不再执行第二个表达式的值

利用上述知识点可知，A中 userAge++的值为10,10>10值为false，而对于&&运算，只要第一个表达式值为false，则整个表达式的值为false，无需再执行后面的判定了；

D中，++userAge的值为11,11>10值为true，对于||运算，只要第一个表达式值为true，则整个表达式的值为true，无需再执行后面的判定了；

故，正确答案为AD。

### 5 成绩等级判定

由用户输入一个学生的语文成绩，若输入的值小于0或大于100，在控制台中输出“成绩无效”;若在80以上，输出“优秀";若在60以上，输出“及格”；否则输出不及格。

**参考答案**

实现此案例需要按照如下步骤进行。

**步骤一：创建页面**

首先创建一个纯文本文件，并修改名称为 exercise02\_05.html，然后在文本编辑工具中打开此文件，为其添加 HTML 代码，以创建一个标准结构的 HTML 文档（使用过渡型文档类型声明），并设置文档的标题以及编码格式。代码如下所示：

1. **<!**DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN" "http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd"**>**
2. **<**html xmlns**=**"http://www.w3.org/1999/xhtml"**>**
3. **<**head**>**
4. **<**meta http**-**equiv**=**"Content-Type" content**=**"text/html; charset=utf-8" **/>**
5. **<**title**>**成绩判定**</**title**>**
6. **</**head**>**
7. **<**body**>**
8. **</**body**>**
9. **</**html**>**

**步骤二：在页面主体中添加JavaScript代码块声明**

1. **<!**DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN" "http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd"**>**
2. **<**html xmlns**=**"http://www.w3.org/1999/xhtml"**>**
3. **<**head**>**
4. **<**meta http**-**equiv**=**"Content-Type" content**=**"text/html; charset=utf-8" **/>**
5. **<**title**>**成绩判定**</**title**>**
6. **</**head**>**
7. **<**body**>**
8. **<script type=**"text/javascript"**>**
10. **</script>**
11. **</**body**>**
12. **</**html**>**

**步骤三：编写嵌入式JavaScript代码**

读取用户输入，对输入进行解析，使用三目运算符对成绩等级进行判定。主要三目运算符中小括号的使用技巧。

1. **<!**DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN" "http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd"**>**
2. **<**html xmlns**=**"http://www.w3.org/1999/xhtml"**>**
3. **<**head**>**
4. **<**meta http**-**equiv**=**"Content-Type" content**=**"text/html; charset=utf-8" **/>**
5. **<**title**>**成绩判定**</**title**>**
6. **</**head**>**
7. **<**body**>**
8. **<**script type**=**"text/javascript"**>**
9. **var input =** prompt**(**'请输入学生语文成绩'**);**
10. **var score = parseFloat(input);**
11. **var msg = (score > 100 || score < 0 || isNaN(score) ) ?** '成绩无效' **:**
12. **(**score **>=** **80** **?** '优秀'**:**
13. **(**score **>=** **60** **?** '及格' **:** '不及格'**));**
14. console**.**log**(**'该成绩所在等级:'**+**msg**);**
15. **</**script**>**
16. **</**body**>**
17. **</**html**>**

**1 创建数字转换函数**

编写一个函数，接收一个一位的整形阿拉伯数字，返回该数字的汉字形式，如下所示：

1——壹

2——贰

3——叁

4——肆

5——伍

6——陆

7——柒

8——捌

9——玖

0——零

**参考答案**

实现此案例需要按照如下步骤进行。

**步骤一：创建页面**

首先创建一个纯文本文件，并修改名称为 exercise03\_01.html，然后在文本编辑工具中打开此文件，为其添加 HTML 代码，以创建一个标准结构的 HTML 文档（使用过渡型文档类型声明），并设置文档的标题以及编码格式。代码如下所示：

1. **<!**DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN" "http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd"**>**
2. **<**html xmlns**=**"http://www.w3.org/1999/xhtml"**>**
3. **<**head**>**
4. **<**meta http**-**equiv**=**"Content-Type" content**=**"text/html; charset=utf-8" **/>**
5. **<**title**>**数字转换函数**</**title**>**
6. **</**head**>**
7. **<**body**>**
8. **</**body**>**
9. **</**html**>**

**步骤二：在head标签中添加外部JS文件声明**

1. **<!**DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN" "http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd"**>**
2. **<**html xmlns**=**"http://www.w3.org/1999/xhtml"**>**
3. **<**head**>**
4. **<**meta http**-**equiv**=**"Content-Type" content**=**"text/html; charset=utf-8" **/>**
5. **<**title**>**数字转换函数**</**title**>**
6. **<script type=**"text/javascript" **src=**"exercise03\_01.js"**></script>**
7. **</**head**>**
8. **<**body**>**
9. **</**body**>**
10. **</**html**>**

**步骤三：创建exercise03\_01.js文件，在其中定义函数，接收一位整数，返回该整数对应的汉字表示**

使用三目运算符的嵌套对参数的值进行判断。

1. //把一位的阿拉伯数字转换为对应的汉字表示
2. **function** transform**(**num**){**
3. **return** num**==0?**'零'**:(**
4. num**==1?**'壹'**:(**
5. num**==2?**'贰'**:(**
6. num**==3?**'叁'**:(**
7. num**==4?**'肆'**:(**
8. num**==5?**'伍'**:(**
9. num**==6?**'陆'**:(**
10. num**==7?**'柒'**:(**
11. num**==8?**'捌'**:(**
12. num**==9?**'玖'**:**'无法转换'
13. **)**
14. **)**
15. **)**
16. **)**
17. **)**
18. **)**
19. **)**
20. **)**
21. **);**
22. **}**

**步骤四：在exercise03\_01.html中添加测试按钮，单击事件处理方法**

1. **<!**DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN" "http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd"**>**
2. **<**html xmlns**=**"http://www.w3.org/1999/xhtml"**>**
3. **<**head**>**
4. **<**meta http**-**equiv**=**"Content-Type" content**=**"text/html; charset=utf-8" **/>**
5. **<**title**>**数字转换函数**</**title**>**
6. **<**script type**=**"text/javascript" src**=**"exercise03\_01.js"**>**    **</**script**>**
7. **</**head**>**
8. **<**body**>**
9. **<button onclick=**"test()"**>数字转换</button>**
10. **</**body**>**
11. **</**html**>**

**步骤五：在exercise03\_01.js中定义全局变量，及测试方法**

全局变量充当计数器，记录着按钮被点击的次数。

1. //用于计数的全局变量
2. **var** num **=** **0;**
3. //测试方法
4. **function** test**(){**
5. **var** result **=** transform**(** num**++** **);**
6. console**.**log**(**result**);**
7. **}**

exercise03\_01.html文件中的完整代码如下所示：

1. **<!**DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN" "http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd"**>**
2. **<**html xmlns**=**"http://www.w3.org/1999/xhtml"**>**
3. **<**head**>**
4. **<**meta http**-**equiv**=**"Content-Type" content**=**"text/html; charset=utf-8" **/>**
5. **<**title**>**数字转换函数**</**title**>**
6. **<**script type**=**"text/javascript" src**=**"exercise03\_01.js"**></**script**>**
7. **</**head**>**
8. **<**body**>**
9. **<**button onclick**=**"test()"**>**数字转换**</**button**>**
10. **</**body**>**
11. **</**html**>**

exercise03\_01.js文件中的完整代码如下所示：

1. //把一位的阿拉伯数字转换为对应的汉字表示
2. **function** transform**(**num**){**
3. **return** num**==0?**'零'**:(**
4. num**==1?**'壹'**:(**
5. num**==2?**'贰'**:(**
6. num**==3?**'叁'**:(**
7. num**==4?**'肆'**:(**
8. num**==5?**'伍'**:(**
9. num**==6?**'陆'**:(**
10. num**==7?**'柒'**:(**
11. num**==8?**'捌'**:(**
12. num**==9?**'玖'**:**'无法转换'
13. **)**
14. **)**
15. **)**
16. **)**
17. **)**
18. **)**
19. **)**
20. **)**
21. **);**
22. **}**
23. //用于计数的全局变量
24. **var** num **=** **0;**
25. //测试方法
26. **function** test**(){**
27. **var** result **=** transform**(** num**++** **);**
28. console**.**log**(**result**);**
29. **}**

**2 使用 if 结构实现“个人所得税计算器”**

编写“个人所得税计算器”函数，该函数接收一个参数表示税前的薪资，方法最终返回应缴纳的个人所得税金额。假设个税计算梯度如下：

* 3500元以下部分免征
* 3500~5000的部分缴纳3%的个税
* 5000~9000的部分缴纳10%的个税
* 9000以上的部分缴纳20%的个税

**参考答案**

实现此案例需要按照如下步骤进行。

**步骤一：创建页面**

首先创建一个纯文本文件，并修改名称为 exercise03\_02.html，然后在文本编辑工具中打开此文件，为其添加 HTML 代码，以创建一个标准结构的 HTML 文档，并设置文档的标题以及编码格式。代码如下所示：

1. **<!**DOCTYPE HTML**>**
2. **<**html**>**
3. **<**head**>**
4. **<**title**>if</**title**>**
5. **<**meta http**-**equiv**=**"content-type" content**=**"text/html;charset=utf-8" **/>**
6. **</**head**>**
7. **<**body**>**
8. **</**body**>**
9. **</**html**>**

**步骤二：创建测试按钮，绑定单击事件处理方法**

为页面添加按钮，并设置其单击事件，然后在页面的 <head> 元素中添加 <script> 代码块。代码如下所示：

1. **<!**DOCTYPE HTML**>**
2. **<**html**>**
3. **<**head**>**
4. **<**title**>if</**title**>**
5. **<**meta http**-**equiv**=**"content-type" content**=**"text/html;charset=utf-8" **/>**
6. **<script type=**"text/javascript"**>**
7. **</script>**
8. **</**head**>**
9. **<**body**>**
10. **<button onclick=**"console.log(calculatePersonalTax(9500))"**>**
11. **计算个人所得税</button>**
12. **</**body**>**
13. **</**html**>**

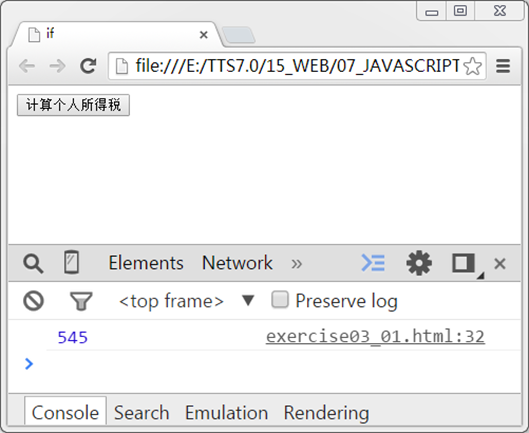
**步骤三：定义函数**

在 <script> 块中添加函数 calculatePersonalTax，并添加代码以实现个税的计算。代码如下所示：

1. **<!**DOCTYPE HTML**>**
2. **<**html**>**
3. **<**head**>**
4. **<**title**>if</**title**>**
5. **<**meta http**-**equiv**=**"content-type" content**=**"text/html;charset=utf-8" **/>**
6. **<**script type**=**"text/javascript"**>**
7. /\*\*
8. \*1、编写一个“个人所得税计算器”函数，接收一个参数表示一个工资值，方法最终返回该工资值应缴纳的个人所得税金额。假设个税计算梯度如下：
9. \* 3500元以下部分免征
10. \* 3500~5000的部分缴纳3%的个税
11. \* 5000~9000的部分缴纳10%的个税
12. \* 9000以上的部分缴纳20%的个税
13. \*/
14. **function** calculatePersonalTax**(salary) {**
15. **var tax = 0;**
16. **if (salary > 9000) {**
17. //9000以上的部分缴纳20%的个税
18. **tax += (salary - 9000) \* 0.2;**
19. **salary = 9000;**
20. **}**
21. **if (salary > 5000) {**
22. //5000~9000的部分缴纳10%的个税
23. **tax += (salary - 5000) \* 0.1;**
24. **salary = 5000;**
25. **}**
26. **if (salary > 3500) {**
27. //3500~5000的部分缴纳3%的个税
28. **tax += (salary - 3500) \* 0.03;**
29. **}**
30. **return tax;**
31. **}**
32. **</**script**>**
33. **</**head**>**
34. **<**body**>**
35. **<**button onclick**=**"console.log(calculatePersonalTax(9500))"**>**
36. 计算个人所得税**</**button**>**
37. **</**body**>**
38. **</**html**>**

**步骤四：测试**

在浏览器中查看 HTML 文件，并查看控制台的输出。效果如图－1所示，即，如果税前薪资为9500，则需要交纳个税金额为 545。



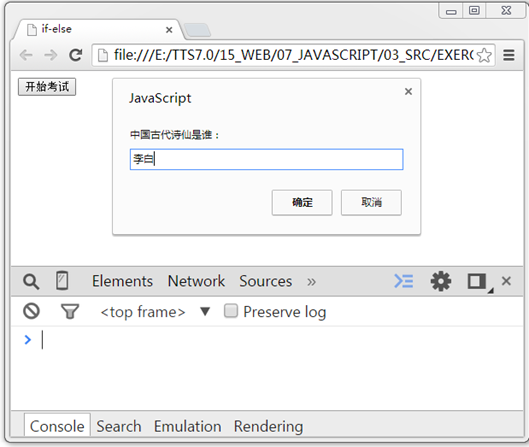
图－1

exercise03\_02.html 文件中的完整代码如下所示：

[代码](http://pdf7.tarena.com.cn/tts8_source/ttsPage/WEB/WEB_A_V02/JAVASCRIPTBASIC/DAY03/EXERCISE/01/index_answer.html)

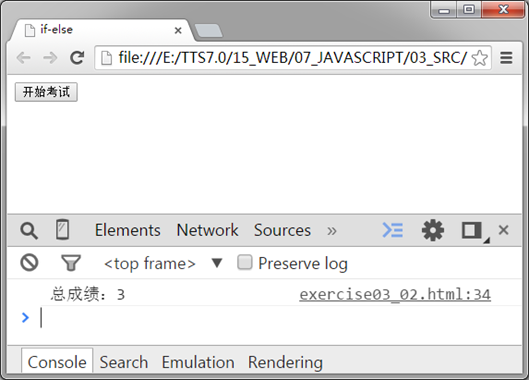
**3 使用 if-else 结构实现“在线考试”**

用户单击页面上的“开始考试”按钮，则会使用弹出框显示试题，并由用户作答。效果如图－2所示。



图－2

如图－2所示，将在弹出框中提出问题，并由用户录入答案。三道试题全部完成之后，将在控制台输出最终得分。效果如图－3所示。



图－3

**参考答案**

实现此案例需要按照如下步骤进行。

**步骤一：创建页面**

首先创建一个纯文本文件，并修改名称为 exercise03\_03.html，然后在文本编辑工具中打开此文件，为其添加 HTML 代码，以创建一个标准结构的 HTML 文档，并设置文档的标题以及编码格式。代码如下所示：

1. **<!**DOCTYPE HTML**>**
2. **<**html**>**
3. **<**head**>**
4. **<**title**>if-else</**title**>**
5. **<**meta http**-**equiv**=**"content-type" content**=**"text/html;charset=utf-8" **/>**
6. **</**head**>**
7. **<**body**>**
8. **</**body**>**
9. **</**html**>**

**步骤二：创建测试按钮，绑定单击事件处理方法**

为页面添加按钮，并设置其单击事件，然后在页面的 <head> 元素中添加 <script> 代码块。代码如下所示：

1. **<!**DOCTYPE HTML**>**
2. **<**html**>**
3. **<**head**>**
4. **<**title**>if-else</**title**>**
5. **<**meta http**-**equiv**=**"content-type" content**=**"text/html;charset=utf-8" **/>**
6. **<script type=**"text/javascript"**>**
7. **</script>**
8. **</**head**>**
9. **<**body**>**
10. **<button onclick=**"onlineTest()"**>开始考试</button>**
11. **</**body**>**
12. **</**html**>**

**步骤三：定义函数**

在 <script> 块中添加函数 onLineTest，并添加代码以答题及分数的计算。代码如下所示：

1. **<!**DOCTYPE HTML**>**
2. **<**html**>**
3. **<**head**>**
4. **<**title**>if-else</**title**>**
5. **<**meta http**-**equiv**=**"content-type" content**=**"text/html;charset=utf-8" **/>**
6. **<**script type**=**"text/javascript"**>**
7. /\*\*
8. \*实现一个“在线考试”应用，连续向用户提出三个问题，回答正确+1分，回答错误-1分，最后显示出用户的总得分
9. \*/
10. **function** onlineTest**() {**
11. **var score = 0;** //总分
12. //问题1
13. **var answer1 =** prompt**(**'中国古代诗仙是谁：'**);**
14. **if (answer1 ===** '李白'**) {**
15. **score++;**
16. **} else {**
17. **score--;**
18. **}**
19. //问题2
20. **var answer2 =** prompt**(**'中国古代诗圣是谁：'**);**
21. **if (answer2 ===** '杜甫'**) {**
22. **score++;**
23. **} else {**
24. **score--;**
25. **}**
26. //问题3
27. **var answer3 =** prompt**(**'中国古代诗佛是谁：'**);**
28. **if (answer3 ===** '王维'**) {**
29. **score++;**
30. **} else {**
31. **score--;**
32. **}**
33. **console.**log**(**'总成绩：' **+ score);**
34. **}**
35. **</**script**>**
36. **</**head**>**
37. **<**body**>**
38. **<**button onclick**=**"onlineTest()"**>**开始考试**</**button**>**
39. **</**body**>**
40. **</**html**>**

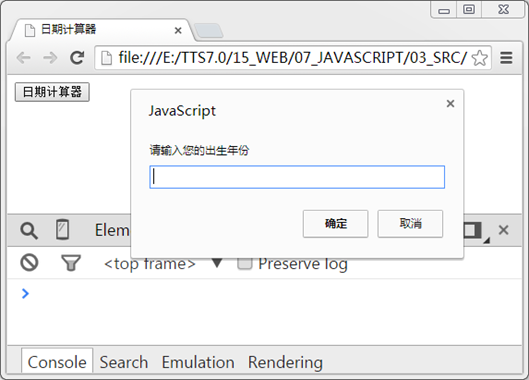
exercise03\_03.html 文件中的完整代码如下所示：

[代码](http://pdf7.tarena.com.cn/tts8_source/ttsPage/WEB/WEB_A_V02/JAVASCRIPTBASIC/DAY03/EXERCISE/01/index_answer.html)

**4 使用else if语句实现“日期计算器”**

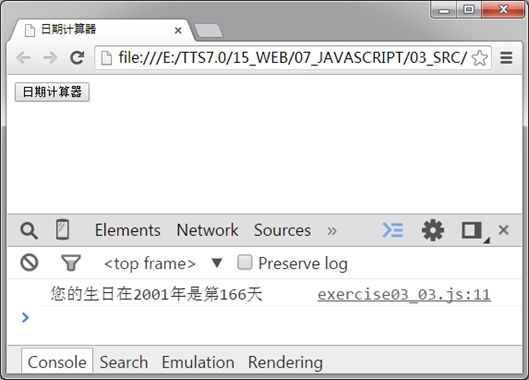
先后弹出三个输入提示框，让用户输入自己生日的年月日，创建函数，接收这三个参数，函数中使用if-else结构计算这一天是该年当中的第几天，并在控制台输出结果。

在浏览器中查看 HTML 文件，并单击页面上的按钮“日期计算器”，则要求用户录入出生的日期。页面效果如图－4所示。



图－4

用户在弹出的三个框中分别录入出生的年份、月份和日期后，将计算该出生日期在该年份中的天数。页面效果如图－5所示。



图－5

**参考答案**

实现此案例需要按照如下步骤进行。

**步骤一：创建页面**

首先创建一个纯文本文件，并修改名称为 exercise03\_04.html，然后在文本编辑工具中打开此文件，为其添加 HTML 代码，以创建一个标准结构的 HTML 文档，并设置文档的标题以及编码格式。代码如下所示：

1. **<!**DOCTYPE html**>**
2. **<**html**>**
3. **<**head**>**
4. **<**title**>**日期计算器**</**title**>**
5. **<**meta http**-**equiv**=**"content-type" content**=**"text/html;charset=utf-8" **/>**
6. **</**head**>**
7. **<**body**>**
8. **</**body**>**
9. **</**html**>**

**步骤二：创建测试按钮，绑定单击事件处理方法**

为页面添加测试按钮，为其定义单击事件，并为页面引入外部 js 文件。代码如下所示：

1. **<!**DOCTYPE html**>**
2. **<**html**>**
3. **<**head**>**
4. **<**meta http**-**equiv**=**"Content-Type" content**=**"text/html; charset=utf-8" **/>**
5. **<**title**>**日期计算器**</**title**>**
6. **<script type=**"text/javascript" **src=**"exercise03\_04.js"**></script>**
7. **</**head**>**
8. **<**body**>**
9. **<button onclick=**"testCalcDate()"**>日期计算器</button>**
10. **</**body**>**
11. **</**html**>**

**步骤三：创建外部js文件，编写testCalcDate()函数**

打开外部 js 文件，并在该文件中添加代码，编写函数以计算日期。代码如下所示：

1. //测试方法
2. **function** testCalcDate**(){**
3. **var** year **=** prompt**(**'请输入您的出生年份'**);**
4. year **=** parseInt**(**year**);**
5. **var** month **=** prompt**(**'请输入您的出生月份'**);**
6. month **=** parseInt**(**month**);**
7. **var** day **=** prompt**(**'请输入您的出生日期'**);**
8. day **=** parseInt**(**day**);**
10. **var** result **=** calcDate**(**year**,** month**,** day**);**
11. console**.**log**(**'您的生日在'**+**year**+**'年是第'**+**result**+**'天'**);**
12. **}**

**步骤四：编写calcDate()函数，用于执行计算操作**

由于闰年中的2月份有29天，而非闰年中的2月份有28天，故必须考虑闰年的问题。

1. //根据参数年月日的值，计算该日是当年中的第几天
2. **function** calcDate**(**year**,** month**,** day**){**
3. **var** counter **=** **0;**
4. //判定当年是否为闰年
5. **var** isLeapYear **=** **(**year**%4==0** **&&** year**%100!=0)** **||** **(**year**%400==0);**
6. **if(**month**==1){**
7. counter **=** day**;**
8. **}else** **if(**month**==2){**
9. counter **=** **31** **+** day**;**
10. **}else** **if(**month**==3){**
11. counter **=** **31** **+** **(**isLeapYear**?29:28)** **+** day**;**
12. **}else** **if(**month**==4){**
13. counter **=** **31** **+** **(**isLeapYear**?29:28)** **+** **31** **+** day**;**
14. **}else** **if(**month**==5){**
15. counter **=** **31** **+** **(**isLeapYear**?29:28)** **+** **31** **+** **30** **+** day**;**
16. **}else** **if(**month**==6){**
17. counter **=** **31** **+** **(**isLeapYear**?29:28)** **+** **31** **+** **30** **+** **31** **+** day**;**
18. **}else** **if(**month**==7){**
19. counter **=** **31** **+** **(**isLeapYear**?29:28)** **+** **30** **+** **30** **+** **31** **+** **30** **+** day**;**
20. **}else** **if(**month**==8){**
21. counter **=** **31** **+** **(**isLeapYear**?29:28)** **+** **30** **+** **30** **+** **31** **+** **30** **+** **31** **+** day**;**
22. **}else** **if(**month**==9){**
23. counter **=** **31** **+** **(**isLeapYear**?29:28)** **+** **30** **+** **30** **+** **31** **+** **30** **+** **31** **+** **31** **+** day**;**
24. **}else** **if(**month**==10){**
25. counter **=** **31** **+** **(**isLeapYear**?29:28)** **+** **30** **+** **30** **+** **31** **+** **30** **+** **31** **+** **31** **+** **30** **+** day**;**
26. **}else** **if(**month**==11){**
27. counter **=** **31** **+** **(**isLeapYear**?29:28)** **+** **30** **+** **30** **+** **31** **+** **30** **+** **31** **+** **31** **+** **30** **+** **31** **+** day**;**
28. **}else** **if(**month**==12){**
29. counter **=** **31** **+** **(**isLeapYear**?29:28)** **+** **30** **+** **30** **+** **31** **+** **30** **+** **31** **+** **31** **+** **30** **+** **31** **+** **30** **+** day**;**
30. **}**
31. **return** counter**;**
32. **}**

exercise03\_04.html 文件中的完整代码如下所示：

[代码](http://pdf7.tarena.com.cn/tts8_source/ttsPage/WEB/WEB_A_V02/JAVASCRIPTBASIC/DAY03/EXERCISE/01/index_answer.html)

exercise03\_04.js 文件中的完整代码如下所示：

[代码](http://pdf7.tarena.com.cn/tts8_source/ttsPage/WEB/WEB_A_V02/JAVASCRIPTBASIC/DAY03/EXERCISE/01/index_answer.html)

### 1 使用switch-case改写“日期计算器”

题目要求与之前的练习相同：先后弹出三个输入提示框，让用户输入自己生日的年月日，创建函数，接收这三个参数，函数中使用switch-case结构计算这一天是该年当中的第几天，并在控制台输出结果。

**参考答案**

其它代码请参考之前的练习，calcDate()函数使用switch-case改写为如下形式：

1. //根据参数年月日的值，计算该日是当年中的第几天
2. **function** calcDate**(**year**,** month**,** day**){**
3. **var** counter **=** **0;**
4. //判定当年是否为闰年
5. **var** isLeapYear **=** **(**year**%4==0** **&&** year**%100!=0)** **||** **(**year**%400==0);**
6. **switch(**year**){**
7. **case** **1:**
8. counter **=** day**;**
9. **break;**
10. **case** **2:**
11. counter **=** **31** **+** day**;**
12. **break;**
13. **case** **3:**
14. counter **=** **31** **+** **(**isLeapYear**?29:28)** **+** day**;**
15. **break;**
16. **case** **4:**
17. counter **=** **31** **+** **(**isLeapYear**?29:28)** **+** **31** **+** day**;**
18. **break;**
19. **case** **5:**
20. counter **=** **31** **+** **(**isLeapYear**?29:28)** **+** **31** **+** **30** **+** day**;**
21. **break;**
22. **case** **6:**
23. counter **=** **31** **+** **(**isLeapYear**?29:28)** **+** **31** **+** **30** **+** **31** **+** day**;**
24. **break;**
25. **case** **7:**
26. counter **=** **31** **+** **(**isLeapYear**?29:28)** **+** **30** **+** **30** **+** **31** **+** **30** **+** day**;**
27. **break;**
28. **case** **8:**
29. counter **=** **31** **+** **(**isLeapYear**?29:28)** **+** **30** **+** **30** **+** **31** **+** **30** **+** **31** **+** day**;**
30. **break;**
31. **case** **9:**
32. counter **=** **31** **+** **(**isLeapYear**?29:28)** **+** **30** **+** **30** **+** **31** **+** **30** **+** **31** **+** **31** **+** day**;**
33. **break;**
34. **case** **10:**
35. counter **=** **31** **+** **(**isLeapYear**?29:28)** **+** **30** **+** **30** **+** **31** **+** **30** **+** **31** **+** **31** **+** **30** **+** day**;**
36. **break;**
37. **case** **11:**
38. counter **=** **31** **+** **(**isLeapYear**?29:28)** **+** **30** **+** **30** **+** **31** **+** **30** **+** **31** **+** **31** **+** **30** **+** **31** **+** day**;**
39. **break;**
40. **case** **12:**
41. counter **=** **31** **+** **(**isLeapYear**?29:28)** **+** **30** **+** **30** **+** **31** **+** **30** **+** **31** **+** **31** **+** **30** **+** **31** **+** **30** **+** day**;**
42. **break;**
43. **}**
44. **return** counter**;**
45. **}**

### 2 请描述while语句的执行流程

while语句的语法格式如下：

1. **while(**表达式**)** **{**
2. 语句块
3. **}**

**参考答案**

while语句，首先判断表达式的值是否为 true，如果为 true，则执行大括号中的语句块；然后再判断表达式的值是否为 true，依次重复下去，直到表达式的值等于false，则不再执行大括号中的语句块，循环结束。

### 3 指出下列代码运行后的结果

1. **function** testFunc**()** **{**
2. **var** count**=10;**
3. **do**
4. **{**
5. console**.**log**(**count**+**" "**);**
6. count**++;**
7. **}while(**count**<5);**
8. **}**

**参考答案**

输出结果为：10。

从上面程序看，虽然count的值为10，表达式count<5返回false，但是do-while语句还是会执行一次循环体，输出10。

**1 简述for循环与while循环各自的使用场景**

**参考答案**

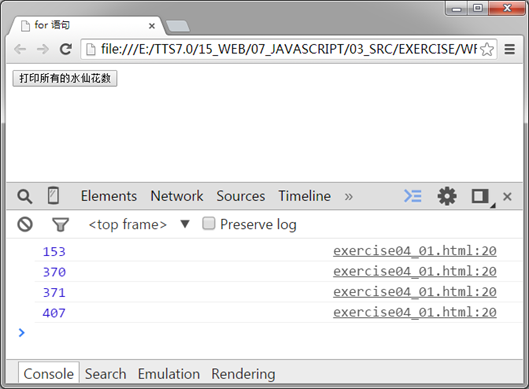
严格来讲，所有能够使用while循环的场景，也可以使用for循环代替；反之亦然。

但for循环更适合于循环起止条件明确已知的情形，如从x循环到y；而while循环更适合于循环的起止条件不确定的情形，如成绩录入应用，尤其适合于“死循环”情形。

**2 打印水仙花数**

水仙花数是这样的一个三位数，其个位、十位和百位上的数字的立方和等于这个数本身。比如，153 = 1\*1\*1 + 5\*5\*5 + 3\*3\*3，因此，153 就是一个水仙花数。

创建一个函数，在控制台输出所有的水仙花数，页面效果如图－1所示。



图－1

**参考答案**

实现此案例需要按照如下步骤进行。

**步骤一：创建页面**

首先创建一个纯文本文件，并修改名称为 exercise05\_01.html，然后在文本编辑工具中打开此文件，为其添加 HTML 代码，以创建一个标准结构的 HTML 文档，并设置文档的标题以及编码格式。代码如下所示：

1. **<!**DOCTYPE HTML**>**
2. **<**html**>**
3. **<**head**>**
4. **<**title**>for** 语句**</**title**>**
5. **<**meta http**-**equiv**=**"content-type" content**=**"text/html;charset=utf-8" **/>**
6. **</**head**>**
7. **<**body**>**
8. **</**body**>**
9. **</**html**>**

**步骤二：创建测试按钮，并添加script 代码块**

在页面上添加按钮，为其定义单击事件，并为页面添加 <script> 代码块。代码如下所示：

1. **<!**DOCTYPE HTML**>**
2. **<**html**>**
3. **<**head**>**
4. **<**title**>for** 语句**</**title**>**
5. **<**meta http**-**equiv**=**"content-type" content**=**"text/html;charset=utf-8" **/>**
6. **<script type=**"text/javascript"**>**
7. **</script>**
8. **</**head**>**
9. **<**body**>**
10. **<button onclick=**"printNarcissusNumbers()"**>打印所有的水仙花数</button>**
11. **</**body**>**
12. **</**html**>**

**步骤三：添加函数，打印水仙花数**

定义函数 printNarcissusNumbers，计算所有的水仙花数，并打印显示。

需要注意的是，因为本试题中，只要求打印水仙花数，即三位数，因此，循环时，只需要考虑100到1000之前的数值即可。

代码如下所示：

1. **<!**DOCTYPE HTML**>**
2. **<**html**>**
3. **<**head**>**
4. **<**title**>for** 语句**</**title**>**
5. **<**meta http**-**equiv**=**"content-type" content**=**"text/html;charset=utf-8" **/>**
6. **<**script type**=**"text/javascript"**>**
7. //打印出所有的水仙花数
8. **function** printNarcissusNumbers**() {**
9. **for (var i = 100; i < 1000; i++) {**
10. //获得百位上的数字
11. **var hundreds = parseInt(i / 100);**
12. //获得十位上的数字
13. **var tens = parseInt((i - hundreds \* 100) / 10);**
14. //获得个位上的数字
15. **var ones = i % 10;**
16. **if (i == hundreds \* hundreds \* hundreds + tens \* tens \* tens + ones \* ones \* ones) {**
17. **console.**log**(i);**
18. **}**
19. **}**
20. **}**
21. **</**script**>**
22. **</**head**>**
23. **<**body**>**
24. **<**button onclick**=**"printNarcissusNumbers()"**>**打印所有的水仙花数**</**button**>**
25. **</**body**>**
26. **</**html**>**

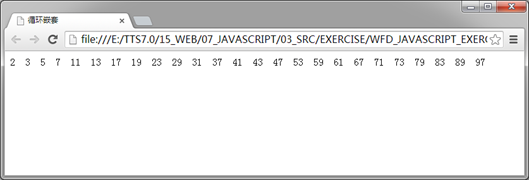
exercise05\_01.html文件中的完整代码如下所示：

[代码](http://pdf7.tarena.com.cn/tts8_source/ttsPage/WEB/WEB_A_V02/JAVASCRIPTBASIC/DAY05/EXERCISE/01/index_answer.html)

**3 打印100以内的素数**

素数，又称为质数（prime number）。素数是指，一个大于1的自然数，除了1和它本身外，不能被其他自然数（质数）整除。换句话说就是，该数除了1和它本身以外不再有其他的因数，则为素数。

要求编写函数，在页面输出显示 100 以内的所有素数。页面效果如图－2所示。



图－2

**参考答案**

此题需要使用嵌套的循环来实现。当判断某个数值是否为素数时，先用2去除这个数值，如果不能整除，则使用 3 继续去除，直到计算到该数值的一半为止。如果发生了整除，则循环中断，表示该数值不是素数。然后继续下一个数值进行判断。

实现此案例需要按照如下步骤进行。

**步骤一：创建页面**

首先创建一个纯文本文件，并修改名称为 exercise05\_02.html，然后在文本编辑工具中打开此文件，为其添加 HTML 代码，以创建一个标准结构的 HTML 文档，并设置文档的标题以及编码格式。代码如下所示：

1. **<!**DOCTYPE HTML**>**
2. **<**html**>**
3. **<**head**>**
4. **<**title**>**循环嵌套**</**title**>**
5. **<**meta http**-**equiv**=**"content-type" content**=**"text/html;charset=utf-8" **/>**
6. **</**head**>**
7. **<**body**>**
8. **</**body**>**
9. **</**html**>**

**步骤二：编写 <script> 代码块，并定义测试方法**

为页面添加 <script> 代码块，并定义测试方法。代码如下所示：

1. **<!**DOCTYPE HTML**>**
2. **<**html**>**
3. **<**head**>**
4. **<**title**>**循环嵌套**</**title**>**
5. **<**meta http**-**equiv**=**"content-type" content**=**"text/html;charset=utf-8" **/>**
6. **<script type=**"text/javascript"**>**
7. printPrimeNumbers**();**
8. **function** printPrimeNumbers**() {**
9. **}**
10. **</script>**
11. **</**head**>**
12. **<**body**>**
13. **</**body**>**
14. **</**html**>**

**步骤三：判断素数**

为函数 printPrimeNumbers 添加代码，找到100以内的所有素数，并在页面显示输出。代码如下所示：

1. **<!**DOCTYPE HTML**>**
2. **<**html**>**
3. **<**head**>**
4. **<**title**>**循环嵌套**</**title**>**
5. **<**meta http**-**equiv**=**"content-type" content**=**"text/html;charset=utf-8" **/>**
6. **<**script type**=**"text/javascript"**>**
7. printPrimeNumbers**();**
8. **function** printPrimeNumbers**()** **{**
9. **for (var i = 2; i < 100; i++) {**
10. **var j = 2;**
11. **for (; j <= i / 2; j++) {**
12. **if (i % j == 0) {**
13. **break;**
14. **}**
15. **}**
16. **if (j > i / 2) {**
17. **document.**write**(i +** "&nbsp;&nbsp;"**);**
18. **}**
19. **}**
20. **}**
21. **</**script**>**
22. **</**head**>**
23. **<**body**>**
24. **</**body**>**
25. **</**html**>**

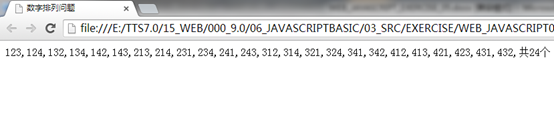
exercise05\_02.html文件中的完整代码如下所示：

[代码](http://pdf7.tarena.com.cn/tts8_source/ttsPage/WEB/WEB_A_V02/JAVASCRIPTBASIC/DAY05/EXERCISE/01/index_answer.html)

**4 数字排列问题**

有 1 、 2 、 3 、 4 共4个数字，能组成多少个互不相同且无重复数字的三位数？都是多少？

要求编写函数，输出结果。页面效果如图－3所示。



图－3

**参考答案**

实现数字排列，需要使用每个数字逐一与其他数字进行组合，因此需要使用嵌套的循环来实现。

实现此案例需要按照如下步骤进行。

**步骤一：创建页面**

首先创建一个纯文本文件，并修改名称为 exercise05\_03.html，然后在文本编辑工具中打开此文件，为其添加 HTML 代码，以创建一个标准结构的 HTML 文档，并设置文档的标题以及编码格式。代码如下所示：

1. **<!**DOCTYPE HTML**>**
2. **<**html**>**
3. **<**head**>**
4. **<**title**>**数字排列问题**</**title**>**
5. **<**meta http**-**equiv**=**"content-type" content**=**"text/html;charset=utf-8" **/>**
6. **</**head**>**
7. **<**body**>**
8. **</**body**>**
9. **</**html**>**

**步骤二：编写 <script> 代码块，并定义测试方法**

为页面添加 <script> 代码块，并定义测试方法。代码如下所示：

1. **<!**DOCTYPE HTML**>**
2. **<**html**>**
3. **<**head**>**
4. **<**title**>**数字排列问题**</**title**>**
5. **<**meta http**-**equiv**=**"content-type" content**=**"text/html;charset=utf-8" **/>**
6. **<script type=**"text/javascript"**>**
7. listNumbers **();**
8. **function** listNumbers **() {**
9. **}**
10. **</script>**
11. **</**head**>**
12. **<**body**>**
13. **</**body**>**
14. **</**html**>**

**步骤三：循环找出数字排列**

为函数 listNumbers添加代码，循环输出数字排列，并在页面显示输出。

需要注意的是，为了输出为字符串，在输出时，需要对数字类型的变量与字符串相加，实现隐式转换。代码如下所示：

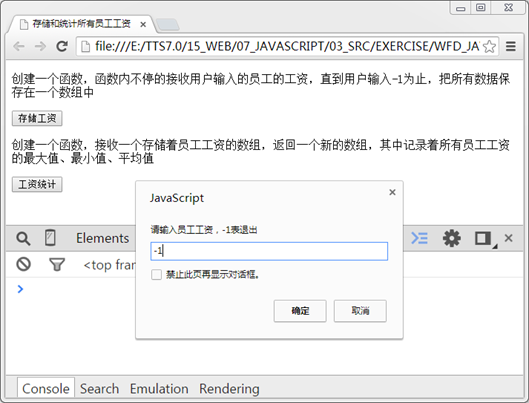
1. **<!**DOCTYPE HTML**>**
2. **<**html**>**
3. **<**head**>**
4. **<**title**>**数字排列问题**</**title**>**
5. **<**meta http**-**equiv**=**"content-type" content**=**"text/html;charset=utf-8" **/>**
6. **<**script type**=**"text/javascript"**>**
7. listNumbers **();**
8. **function** listNumbers **()** **{**
9. **var count=0;**
10. **for(var i=1;i<5;i++){**
11. **for(var j=1;j<5;j++){**
12. **if(i!=j){**
13. **for(var k=1;k<5;k++){**
14. **if(j!=k&&i!=k){**
15. **count++;**
16. **document.**write**(**''**+i+j+k+**','**);**
17. **}**
18. **}**
19. **}**
20. **}**
21. **}**
22. **document.**write**(**'共' **+ count +** '个'**);**
23. **}**
24. **</**script**>**
25. **</**head**>**
26. **<**body**>**
27. **</**body**>**
28. **</**html**>**

exercise05\_03.html文件中的完整代码如下所示：

[代码](http://pdf7.tarena.com.cn/tts8_source/ttsPage/WEB/WEB_A_V02/JAVASCRIPTBASIC/DAY05/EXERCISE/01/index_answer.html)

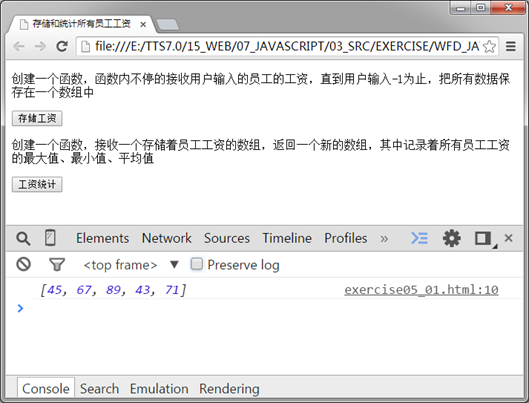
**1 存储和统计所有员工工资**

创建一个函数，不停的接收用户输入的员工工资，直到用户输入-1为止，把所有数据保存在一个数组中。单击页面上的“存储工资”按钮，页面效果如图－1所示。



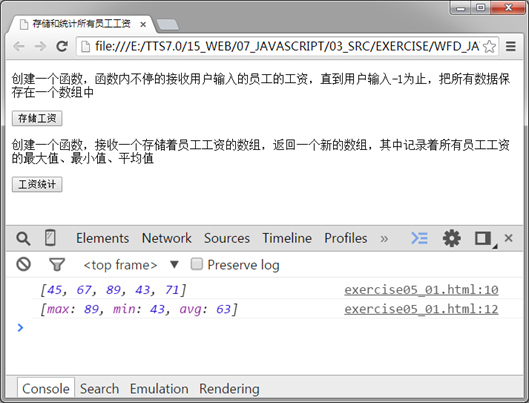
图－1

输入完毕后，页面显示所输入的数据内容，效果如图－2所示。



图－2

再创建一个函数，返回一个新的数组，其中记录着上一个函数所记录的所有员工工资的最大值、最小值、平均值，并在页面上显示输出。单击页面上的“工资统计”按钮，页面效果如图－3所示。



图－3

**参考答案**

由于要存储若干个员工的工资，则涉及到数组的使用。且该数组需要在两个函数间使用，故需要定义为全局变量。

实现此案例需要按照如下步骤进行。

**步骤一：创建页面**

首先创建一个纯文本文件，并修改名称为 exercise01\_01.html，然后在文本编辑工具中打开此文件，为其添加 HTML 代码，以创建一个标准结构的 HTML 文档，并设置文档的标题以及编码格式。代码如下所示：

1. **<!**DOCTYPE html**>**
2. **<**html**>**
3. **<**head**>**
4. **<**meta http**-**equiv**=**"content-Type" content**=**"text/html; charset=utf-8" **/>**
5. **<**title**>**存储和统计所有员工工资**</**title**>**
6. **</**head**>**
7. **<**body**>**
8. **</**body**>**
9. **</**html**>**

**步骤二：添加提示信息，创建保存工资的按钮，绑定单击事件处理方法**

为页面添加按钮，并定义单击事件。代码如下所示：

1. **<!**DOCTYPE html**>**
2. **<**html**>**
3. **<**head**>**
4. **<**meta http**-**equiv**=**"content-Type" content**=**"text/html; charset=utf-8" **/>**
5. **<**title**>**存储和统计所有员工工资**</**title**>**
6. **</**head**>**
7. **<**body**>**
8. **<p>创建一个函数，函数内不停的接收用户输入的员工的工资，直到用户输入-1为止，把所有数据保存在一个数组中</p>**
9. **<button onclick=**"console.log( save() )"**>存储工资</button>**
10. **<p>创建一个函数，接收一个存储着员工工资的数组，返回一个新的数组，其中记录着所有员工工资的最大值、最小值、平均值</p>**
11. **<button onclick=**"console.log( statistic() )"**>工资统计</button>**
12. **</**body**>**
13. **</**html**>**

**步骤三：添加外部js文件引用**

为页面添加外部 js 文件引用，代码如下所示：

1. **<!**DOCTYPE html**>**
2. **<**html**>**
3. **<**head**>**
4. **<**meta http**-**equiv**=**"content-Type" content**=**"text/html; charset=utf-8" **/>**
5. **<**title**>**存储和统计所有员工工资**</**title**>**
6. **<script type=**"text/javascript" **src=**"exercise01\_01.js"**></script>**
7. **</**head**>**
8. **<**body**>**
9. **<**p**>**创建一个函数，函数内不停的接收用户输入的员工的工资，直到用户输入**-1**为止，把所有数据保存在一个数组中**</**p**>**
10. **<**button onclick**=**"console.log( save() )"**>**存储工资**</**button**>**
11. **<**p**>**创建一个函数，接收一个存储着员工工资的数组，返回一个新的数组，其中记录着所有员工工资的最大值、最小值、平均值**</**p**>**
12. **<**button onclick**=**"console.log( statistic() )"**>**工资统计**</**button**>**
13. **</**body**>**
14. **</**html**>**

**步骤四：创建外部exercise01\_01.js文件，编写save()函数**

为 js 文件添加代码，把用于保存所有工资的数组声明全局变量，便于在两个函数间使用。由于不确定用户会输入多少个员工工资，故可以使用do-while循环进行录入。代码如下所示。

1. //全局变量, 存储所有工资的数组
2. **var** salaryArr **=** **[];**
3. //存储工资
4. **function** save**()** **{**
5. **do** **{**
6. **var** input **=** prompt**(**'请输入员工工资，-1表退出'**);**
7. input **=** parseFloat**(**input**);**
8. **if** **(!**isNaN**(**input**)** **&&** input **!=** **-1)** **{**
9. salaryArr**[**salaryArr**.**length**]=**input**;**
10. **}**
11. **}** **while** **(**input **!=** **-1);**
12. **return** salaryArr**;**
13. **}**

**步骤五：编写统计函数 statistic**

继续为 js 文件添加代码，定义统计用的函数。代码如下所示：

1. //统计工资信息
2. **function** statistic**()** **{**
3. //声明变量，保存工资的最大值、最小值、平均值和总和
4. **var** max **=** min **=** sum **=** salaryArr**[0];**
5. **var** avg **=** **0;**
6. //循环数组
7. **for** **(var** i **=** **1;** i **<** salaryArr**.**length**;**i**++** **)** **{**
8. **if** **(**salaryArr**[**i**]** **>** max**)** **{**
9. max **=** salaryArr**[**i**];**
10. **}**
11. **if** **(**salaryArr**[**i**]** **<** min**)** **{**
12. min **=** salaryArr**[**i**];**
13. **}**
14. sum **+=** salaryArr**[**i**];**
15. **}**
16. avg **=** sum **/** salaryArr**.**length**;**
17. **var** result **=** **[];**
18. //下标是字符串的数组
19. result**[**"max"**]** **=** max**;**
20. result**[**"min"**]** **=** min**;**
21. result**[**"avg"**]** **=** avg**;**
22. **return** result**;**
23. **}**

exercise01\_01.html文件中的完整代码如下所示：

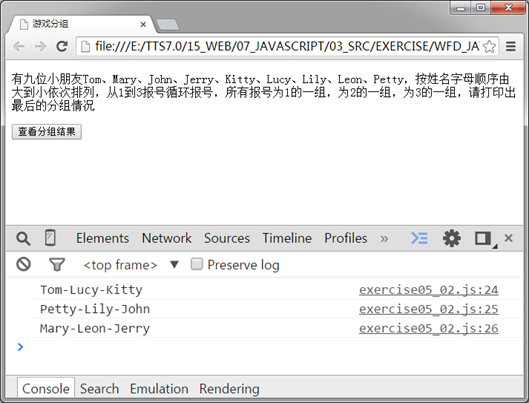
[代码](http://pdf7.tarena.com.cn/tts8_source/ttsPage/WEB/WEB_A_V02/JAVASCRIPTCORE/DAY01/EXERCISE/01/index_answer.html)

exercise01\_01.js文件中的完整代码如下所示：

[代码](http://pdf7.tarena.com.cn/tts8_source/ttsPage/WEB/WEB_A_V02/JAVASCRIPTCORE/DAY01/EXERCISE/01/index_answer.html)

**2 游戏分组**

有九位小朋友，名字分别为Tom、Mary、John、Jerry、Kitty、Lucy、Lily、Leon、Petty。现需要按姓名字母顺序由大到小依次排列，然后让小朋友们从1到3报号循环报号。报号完毕后，按照报号分组：所有报号为1的一组，为2的一组，为3的一组。打印出最后的分组情况，效果如图－4所示。



图－4

由图－4可以看出，Tom、Lucy、Kitty均为报号1 的小朋友，分为一组；Petty、Lily、John为报号 2 的小朋友，分为一组，最后一组为报号 3 的下朋友，即Mary、Leon和Jerry。

**参考答案**

由于涉及到多个字符串的排序和查找，应该把这些字符串保存到数组中；接下来对该数组中的数据进行排序和倒序；再遍历数组，把特定下标的元素保存到三个新的数组中。

实现此案例需要按照如下步骤进行。

**步骤一：创建页面**

首先创建一个纯文本文件，并修改名称为 exercise01\_02.html，然后在文本编辑工具中打开此文件，为其添加 HTML 代码，以创建一个标准结构的 HTML 文档，并设置文档的标题以及编码格式。代码如下所示：

1. **<!**DOCTYPE html**>**
2. **<**html**>**
3. **<**head**>**
4. **<**meta http**-**equiv**=**"Content-Type" content**=**"text/html; charset=utf-8" **/>**
5. **<**title**>**游戏分组**</**title**>**
6. **</**head**>**
7. **<**body**>**
8. **</**body**>**
9. **</**html**>**

**步骤二：添加按钮，绑定单击事件处理方法，并引入外部 js 文件**

为页面添加按钮，定义单击事件，并引入外部 js 文件。代码如下所示：

1. **<!**DOCTYPE html**>**
2. **<**html**>**
3. **<**head**>**
4. **<**meta http**-**equiv**=**"Content-Type" content**=**"text/html; charset=utf-8" **/>**
5. **<**title**>**游戏分组**</**title**>**
6. **<script type=**"text/javascript" **src=**"exercise01\_02.js"**></script>**
7. **</**head**>**
8. **<**body**>**
9. **<p>有九位小朋友Tom、Mary、John、Jerry、Kitty、Lucy、Lily、Leon、Petty，按姓名字母顺序由大到小依次排列，从1到3报号循环报号，所有报号为1的一组，为2的一组，为3的一组，请打印出最后的分组情况</p>**
10. **<button onclick=**"grouping()"**>查看分组结果</button>**
11. **</**body**>**
12. **</**html**>**

**步骤三：创建外部exercise01\_02.js文件，编写grouping()函数**

为外部 js 文件添加代码，编写 grouping 函数实现分组功能。代码如下所示：

1. //对姓名字符串进行排序，并分组
2. **function** grouping**()** **{**
3. **var** names **=** **[**'Tom'**,** 'Mary'**,** 'John'**,** 'Jerry'**,** 'Kitty'**,** 'Lucy'**,** 'Lily'**,** 'Leon'**,** 'Petty'**];**
4. //元素按字符大小(Unicode码)排序
5. names**.**sort**();**
6. //逆序排列
7. names**.**reverse**();**
8. //定义三个分组
9. **var** group1 **=** **[];**
10. **var** group2 **=** **[];**
11. **var** group3 **=** **[];**
12. //统计
13. **for** **(var** i **=** **0;** i **<** names**.**length**;** i**++)** **{**
14. **if** **((**i **+** **1)** **%** **3** **==** **1)** **{**
15. group1**[**group1**.**length**]=**names**[**i**];**
16. **}** **else** **if** **((**i **+** **1)** **%** **3** **==** **2)** **{**
17. group2**[**group2**.**length**]=**names**[**i**];**
18. **}** **else** **if** **((**i **+** **1)** **%** **3** **==** **0)** **{**
19. group3**[**group3**.**length**]=**names**[**i**];**
20. **}**
21. **}**
22. //输出显示
23. console**.**log**(**group1**.**join**(**'-'**));**
24. console**.**log**(**group2**.**join**(**'-'**));**
25. console**.**log**(**group3**.**join**(**'-'**));**
26. **}**

exercise01\_02.html中的完整代码如下所示：

[代码](http://pdf7.tarena.com.cn/tts8_source/ttsPage/WEB/WEB_A_V02/JAVASCRIPTCORE/DAY01/EXERCISE/01/index_answer.html)

exercise01\_02.js中的完整代码如下所示：

[代码](http://pdf7.tarena.com.cn/tts8_source/ttsPage/WEB/WEB_A_V02/JAVASCRIPTCORE/DAY01/EXERCISE/01/index_answer.html)

**3 插入排序（提高题，选做）**

排序，就是使一串记录，按照某种比较规则，递增或递减的排列起来的操作。常用的排序算法有很多，比如插入排序、冒泡排序等。

本案例要求使用插入排序算法实现对数组的排序。有一个长度为 12 的整型数组，使用插入排序算法将数组按照升序排列，并输出排序的过程以及结果。程序交互情况如图-5所示：

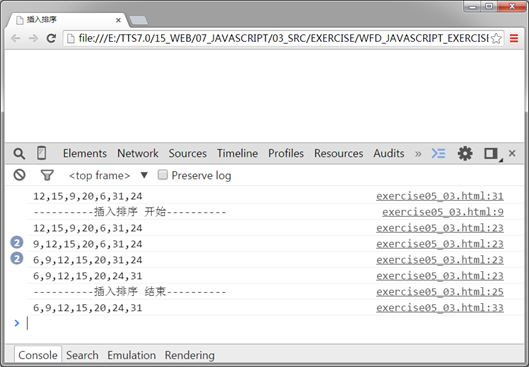


图- 5

**参考答案**

插入排序的基本操作就是将一个数据插入到已经排好序的有序数据中，从而得到一个新的、个数加一的有序数据。因此，在使用插入排序时，先假定将 n 个元素的数列分为已有序和无序两个部分，每次处理就是，将无序数列的第一个元素与有序数列的元素从后往前逐个进行比较，找出插入位置，将该元素插入到有序数列的合适位置中。插入排序的整个过程如图－6所示：

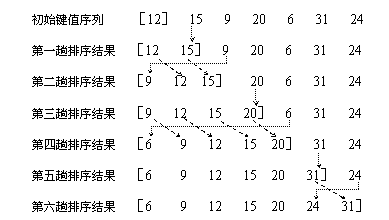


图- 6

图－6演示了插入排序算法的过程，其过程解释如下：

1、先将数组分为两部分：第一个元素分出来，作为有序数列（图－6中括号中的部分）；剩余的元素作为无序的数列；

2、第一次比较：取出无序数列中的第一个元素（数值15），和有序数列中的元素进行比较。比较时，在有序数列中从后向前进行逐一比较，如果该元素（数值15）大于有序数列中的某元素（数值12），则将该元素（数值15）放置在所比较元素的下一位置（数值12 之后）；

3、第一次比较之后，有序数列中有两个排序后的数值：12和15，剩余的是无序数列；

4、第二次比较：取出无序数列中的第一个元素（数值 9），和有序数列中的元素从后向前逐一比较。如果该元素（数值 9）小于有序数列中的某元素，则移动有序数列中的元素到下一个位置，直到在有序列表中找到比它小的元素，则插入此位置；

5、重复比较下去，直到末尾。

实现此案例需要按照如下步骤进行。

**步骤一：创建页面**

首先创建一个纯文本文件，并修改名称为 exercise01\_03.html，然后在文本编辑工具中打开此文件，为其添加 HTML 代码，以创建一个标准结构的 HTML 文档，并设置文档的标题以及编码格式。代码如下所示：

1. **<!**DOCTYPE html**>**
2. **<**html**>**
3. **<**head**>**
4. **<**meta http**-**equiv**=**"Content-Type" content**=**"text/html; charset=utf-8" **/>**
5. **<**title**>**插入排序**</**title**>**
6. **</**head**>**
7. **<**body**>**
8. **</**body**>**
9. **</**html**>**

**步骤二：为页面添加 <script> 代码块**

为页面添加<script> 代码块，代码如下所示：

1. **<!**DOCTYPE html**>**
2. **<**html**>**
3. **<**head**>**
4. **<**meta http**-**equiv**=**"Content-Type" content**=**"text/html; charset=utf-8" **/>**
5. **<**title**>**插入排序**</**title**>**
6. **<script type=**"text/javascript"**>**
7. **</script>**
8. **</**head**>**
9. **<**body**>**
10. **</**body**>**
11. **</**html**>**

**步骤三：定义函数 insertionSort**

在 <script> 块中定义函数insertionSort，实现插入排序。代码如下所示：

1. **<!**DOCTYPE html**>**
2. **<**html**>**
3. **<**head**>**
4. **<**meta http**-**equiv**=**"Content-Type" content**=**"text/html; charset=utf-8" **/>**
5. **<**title**>**插入排序**</**title**>**
6. **<**script type**=**"text/javascript"**>**
7. //排序方法
8. **function** insertionSort**(arr) {**
9. **console.**log**(**"----------插入排序 开始----------"**);**
10. //从第二个元素开始
11. **for (var i = 1; i < arr.length; i++) {**
12. // 取出待比较的元素
13. **var k = arr[i];**
14. // 向前找，找到比当前元素大的位置
15. **var j;**
16. **for (j = i - 1; j >= 0 && k < arr[j]; j--) {**
17. //向后移动一位
18. **arr[j + 1] = arr[j];**
19. **}**
20. // 插入元素
21. **arr[j + 1] = k;**
22. //输出中间排序的结果
23. **console.**log**(arr.**toString**());**
24. **}**
25. **console.**log**(**"----------插入排序 结束----------"**);**
26. **}**
27. **</**script**>**
28. **</**head**>**
29. **<**body**>**
30. **</**body**>**
31. **</**html**>**

**步骤四：测试排序**

定义无序的数组，并调用函数insertionSort，输出排序前和排序后的结果，测试排序算法。代码如下所示：

1. **<!**DOCTYPE html**>**
2. **<**html**>**
3. **<**head**>**
4. **<**meta http**-**equiv**=**"Content-Type" content**=**"text/html; charset=utf-8" **/>**
5. **<**title**>**插入排序**</**title**>**
6. **<**script type**=**"text/javascript"**>**
7. //排序方法
8. **function** insertionSort**(**arr**)** **{**
9. console**.**log**(**"----------插入排序 开始----------"**);**
10. //从第二个元素开始
11. **for** **(var** i **=** **1;** i **<** arr**.**length**;** i**++)** **{**
12. // 取出待比较的元素
13. **var** k **=** arr**[**i**];**
14. // 向前找，找到比当前元素大的位置
15. **var** j**;**
16. **for** **(**j **=** i **-** **1;** j **>=** **0** **&&** k **<** arr**[**j**];** j**--)** **{**
17. //向后移动一位
18. arr**[**j **+** **1]** **=** arr**[**j**];**
19. **}**
20. // 插入元素
21. arr**[**j **+** **1]** **=** k**;**
22. //输出中间排序的结果
23. console**.**log**(**arr**.**toString**());**
24. **}**
25. console**.**log**(**"----------插入排序 结束----------"**);**
26. **}**
27. //测试
28. **var arr = [12, 15, 9, 20, 6, 31, 24];**
29. //输出
30. **console.**log**(arr.**toString**());**
31. insertionSort**(arr);**
32. **console.**log**(arr.**toString**(arr));**
33. **</**script**>**
34. **</**head**>**
35. **<**body**>**
36. **</**body**>**
37. **</**html**>**

exercise01\_03.html中的完整代码如下所示：

[代码](http://pdf7.tarena.com.cn/tts8_source/ttsPage/WEB/WEB_A_V02/JAVASCRIPTCORE/DAY01/EXERCISE/01/index_answer.html)

**4 十进制转八进制**

八进制，是一种以8为基数的计数法，采用0，1，2，3，4，5，6，7八个数字，逢八进1。当需要将十进制转换为八进制时，可以采用除8取余法。

比如，将十进制数115转化为八进制数，则按照如下方式：

115 除以 8，商为 14，余数为 3；

14 除以 8 ，商为 1，余数为 6；

1 除以 8，商为 0，余数为 1。

因此，十进制数 115 转换为 8 进制为 163。

需要编写函数，实现将十进制数转换为八进制数的功能，并打印显示。

**参考答案**

实现此案例需要按照如下步骤进行。

**步骤一：创建页面**

首先创建一个纯文本文件，并修改名称为 exercise01\_04.html，然后在文本编辑工具中打开此文件，为其添加 HTML 代码，以创建一个标准结构的 HTML 文档，并设置文档的标题以及编码格式。代码如下所示：

1. **<!**DOCTYPE html**>**
2. **<**html**>**
3. **<**head**>**
4. **<**meta http**-**equiv**=**"Content-Type" content**=**"text/html; charset=utf-8" **/>**
5. **<**title**>**插入排序**</**title**>**
6. **</**head**>**
7. **<**body**>**
8. **</**body**>**
9. **</**html**>**

**步骤二：为页面添加 <script> 代码块**

为页面添加<script> 代码块，代码如下所示：

1. **<!**DOCTYPE HTML**>**
2. **<**html**>**
3. **<**head**>**
4. **<**title**>**十进制转为八进制**</**title**>**
5. **<**meta http**-**equiv**=**"content-type" content**=**"text/html;charset=utf-8" **/>**
6. **<script type=**"text/javascript"**>**
7. **</script>**
8. **</**head**>**
9. **<**body**>**
10. **</**body**>**
11. **</**html**>**

**步骤三：定义函数 dec8bin**

在 <script> 块中定义函数dec8bin，实现十进制转换为八进制。代码如下所示：

1. **<!**DOCTYPE HTML**>**
2. **<**html**>**
3. **<**head**>**
4. **<**title**>**十进制转为八进制**</**title**>**
5. **<**meta http**-**equiv**=**"content-type" content**=**"text/html;charset=utf-8" **/>**
6. **<**script type**=**"text/javascript"**>**
7. /\*
8. \* 创建一个函数，接收一个十进制的整数，返回其对应的八进制数表示
9. \*/
10. **function** dec8bin**(num) {**
11. **var stack = [];** //使用栈结构来保存每次除得的余数
12. **while (num > 0) {**
13. **var m = num % 8;**
14. **stack.**push**(m);** //把每次除得的余数入栈
15. **num = parseInt(num / 8);**
16. **}**
17. //出栈，获得转换后的结果
18. **var result =** ''**;**
19. **while (stack.length > 0) {**
20. **result += stack.**pop**();**
21. **}**
22. **return result;**
23. **}**
24. **</**script**>**
25. **</**head**>**
26. **<**body**>**
27. **</**body**>**
28. **</**html**>**

**步骤四：测试**

在 <body> 中添加按钮，调用函数dec8bin，输出结果，测试算法。代码如下所示：

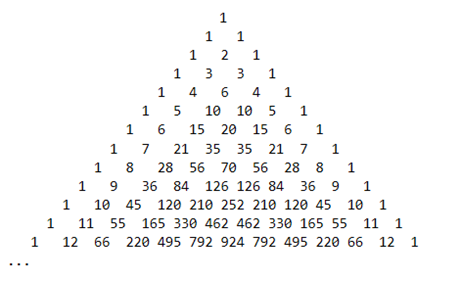
1. **<!**DOCTYPE HTML**>**
2. **<**html**>**
3. **<**head**>**
4. **<**title**>**十进制转为八进制**</**title**>**
5. **<**meta http**-**equiv**=**"content-type" content**=**"text/html;charset=utf-8" **/>**
6. **<**script type**=**"text/javascript"**>**
7. /\*
8. \* 创建一个函数，接收一个十进制的整数，返回其对应的八进制数表示
9. \*/
10. **function** dec8bin**(**num**)** **{**
11. **var** stack **=** **[];** //使用栈结构来保存每次除得的余数
12. **while** **(**num **>** **0)** **{**
13. **var** m **=** num **%** **8;**
14. stack**.**push**(**m**);** //把每次除得的余数入栈
15. num **=** parseInt**(**num **/** **8);**
16. **}**
17. //出栈，获得转换后的结果
18. **var** result **=** ''**;**
19. **while** **(**stack**.**length **>** **0)** **{**
20. result **+=** stack**.**pop**();**
21. **}**
22. **return** result**;**
23. **}**
24. **</**script**>**
25. **</**head**>**
26. **<**body**>**
27. **<button onclick=**"console.log(dec8bin(19968))"**>进制转换为八进制</button>**
28. **</**body**>**
29. **</**html**>**

exercise01\_04.html中的完整代码如下所示：

[代码](http://pdf7.tarena.com.cn/tts8_source/ttsPage/WEB/WEB_A_V02/JAVASCRIPTCORE/DAY01/EXERCISE/01/index_answer.html)

**1 打印杨辉三角形**

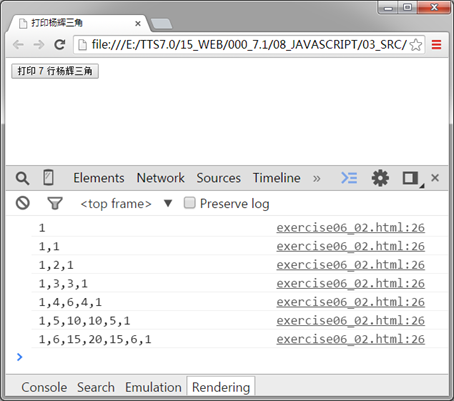
杨辉三角，又称贾宪三角形，帕斯卡三角形，是二项式系数在三角形中的一种几何排列。其内容如图－1所示：



图－1

由图－1可以看出，每行数字左右对称，由1开始逐渐变大；第n行的数字有n项；每个数字等于上一行的左右两个数字之和。即，第n+1行的第i个数等于第n行的第i-1个数和第i个数之和，这也是组合数的性质之一。

请编写程序，在控制台打印输出固定行数的杨辉三角，效果如图－2所示。



图－2

**参考答案**

实现此案例需要按照如下步骤进行。

**步骤一：创建页面**

首先创建一个纯文本文件，并修改名称为 exercise02\_01.html，然后在文本编辑工具中打开此文件，为其添加 HTML 代码，以创建一个标准结构的 HTML 文档，并设置文档的标题以及编码格式。代码如下所示：

1. **<!**DOCTYPE html**>**
2. **<**html**>**
3. **<**head**>**
4. **<**meta http**-**equiv**=**"Content-Type" content**=**"text/html; charset=utf-8" **/>**
5. **<**title**>**打印杨辉三角**</**title**>**
6. **</**head**>**
7. **<**body**>**
8. **</**body**>**
9. **</**html**>**

**步骤二：为页面添加 <script> 代码块**

为页面添加<script> 代码块，代码如下所示：

1. **<!**DOCTYPE HTML**>**
2. **<**html**>**
3. **<**head**>**
4. **<**title**>**打印杨辉三角**</**title**>**
5. **<**meta http**-**equiv**=**"content-type" content**=**"text/html;charset=utf-8" **/>**
6. **<script type=**"text/javascript"**>**
7. **</script>**
8. **</**head**>**
9. **<**body**>**
10. **</**body**>**
11. **</**html**>**

**步骤三：定义函数 triangle**

在 <script> 块中定义函数triangle，实现计算杨辉三角。代码如下所示：

1. **<!**DOCTYPE HTML**>**
2. **<**html**>**
3. **<**head**>**
4. **<**title**>**打印杨辉三角**</**title**>**
5. **<**meta http**-**equiv**=**"content-type" content**=**"text/html;charset=utf-8" **/>**
6. **<**script type**=**"text/javascript"**>**
7. /\*\*
8. \* 创建一个函数，接收一个十进制的整数，返回杨辉三角数据的二维数组表示
9. \*/
10. **function** triangle**(num) {**
11. //先定义前两行的表示
12. **var result = [[1], [1, 1]];**
13. //计算
14. **for (var i = 2; i < num; i++) {**
15. //第一个数为1
16. **result.**push**([1]);**
17. //中间的每个数为上一行的左右两个数字之和
18. **for (var j = 1; j < i; j++) {**
19. **result[i][j] = result[i - 1][j - 1] + result[i - 1][j];**
20. **}**
21. //最后一个数为1
22. **result[i][i] = 1;**
23. **}**
24. **return result;**
25. **}**
26. **</**script**>**
27. **</**head**>**
28. **<**body**>**
29. **</**body**>**
30. **</**html**>**

**步骤四：定义函数 triangle**

在 <script> 块中定义函数printArray，打印计算结果。代码如下所示：

1. **<!**DOCTYPE HTML**>**
2. **<**html**>**
3. **<**head**>**
4. **<**title**>**打印杨辉三角**</**title**>**
5. **<**meta http**-**equiv**=**"content-type" content**=**"text/html;charset=utf-8" **/>**
6. **<**script type**=**"text/javascript"**>**
7. /\*\*
8. \* 创建一个函数，接收一个十进制的整数，返回杨辉三角数据的二维数组表示
9. \*/
10. **function** triangle**(**num**)** **{**
11. //先定义前两行的表示
12. **var** result **=** **[[1],** **[1,** **1]];**
13. //计算
14. **for** **(var** i **=** **2;** i **<** num**;** i**++)** **{**
15. //第一个数为1
16. result**.**push**([1]);**
17. //中间的每个数为上一行的左右两个数字之和
18. **for** **(var** j **=** **1;** j **<** i**;** j**++)** **{**
19. result**[**i**][**j**]** **=** result**[**i **-** **1][**j **-** **1]** **+** result**[**i **-** **1][**j**];**
20. **}**
21. //最后一个数为1
22. result**[**i**][**i**]** **=** **1;**
23. **}**
24. **return** result**;**
25. **}**
26. //分行打印数组
27. **function** printArray**(num) {**
28. **var arr =** triangle**(num);**
29. **for (var i = 0; i < num; i++) {**
30. **console.**log**(arr[i].**toString**());**
31. **}**
32. **}**
33. **</**script**>**
34. **</**head**>**
35. **<**body**>**
36. **</**body**>**
37. **</**html**>**

**步骤五：测试**

在 <body> 中添加按钮，调用函数printArray，输出结果进行测试。代码如下所示：

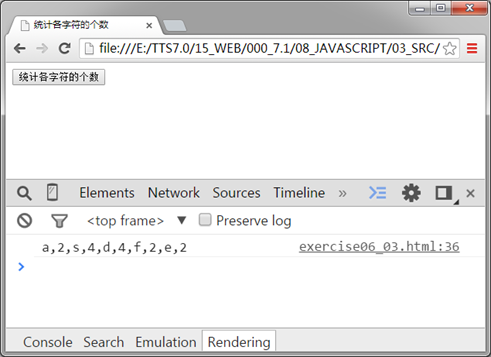
1. **<!**DOCTYPE HTML**>**
2. **<**html**>**
3. **<**head**>**
4. **<**title**>**打印杨辉三角**</**title**>**
5. **<**meta http**-**equiv**=**"content-type" content**=**"text/html;charset=utf-8" **/>**
6. **<**script type**=**"text/javascript"**>**
7. /\*\*
8. \* 创建一个函数，接收一个十进制的整数，返回杨辉三角数据的二维数组表示
9. \*/
10. **function** triangle**(**num**)** **{**
11. //先定义前两行的表示
12. **var** result **=** **[[1],** **[1,** **1]];**
13. //计算
14. **for** **(var** i **=** **2;** i **<** num**;** i**++)** **{**
15. //第一个数为1
16. result**.**push**([1]);**
17. //中间的每个数为上一行的左右两个数字之和
18. **for** **(var** j **=** **1;** j **<** i**;** j**++)** **{**
19. result**[**i**][**j**]** **=** result**[**i **-** **1][**j **-** **1]** **+** result**[**i **-** **1][**j**];**
20. **}**
21. //最后一个数为1
22. result**[**i**][**i**]** **=** **1;**
23. **}**
24. **return** result**;**
25. **}**
26. //分行打印数组
27. **function** printArray**(**num**)** **{**
28. **var** arr **=** triangle**(**num**);**
29. **for** **(var** i **=** **0;** i **<** num**;** i**++)** **{**
30. console**.**log**(**arr**[**i**].**toString**());**
31. **}**
32. **}**
33. **</**script**>**
34. **</**head**>**
35. **<**body**>**
36. **<button onclick=**"printArray(7);"**>打印 7 行杨辉三角</button>**
37. **</**body**>**
38. **</**html**>**

exercise02\_01.html中的完整代码如下所示：

[代码](http://pdf7.tarena.com.cn/tts8_source/ttsPage/WEB/WEB_A_V02/JAVASCRIPTCORE/DAY02/EXERCISE/01/index_answer.html)

**2 统计各字符的个数。**

请编写函数，统计一个字符串中各字符的个数。比如，有字符串“asdfeasddessdf”，使用该函数统计的结果如图－3所示。

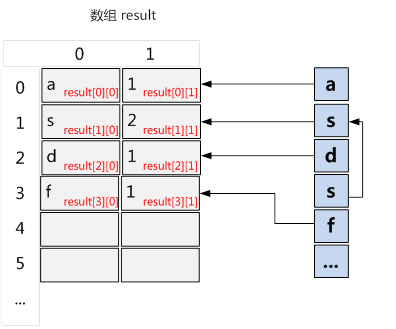


图－3

由图－3可以看出，字符a有两个，字符s有4个，依次类推。

**参考答案**

分析此案例可以看出，需要存储各字符，以及各字符所对应的个数，因此，可以使用二维数组来实现数据的存储。存储方式如图－4所示：



图－4

图－4中右侧的蓝色框，用于挨个显示字符串中的每个字符，左侧表示一个二维数组，其中：第一个维度上存储字符本身，第二个维度上存储该字符的个数。

在算法实现上，只需要挨个读取每个字符，然后查找数组 result 中是否存在此字符：如果存在，则将其对应的数值加 1；如果不存在，则存入新数据即可。

实现此案例需要按照如下步骤进行。

**步骤一：创建页面**

首先创建一个纯文本文件，并修改名称为 exercise02\_02.html，然后在文本编辑工具中打开此文件，为其添加 HTML 代码，以创建一个标准结构的 HTML 文档，并设置文档的标题以及编码格式。代码如下所示：

1. **<!**DOCTYPE html**>**
2. **<**html**>**
3. **<**head**>**
4. **<**meta http**-**equiv**=**"Content-Type" content**=**"text/html; charset=utf-8" **/>**
5. **<**title**>**统计各字符的个数**</**title**>**
6. **</**head**>**
7. **<**body**>**
8. **</**body**>**
9. **</**html**>**

**步骤二：为页面添加 <script> 代码块**

为页面添加<script> 代码块，代码如下所示：

1. **<!**DOCTYPE HTML**>**
2. **<**html**>**
3. **<**head**>**
4. **<**title**>**统计各字符的个数**</**title**>**
5. **<**meta http**-**equiv**=**"content-type" content**=**"text/html;charset=utf-8" **/>**
6. **<script type=**"text/javascript"**>**
7. **</script>**
8. **</**head**>**
9. **<**body**>**
10. **</**body**>**
11. **</**html**>**

**步骤三：定义函数 countString**

在 <script> 块中定义函数countString，用于统计各字符的个数。代码如下所示：

1. **<!**DOCTYPE HTML**>**
2. **<**html**>**
3. **<**head**>**
4. **<**title**>**统计各字符的个数**</**title**>**
5. **<**meta http**-**equiv**=**"content-type" content**=**"text/html;charset=utf-8" **/>**
6. **<**script type**=**"text/javascript"**>**
7. //统计各字符的个数：使用二维数组
8. **function** countString**(str) {**
9. **var result = new Array();**
10. //直接循环str
11. **for (var i = 0; i < str.length; i++) {**
12. //得到当前字符
13. **var curChar = str.**charAt**(i);**
14. //声明一个变量，标识char在结果中是否出现过
15. **var isHas = false;**
16. //循环判断当前字符是否在result中出现过
17. **for (var j = 0; j < result.length; j++) {**
18. //如果出现过，则设置标识为true，并增加数量，最后跳出循环
19. **if (curChar == result[j][0]) {**
20. **isHas = true;**
21. **result[j][1]++;**
22. **break;**
23. **}**
24. **}**
25. //循环result完毕，没有出现过，则加入result
26. **if (!isHas)**
27. **result.**push**(new Array(curChar, 1));**
28. **}**
29. **return result;**
30. **}**
31. **</**script**>**
32. **</**head**>**
33. **<**body**>**
34. **</**body**>**
35. **</**html**>**

**步骤四：测试**

在 <body> 中添加按钮，调用函数countString，输出结果进行测试。代码如下所示：

1. **<!**DOCTYPE HTML**>**
2. **<**html**>**
3. **<**head**>**
4. **<**title**>**统计各字符的个数**</**title**>**
5. **<**meta http**-**equiv**=**"content-type" content**=**"text/html;charset=utf-8" **/>**
6. **<**script type**=**"text/javascript"**>**
7. //统计各字符的个数：使用二维数组
8. **function** countString**(**str**)** **{**
9. **var** result **=** **new** Array**();**
10. //直接循环str
11. **for** **(var** i **=** **0;** i **<** str**.**length**;** i**++)** **{**
12. //得到当前字符
13. **var** curChar **=** str**.**charAt**(**i**);**
14. //声明一个变量，标识char在结果中是否出现过
15. **var** isHas **=** **false;**
16. //循环判断当前字符是否在result中出现过
17. **for** **(var** j **=** **0;** j **<** result**.**length**;** j**++)** **{**
18. //如果出现过，则设置标识为true，并增加数量，最后跳出循环
19. **if** **(**curChar **==** result**[**j**][0])** **{**
20. isHas **=** **true;**
21. result**[**j**][1]++;**
22. **break;**
23. **}**
24. **}**
25. //循环result完毕，没有出现过，则加入result
26. **if** **(!**isHas**)**
27. result**.**push**(new** Array**(**curChar**,** **1));**
28. **}**
29. **return** result**;**
30. **}**
31. **</**script**>**
32. **</**head**>**
33. **<**body**>**
34. **<button onclick=**"console.log(countString('asdfeasddessdf').toString());"**>统计各字符的个数</button>**
35. **</**body**>**
36. **</**html**>**

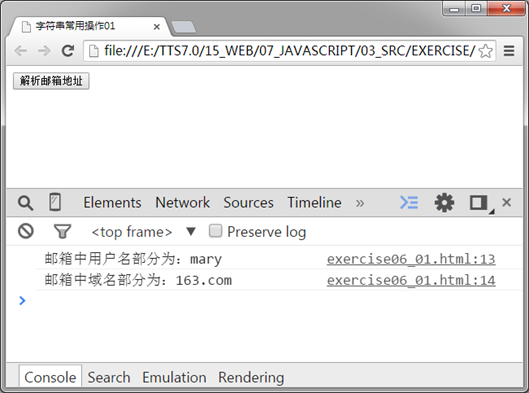
exercise02\_02.html中的完整代码如下所示：

[代码](http://pdf7.tarena.com.cn/tts8_source/ttsPage/WEB/WEB_A_V02/JAVASCRIPTCORE/DAY02/EXERCISE/01/index_answer.html)

**3 解析邮箱地址**

解析一个邮箱地址，分别获得其中的用户名和服务器域名。邮箱地址形如：xxx@xxx.xx。其中，以 @ 字符为分割点，符号之前就是用户名，符号之后是服务器域名。

试编写一个函数，接收一个邮箱地址，找出其中的用户名和服务器域名，并在控制台打印显示。比如，传给解析函数的邮箱地址为 mary@163.com，单击按钮后，调用解析函数，解析结果如图－5所示。



图－5

**参考答案**

实现此案例需要按照如下步骤进行。

**步骤一：创建页面**

首先创建一个纯文本文件，并修改名称为 exercise02\_03.html，然后在文本编辑工具中打开此文件，为其添加 HTML 代码，以创建一个标准结构的 HTML 文档，并设置文档的标题以及编码格式。代码如下所示：

1. **<!**DOCTYPE HTML**>**
2. **<**html**>**
3. **<**head**>**
4. **<**title**>**字符串常用操作**01</**title**>**
5. **<**meta http**-**equiv**=**"content-type" content**=**"text/html;charset=utf-8" **/>**
6. **</**head**>**
7. **<**body**>**
8. **</**body**>**
9. **</**html**>**

**步骤二：创建测试按钮，并定义 <script> 代码块**

为页面添加测试按钮，并定义<script>代码块。代码如下所示：

1. **<!**DOCTYPE HTML**>**
2. **<**html**>**
3. **<**head**>**
4. **<**title**>**字符串常用操作**01</**title**>**
5. **<**meta http**-**equiv**=**"content-type" content**=**"text/html;charset=utf-8" **/>**
6. **<script type=**"text/javascript"**>**
7. **</script>**
8. **</**head**>**
9. **<**body**>**
10. **<button onclick=**"parseEmail('mary@163.com')"**>解析邮箱地址</button>**
11. **</**body**>**
12. **</**html**>**

**步骤三：定义函数 parseEmail**

定义函数 parseEmail，解析邮箱地址。代码如下所示：

1. **<!**DOCTYPE HTML**>**
2. **<**html**>**
3. **<**head**>**
4. **<**title**>**字符串常用操作**01</**title**>**
5. **<**meta http**-**equiv**=**"content-type" content**=**"text/html;charset=utf-8" **/>**
6. **<**script type**=**"text/javascript"**>**
7. //解析一个邮箱地址，分别获得其中的用户名和服务器域名
8. **function** parseEmail**(email) {**
9. //由于邮箱中有且只有@字符，以此为分割点，前后分别就是用户名和服务器域名
10. **var index = email.**indexOf**(**'@'**);**
11. **var userName = email.**substring**(0, index);**
12. **var hostName = email.**substring**(index + 1);**
13. **console.**log**(**'邮箱中用户名部分为：' **+ userName);**
14. **console.**log**(**'邮箱中域名部分为：' **+ hostName);**
15. **}**
16. **</**script**>**
17. **</**head**>**
18. **<**body**>**
19. **<**button onclick**=**"parseEmail('mary@163.com')"**>**解析邮箱地址**</**button**>**
20. **</**body**>**
21. **</**html**>**

文件exercise02\_03.html中的完整代码如下所示：

[代码](http://pdf7.tarena.com.cn/tts8_source/ttsPage/WEB/WEB_A_V02/JAVASCRIPTCORE/DAY02/EXERCISE/01/index_answer.html)

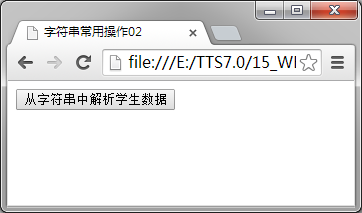
**4 解析字符串**

假设浏览器接收到服务器返回的字符串，其格式如下所示：

Tom@95@88#Mary@79@65#John@91@98

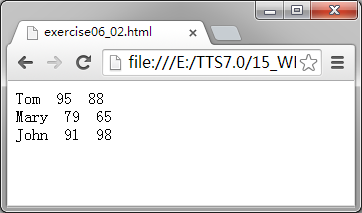
其中，每个学员的信息之间以 # 分隔，如Tom@95@88表示一个学员的信息。其中，Tom表示学员的姓名，95 和 88 表示学员的两个成绩。

试编写一个函数，从字符串中解析学员的姓名以及成绩信息，并存入一个二维数组，然后在控制台打印显示。页面效果如图－6所示。



图－6

单击页面上的按钮后，调用解析函数，解析结果如图－7所示。



图－7

**参考答案**

实现此案例需要按照如下步骤进行。

**步骤一：创建页面**

首先创建一个纯文本文件，并修改名称为 exercise02\_04.html，然后在文本编辑工具中打开此文件，为其添加 HTML 代码，以创建一个标准结构的 HTML 文档，并设置文档的标题以及编码格式。代码如下所示：

1. **<!**DOCTYPE HTML**>**
2. **<**html**>**
3. **<**head**>**
4. **<**title**>**字符串常用操作**02</**title**>**
5. **<**meta http**-**equiv**=**"content-type" content**=**"text/html;charset=utf-8" **/>**
6. **</**head**>**
7. **<**body**>**
8. **</**body**>**
9. **</**html**>**

**步骤二：创建测试按钮，并定义 <script> 代码块**

为页面添加测试按钮，并定义<script>代码块。代码如下所示：

1. **<!**DOCTYPE HTML**>**
2. **<**html**>**
3. **<**head**>**
4. **<**title**>**字符串常用操作**02</**title**>**
5. **<**meta http**-**equiv**=**"content-type" content**=**"text/html;charset=utf-8" **/>**
6. **<script type=**"text/javascript"**>**
7. **</script>**
8. **</**head**>**
9. **<**body**>**
10. **<button onclick=**"parseAndPrint()"**> 从字符串中解析学生数据</button>**
11. **</**body**>**
12. **</**html**>**

**步骤三：定义函数 parseAndPrint**

定义函数 parseAndPrint，解析字符串信息。在该函数中，现调用方法parseStudentData，得到解析后的结果（一个二维数组），再调用方法print2Darray，打印显示结果。代码如下所示：

1. **<!**DOCTYPE HTML**>**
2. **<**html**>**
3. **<**head**>**
4. **<**title**>**字符串常用操作**02</**title**>**
5. **<**meta http**-**equiv**=**"content-type" content**=**"text/html;charset=utf-8" **/>**
6. **<**script type**=**"text/javascript"**>**
7. //解析数据，并打印结果
8. **function** parseAndPrint**() {**
9. **var data =** 'Tom@95@88#Mary@79@65#John@91@98'**;**
10. **var result =** parseStudentData**(data);**
11. print2DArray**(result);**
12. **}**
13. **</**script**>**
14. **</**head**>**
15. **<**body**>**
16. **<**button onclick**=**"parseAndPrint()"**>** 从字符串中解析学生数据**</**button**>**
17. **</**body**>**
18. **</**html**>**

**步骤四：定义函数 parseStudentData**

定义函数 parseStudentData，解析字符串信息。代码如下所示：

1. **<!**DOCTYPE HTML**>**
2. **<**html**>**
3. **<**head**>**
4. **<**title**>**字符串常用操作**02</**title**>**
5. **<**meta http**-**equiv**=**"content-type" content**=**"text/html;charset=utf-8" **/>**
6. **<**script type**=**"text/javascript"**>**
7. //解析数据，并打印结果
8. **function** parseAndPrint**()** **{**
9. **var** data **=** 'Tom@95@88#Mary@79@65#John@91@98'**;**
10. **var** result **=** parseStudentData**(**data**);**
11. print2DArray**(**result**);**
12. **}**
13. //字符串格式如：Tom@95@88#Mary@79@65#John@91@98
14. //试将其解析出来，保存入一个二维数组
15. **function** parseStudentData**(data) {**
16. **var result = [];**
17. //必须先用#进行拆分，而不能先用@拆分
18. **var stuArr = data.**split**(**'#'**);**
19. **for (var i = 0; i < stuArr.length; i++) {**
20. **var stu = stuArr[i];**
21. **var details = stu.**split**(**'@'**);**
22. **result.**push**(details);**
23. **}**
24. **return result;**
25. **}**
26. **</**script**>**
27. **</**head**>**
28. **<**body**>**
29. **<**button onclick**=**"parseAndPrint()"**>** 从字符串中解析学生数据**</**button**>**
30. **</**body**>**
31. **</**html**>**

**步骤五：定义函数 print2DArray**

定义函数 print2DArray，打印解析结果。代码如下所示：

1. **<!**DOCTYPE HTML**>**
2. **<**html**>**
3. **<**head**>**
4. **<**title**>**字符串常用操作**02</**title**>**
5. **<**meta http**-**equiv**=**"content-type" content**=**"text/html;charset=utf-8" **/>**
6. **<**script type**=**"text/javascript"**>**
7. //解析数据，并打印结果
8. **function** parseAndPrint**()** **{**
9. **var** data **=** 'Tom@95@88#Mary@79@65#John@91@98'**;**
10. **var** result **=** parseStudentData**(**data**);**
11. print2DArray**(**result**);**
12. **}**
13. //字符串格式如：Tom@95@88#Mary@79@65#John@91@98
14. //试将其解析出来，保存入一个二维数组
15. **function** parseStudentData**(**data**)** **{**
16. **var** result **=** **[];**
17. //必须先用#进行拆分，而不能先用@拆分
18. **var** stuArr **=** data**.**split**(**'#'**);**
19. **for** **(var** i **=** **0;** i **<** stuArr**.**length**;** i**++)** **{**
20. **var** stu **=** stuArr**[**i**];**
21. **var** details **=** stu**.**split**(**'@'**);**
22. result**.**push**(**details**);**
23. **}**
24. **return** result**;**
25. **}**
26. //打印二维数组中的数据
27. **function** print2DArray**(arr) {**
28. **for (var i = 0; i < arr.length; i++) {**
29. **var row = arr[i];**
30. **for (var j = 0; j < row.length; j++) {**
31. **document.**write**(row[j] +** "&nbsp;&nbsp;"**);**
32. **}**
33. **document.**write**(**"<br/>"**);**
34. **}**
35. **}**
36. **</**script**>**
37. **</**head**>**
38. **<**body**>**
39. **<**button onclick**=**"parseAndPrint()"**>** 从字符串中解析学生数据**</**button**>**
40. **</**body**>**
41. **</**html**>**

文件exercise02\_04.html中的完整代码如下所示：

[代码](http://pdf7.tarena.com.cn/tts8_source/ttsPage/WEB/WEB_A_V02/JAVASCRIPTCORE/DAY02/EXERCISE/01/index_answer.html)

**5 字符串拆分后的结果问题**

已知字符串',a,,b,'其中没有空白字符，使用逗号对其进行拆分，得到的结果是如下哪一项：

(A) [null, 'a', null, 'b', null]

(B) ['', 'a', '', 'b', '']

(C) [null, 'a', null, 'b']

(D) ['', 'a', '', 'b']

(E) ['a', 'b']

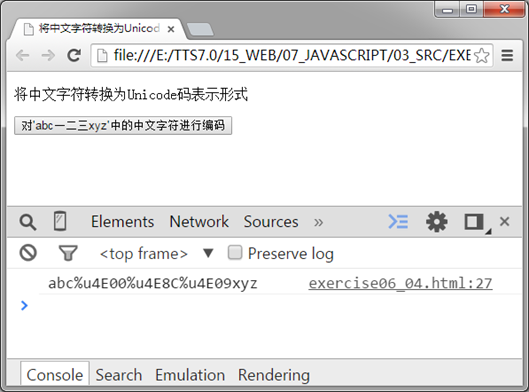
**参考答案**

JavaScript中的字符串拆分与其它语言(如Java)中的拆分结果不同。若原始字符串以拆分字符串开头或结果，那么拆分标记之前和之后都可以拆分出长度为0的字符串，而不是null。故此题选B。

**6 将中文字符转换为Unicode码表示形式**

早期的JavaScript中有一个escape()函数，可用于将字符串中的中文转换为对应的Unicode码表示形式，例如，字符串'abc一二三xyz'对应的Unicode码表示形式为，abc%u4E00%u4E8C%u4E09xyz。

请仿写一个类似的方法myEscape()，接收一个字符串参数，返回一个新的字符串，原串中的中文用形如%uXXXX形式表示。页面效果如图－7所示。



图－7

**参考答案**

解决方案为，依次遍历原字符串中的每个字符，获得该字符对应的unicode码，并判断该字符是否为中文（中文对应的十进制的编码范围为19968-40869），如果是中文，则将其转换为十六进制表示，并用十六进制表示替换原字符串中出现的对应中文字符即可。

实现此案例需要按照如下步骤进行。

**步骤一：创建页面**

首先创建一个纯文本文件，并修改名称为 exercise02\_06.html，然后在文本编辑工具中打开此文件，为其添加 HTML 代码，以创建一个标准结构的 HTML 文档，并设置文档的标题以及编码格式。代码如下所示：

1. **<!**DOCTYPE html**>**
2. **<**html**>**
3. **<**head**>**
4. **<**meta http**-**equiv**=**"Content-Type" content**=**"text/html; charset=utf-8" **/>**
5. **<**title**>**将中文字符转换为Unicode码表示形式**</**title**>**
6. **</**head**>**
7. **<**body**>**
8. **</**body**>**
9. **</**html**>**

**步骤二：添加按钮，绑定单击事件处理方法，并创建 <script> 代码块**

为页面添加测试按钮，并定义<script>代码块。代码如下所示：

1. **<!**DOCTYPE html**>**
2. **<**html**>**
3. **<**head**>**
4. **<**meta http**-**equiv**=**"Content-Type" content**=**"text/html; charset=utf-8" **/>**
5. **<**title**>**将中文字符转换为Unicode码表示形式**</**title**>**
6. **<script type=**"text/javascript"**>**
7. **</script>**
8. **</**head**>**
9. **<**body**>**
10. **<p>将中文字符转换为Unicode码表示形式</p>**
11. **<button onclick=**"console.log( myEscape('abc一二三xyz') )"**>对**'abc一二三xyz'**中的中文字符进行编码</button>**
12. **</**body**>**
13. **</**html**>**

**步骤三：定义函数 myEscape**

定义函数 myEscape，解析字符串。代码如下所示：

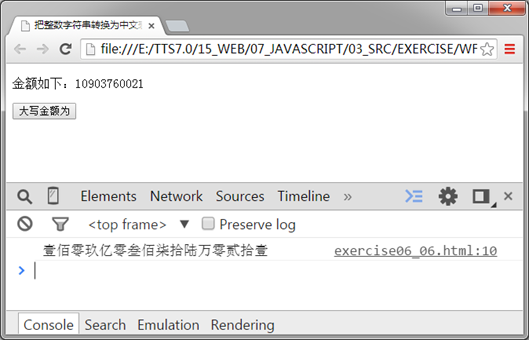
1. **<!**DOCTYPE html**>**
2. **<**html**>**
3. **<**head**>**
4. **<**meta http**-**equiv**=**"Content-Type" content**=**"text/html; charset=utf-8" **/>**
5. **<**title**>**将中文字符转换为Unicode码表示形式**</**title**>**
6. **<**script type**=**"text/javascript"**>**
7. //对中文字符进行编码
8. **function** myEscape**(str) {**
9. **for (var i = 0; i < str.length; i++) {**
10. //获得第i个字符串对应的十进制Unicode码，并判断是否为中文
11. **var code = str.**charCodeAt**(i);**
12. **if (code >= 19968 && code <= 40869) {**
13. //将十进制的编码数转换为十六进制数
14. **var unicode =** '%u' **+ code.**toString**(16).**toUpperCase**();**
15. //替换字符
16. **var c = str.**charAt**(i);**
17. **str = str.**replace**(c, unicode);**
18. **}**
19. **}**
20. **return str;**
21. **}**
22. **</**script**>**
23. **</**head**>**
24. **<**body**>**
25. **<**p**>**将中文字符转换为Unicode码表示形式**</**p**>**
26. **<**button onclick**=**"console.log( myEscape('abc一二三xyz') )"**>**对'abc一二三xyz'中的中文字符进行编码**</**button**>**
27. **</**body**>**
28. **</**html**>**

文件exercise02\_06.html中的完整代码如下所示：

[代码](http://pdf7.tarena.com.cn/tts8_source/ttsPage/WEB/WEB_A_V02/JAVASCRIPTCORE/DAY02/EXERCISE/01/index_answer.html)

**1 把数值类型金额转换为大写表示形式，仅限整数（提高题，选做）**

创建一个函数，接收一个表示金额的整数字符串，且数字长度没有限制，比如10903760000；由函数返回该数字的大写金额表示形式，结果为：壹佰零玖亿零叁佰柒拾陆万。界面效果如图－1所示。



图－1

**参考答案**

对于不限长度的数字字符串，应该从低位到高位依次处理，且从低到高位八位分为一组，然后再将每组中的低四位分为一小组，高四位分为一小组分别处理。处理过程如图-2所示：

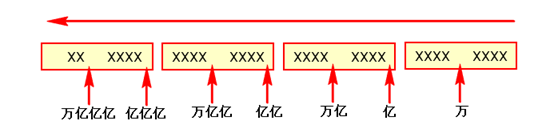


图-2

实现此案例需要按照如下步骤进行。

**步骤一：创建页面**

首先创建一个纯文本文件，并修改名称为 exercise03\_01.html，然后在文本编辑工具中打开此文件，为其添加 HTML 代码，以创建一个标准结构的 HTML 文档（使用过渡型文档类型声明），并设置文档的标题以及编码格式。代码如下所示：

1. **<!**DOCTYPE html**>**
2. **<**html**>**
3. **<**head**>**
4. **<**meta http**-**equiv**=**"Content-Type" content**=**"text/html; charset=utf-8" **/>**
5. **<**title**>**把整数字符串转换为中文表示形式**</**title**>**
6. **</**head**>**
7. **<**body**>**
8. **</**body**>**
9. **</**html**>**

**步骤二：创建测试按钮，并添加外部js文件**

为页面添加测试按钮，并引入外部 js 文件。代码如下所示：

1. **<!**DOCTYPE html**>**
2. **<**html**>**
3. **<**head**>**
4. **<**meta http**-**equiv**=**"Content-Type" content**=**"text/html; charset=utf-8" **/>**
5. **<**title**>**把整数字符串转换为中文表示形式**</**title**>**
6. **<script type=**"text/javascript" **src=**"exercise03\_01.js"**></script>**
7. **</**head**>**
8. **<**body**>**
9. **<p>金额如下：10903760021</p>**
10. **<button onclick=**"console.log( dec2Chinese('10903760021') )"**>大写金额为</button>**
11. **</**body**>**
12. **</**html**>**

**步骤三：定义dec2Chinese()函数**

打开外部 js 文件，创建dec2Chinese函数，首先把函数的形参转换为字符串表示形式，并判断录入的是否为整数数据。代码如下所示：

1. //把无限大整数的金额，转换为大写表示形式
2. **function** dec2Chinese**(**str**)** **{**
3. //保存最终结果的数组
4. **var** result **=** **[];**
5. //将参数转换为字符串表示
6. str **=** str**.**toString**();**
7. //只要包含非数字字符，即返回NaN
8. **var** regexp **=** /[^0-9]/**;**
9. **if** **(**regexp**.**test**(**str**))** **{**
10. **return** Number**.**NaN**;**
11. **}**
12. **}**

**步骤四：去掉多余的数字0，并得到字符串的逆序**

接下来，去除字符串前导的所有字符0，并得到字符串的逆序。代码如下所示：

1. //把无限大整数的金额，转换为大写表示形式
2. **function** dec2Chinese**(**str**)** **{**
3. //保存最终结果的数组
4. **var** result **=** **[];**
5. //将参数转换为字符串表示
6. str **=** str**.**toString**();**
7. //只要包含非数字字符，即返回NaN
8. **var** regexp **=** /[^0-9]/**;**
9. **if** **(**regexp**.**test**(**str**))** **{**
10. **return** Number**.**NaN**;**
11. **}**
12. //去除前导数字0
13. **regexp =** /^0\*/**;**
14. **str = str.**replace**(regexp,** ''**);**
15. //将原字符串逆序存储在一个字符数组中
16. **var data = [];**
17. **for (var i = 0; i < str.length; i++) {**
18. **data.**unshift**(str.**charAt**(i));**
19. **}**

**步骤五：处理数值**

每八位一组，处理整数，并转换为大写形式。代码如下所示：

1. //把无限大整数的金额，转换为大写表示形式
2. **function** dec2Chinese**(**str**)** **{**
3. //保存最终结果的数组
4. **var** result **=** **[];**
5. //将参数转换为字符串表示
6. str **=** str**.**toString**();**
7. //只要包含非数字字符，即返回NaN
8. **var** regexp **=** /[^0-9]/**;**
9. **if** **(**regexp**.**test**(**str**))** **{**
10. **return** Number**.**NaN**;**
11. **}**
12. //去除前导数字0
13. regexp **=** /^0\*/**;**
14. str **=** str**.**replace**(**regexp**,** ''**);**
15. //将原字符串逆序存储在一个字符数组中
16. **var** data **=** **[];**
17. **for** **(var** i **=** **0;** i **<** str**.**length**;** i**++)** **{**
18. data**.**unshift**(**str**.**charAt**(**i**));**
19. **}**
20. //定义对应的大写字符
21. **var digits = [**'零'**,** '壹'**,** '贰'**,** '叁'**,** '肆'**,** '伍'**,** '陆'**,** '柒'**,** '捌'**,** '玖'**];**
22. **var pos = [**''**,** '拾'**,** '佰'**,** '仟'**];**
23. //每八位划分为一组（到亿为止），进行解析
24. **for (var i = 0; i < data.length; i += 8) {**
25. //每次取出八位进行处理
26. **var arr8 = data.**slice**(i, i + 8);**
27. **var yiCount =** ''**;**
28. **for (var k = 0; k < i / 8; k++) {**
29. //亿的数量
30. **yiCount =** '亿' **+ yiCount;**
31. **}**
32. //四位是否都是0
33. **var allZero = true;**
34. //某位上的数字
35. **var num = 0;**
36. //处理这四位数字的一个数组
37. **var arr4 = [];**
38. //八位中的低四位
39. **var arr0\_3 = arr8.**slice**(0, 4);**
40. //处理八位中的低四位
41. **for (var j = 0; j < arr0\_3.length; j++) {**
42. **num = parseInt(arr0\_3[j]);**
43. **if (num == 0) {**
44. //数字中间连续多个0只读出1个零
45. **if (arr4.length > 0 && arr4[0] !=** '零'**) {**
46. **arr4.**unshift**(**'零'**);**
47. **}**
48. **} else {**
49. **arr4.**unshift**(digits[num] + pos[j % 4]);**
50. **allZero = false;**
51. **}**
52. **}**
53. **if (!allZero) {**
54. **result.**unshift**(yiCount);**
55. **result = arr4.**concat**(result);**
56. **}**
57. //处理八位中的高四位
58. **allZero = true;**
59. **num = 0;**
60. **arr4 = [];**
61. //八位中的高四位
62. **var arr4\_7 = arr8.**slice**(4, 8);**
63. **for (var j = 0; j < arr4\_7.length; j++) {**
64. **num = parseInt(arr4\_7[j]);**
65. **if (num == 0) {**
66. //数字中间连续多个0只读出1个零
67. **if (arr4.length > 0 && arr4[0] !=** '零'**) {**
68. **arr4.**unshift**(**'零'**);**
69. **}**
70. **} else {**
71. **arr4.**unshift**(digits[num] + pos[j % 4]);**
72. **allZero = false;**
73. **}**
74. **}**
75. **if (!allZero) {**
76. **result.**unshift**(**'万' **+ yiCount);**
77. **result = arr4.**concat**(result);**
78. **}**
79. **}**
80. **return result.**join**(**''**);**
81. **}**

文件exercise03\_01.html中的完整代码如下所示：

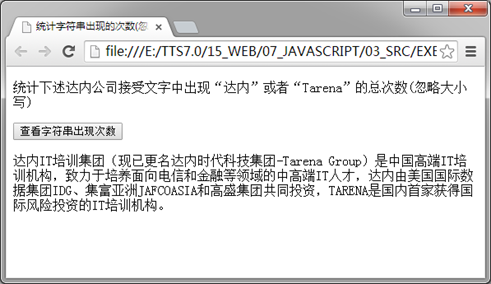
[代码](http://pdf7.tarena.com.cn/tts8_source/ttsPage/WEB/WEB_A_V02/JAVASCRIPTCORE/DAY03/EXERCISE/01/index_answer.html)

文件exercise03\_01.js中的完整代码如下所示：

[代码](http://pdf7.tarena.com.cn/tts8_source/ttsPage/WEB/WEB_A_V02/JAVASCRIPTCORE/DAY03/EXERCISE/01/index_answer.html)

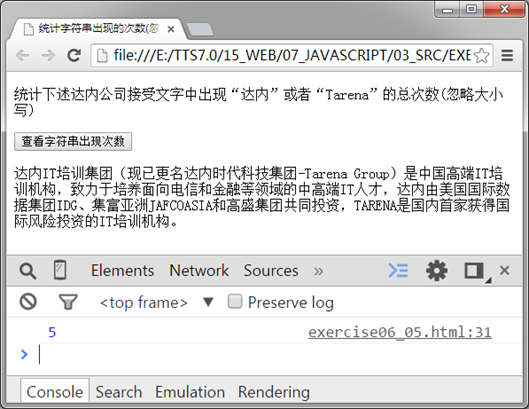
**2 统计字符串出现的次数(忽略参数异常)**

定义一个函数，统计某字符串出现的次数。页面效果如图－3所示。



图－3

图－3中，单击页面上的按钮，需要统计按钮下方大段文字中“达内”、“tarena”字符串出现的次数，其中，“tarena”不区分大小写。单击按钮后的页面效果如图－4所示，即共出现 5 次。



图－4

**参考答案**

此题可以有两种实现方法：既可以使用String类的match()方法，也可以使用RegExp类的exec()方法。本案例使用RegExp对象来实现，按照如下步骤进行。

**步骤一：创建页面**

首先创建一个纯文本文件，并修改名称为 exercise03\_02.html，然后在文本编辑工具中打开此文件，为其添加 HTML 代码，以创建一个标准结构的 HTML 文档，并设置文档的标题以及编码格式。代码如下所示：

1. **<!**DOCTYPE html**>**
2. **<**html**>**
3. **<**head**>**
4. **<**meta http**-**equiv**=**"Content-Type" content**=**"text/html; charset=utf-8" **/>**
5. **<**title**>**统计字符串出现的次数**(**忽略参数异常**)</**title**>**
6. **</**head**>**
7. **<**body**>**
8. **</**body**>**
9. **</**html**>**

**步骤二：添加按钮，绑定单击事件处理方法，并创建 <script> 代码块**

为页面添加测试按钮，并定义<script>代码块。代码如下所示：

1. **<!**DOCTYPE html**>**
2. **<**html**>**
3. **<**head**>**
4. **<**meta http**-**equiv**=**"Content-Type" content**=**"text/html; charset=utf-8" **/>**
5. **<**title**>**统计字符串出现的次数**(**忽略参数异常**)</**title**>**
6. **<script type=**"text/javascript"**>**
7. **</script>**
8. **</**head**>**
9. **<**body**>**
10. **<p>统计下述达内公司接受文字中出现“达内”或者“Tarena”的总次数(忽略大小写)</p>**
11. **<button onclick=**"console.log( getStringCount(src, '达内','tarena') )"**>查看字符串出现次数</button>**
12. **<p>**
13. **达内IT培训集团（现已更名达内时代科技集团-Tarena Group）是中国高端IT培训机构，致力于培养面向电信和金融等领域的中高端IT人才，达内由美国国际数据集团IDG、集富亚洲JAFCOASIA和高盛集团共同投资，TARENA是国内首家获得国际风险投资的IT培训机构。**
14. **</p>**
15. **</**body**>**
16. **</**html**>**

**步骤三：定义函数 getStringCount**

定义函数 getStringCount，实现字符串的查找。代码如下所示：

1. **<!**DOCTYPE html**>**
2. **<**html**>**
3. **<**head**>**
4. **<**meta http**-**equiv**=**"Content-Type" content**=**"text/html; charset=utf-8" **/>**
5. **<**title**>**统计字符串出现的次数**(**忽略参数异常**)</**title**>**
6. **<**script type**=**"text/javascript"**>**
7. //下述字符串中有中文、大小写的英文的达内
8. **var src =** '达内IT培训集团（现已更名达内时代科技集团-Tarena Group）是中国高端IT培训机构，致力于培养面向电信和金融等领域的中高端IT人才，达内由美国国际数据集团IDG、集富亚洲JAFCOASIA和高盛集团共同投资，TARENA是国内首家获得国际风险投资的IT培训机构。'**;**
9. //统计一个字符串出现特定字符串的次数
10. **function** getStringCount**(src, search1, search2) {**
11. //匹配计数器
12. **var count = 0;**
13. //构建匹配
14. **var pattern = search1 +** '|' **+ search2;**
15. **var regexp = new RegExp(pattern,** 'ig'**);**
16. //进行正则表达式匹配，返回所有匹配到的字符串
17. **while ((regexp.**exec**(src)) != null) {**
18. **count++;**
19. **}**
20. **return count;**
21. **}**
22. **</**script**>**
23. **</**head**>**
24. **<**body**>**
25. **<**p**>**统计下述达内公司接受文字中出现“达内”或者“Tarena”的总次数**(**忽略大小写**)</**p**>**
26. **<**button onclick**=**"console.log( getStringCount(src, '达内','tarena') )"**>**查看字符串出现次数**</**button**>**
27. **<**p**>**
28. 达内IT培训集团（现已更名达内时代科技集团**-**Tarena Group）是中国高端IT培训机构，致力于培养面向电信和金融等领域的中高端IT人才，达内由美国国际数据集团IDG、集富亚洲JAFCOASIA和高盛集团共同投资，TARENA是国内首家获得国际风险投资的IT培训机构。
29. **</**p**>**
30. **</**body**>**
31. **</**html**>**

文件exercise03\_02.html中的完整代码如下所示：

[代码](http://pdf7.tarena.com.cn/tts8_source/ttsPage/WEB/WEB_A_V02/JAVASCRIPTCORE/DAY03/EXERCISE/01/index_answer.html)

**3 下列String类的方法中哪些支持正则表达式**

A. split()

B. indexOf()

C. match()

D. replace()

E. search()

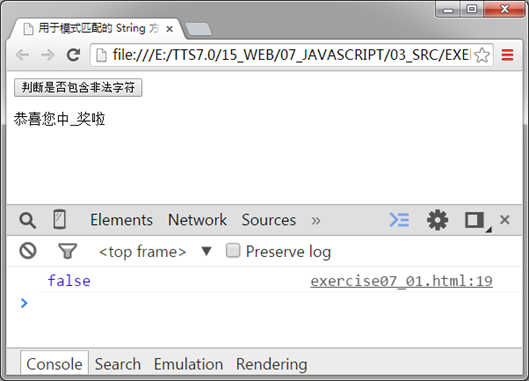
**参考答案**

split()方法使用一个固定的字符串或者正则表达式实现字符串的拆分。indexOf()方法只能接收一个固定内容的字符串，搜索其第一次出现的下标。match()方法接收一个固定字符串或正则表达式，返回匹配的数组。replace()方法实现固定字符串或正则表达式的替换。search()方法检索一个正则表达式第一次出现的下标。

故正确答案：ACDE

**4 判断违禁字符**

编写一个函数，用于判定一段字符串中是否存在违禁字符，如"发票"/"增值税"/"中奖"/"抽奖"等。如果包含违禁字符，则返回true；否则返回false。如图－5所示：



图－5

图－5中，单击按钮，验证按钮下方的文字中，是否包含违禁字符，不包含，则返回false。

**参考答案**

实现此案例需要按照如下步骤进行。

**步骤一：创建页面**

首先创建一个纯文本文件，并修改名称为 exercise03\_04.html，然后在文本编辑工具中打开此文件，为其添加 HTML 代码，以创建一个标准结构的 HTML 文档，并设置文档的标题以及编码格式。代码如下所示：

1. **<!**DOCTYPE HTML**>**
2. **<**html**>**
3. **<**head**>**
4. **<**title**>**用于模式匹配的 String 方法**</**title**>**
5. **<**meta http**-**equiv**=**"content-type" content**=**"text/html;charset=utf-8" **/>**
6. **</**head**>**
7. **<**body**>**
8. **</**body**>**
9. **</**html**>**

**步骤二：创建测试按钮，并定义 <script> 代码块**

为页面添加测试按钮，并定义<script>代码块。代码如下所示：

1. **<!**DOCTYPE HTML**>**
2. **<**html**>**
3. **<**head**>**
4. **<**title**>**用于模式匹配的 String 方法**</**title**>**
5. **<**meta http**-**equiv**=**"content-type" content**=**"text/html;charset=utf-8" **/>**
6. **<script type=**"text/javascript"**>**
7. **</script>**
8. **</**head**>**
9. **<**body**>**
10. **<button onclick=**"console.log( containsIllegal('恭喜您中\_奖啦') )"**>判断是否包含非法字符</button>**
11. **<p>恭喜您中\_奖啦</p>**
12. **</**body**>**
13. **</**html**>**

**步骤三：定义函数 containsIllegal**

定义函数 containsIllegal，解析邮箱地址。代码如下所示：

1. **<!**DOCTYPE HTML**>**
2. **<**html**>**
3. **<**head**>**
4. **<**title**>**用于模式匹配的 String 方法**</**title**>**
5. **<**meta http**-**equiv**=**"content-type" content**=**"text/html;charset=utf-8" **/>**
6. **<**script type**=**"text/javascript"**>**
7. //判定一段字符串中是否存在违禁字符，如"发票"/"增值税"/"中奖"/"抽奖"等
8. **function** containsIllegal**(str) {**
9. **var regexp =** /(发票|增值税|中奖|抽奖)/igm**;**
10. **if (str.**search**(regexp) > -1) {**
11. **return true;**
12. **} else {**
13. **return false;**
14. **}**
15. **}**
16. **</**script**>**
17. **</**head**>**
18. **<**body**>**
19. **<**button onclick**=**"console.log( containsIllegal('恭喜您中\_奖啦') )"**>**判断是否包含非法字符**</**button**>**
20. **<**p**>**恭喜您中\_奖啦**</**p**>**
21. **</**body**>**
22. **</**html**>**

文件exercise03\_04.html中的完整代码如下所示：

[代码](http://pdf7.tarena.com.cn/tts8_source/ttsPage/WEB/WEB_A_V02/JAVASCRIPTCORE/DAY03/EXERCISE/01/index_answer.html)

### 1 下列代码运行运行效果是什么

1. console**.**log**(** Math**.**floor**(** Math**.**random**()\*10** **)** **);**
2. console**.**log**(** Math**.**ceil**(** Math**.**random**()\*10** **)** **);**
3. console**.**log**(** parseInt**(** Math**.**random**()\*40** **)** **+** **60** **);**
4. console**.**log**(** Math**.**ceil**(** **2.1/0.7** **)** **);**

**参考答案**

Math.random()方法用于返回0-1间的随机浮点数num，且0<=num<1；

Math.ceil()方法返回不小于指定参数的最小整数；

Math.floor()方法返回不大于指定参数的最大整数；

parseInt()截取一个字符串或数字的整数部分；

代码行1：返回一个随机整数num，且0<=num<10;

代码行2：返回一个随机整数num，且0<num<=10;

代码行3：返回一个随机整数num，且60<=num<100;

注意代码行4：运行的结果并不是如期望的值3（2.1/0.7=3），而是 4，这是由浮点数的存储精度造成的。计算机使用二进制存储数字，不可能对浮点数表示做到完全精确（即通常会损失一点进度），所以导致 2.1/0.7 的计算结果并不是正好等于 3 ，而是超过 3 一点点（约：3.00000000000000044409）。所以该表达式应用了 Math.ceil() 之后，得到的结果是 4 。

### 2 实现倒计时牌

在页面中实现一个倒计时牌，形如：“距离2017年高考还有XXX天”。

要求：创建一个函数，接收指定的年、月、日作参数，返回当前日期距离指定的日期还差多少天，不足1天算1天，若指定日期晚于当前日期，则返回0。

**参考答案**

实现此案例需要按照如下步骤进行。

**步骤一：创建页面**

首先创建一个纯文本文件，并修改名称为 exercise04\_01.html，然后在文本编辑工具中打开此文件，为其添加 HTML 代码，以创建一个标准结构的 HTML 文档，并设置文档的标题以及编码格式。代码如下所示：

1. **<!**DOCTYPE HTML**>**
2. **<**html**>**
3. **<**head**>**
4. **<**title**>**实现倒计时牌**</**title**>**
5. **<**meta http**-**equiv**=**"content-type" content**=**"text/html;charset=utf-8" **/>**
6. **</**head**>**
7. **<**body**>**
8. **</**body**>**
9. **</**html**>**

**步骤二：创建提示信息，并定义 <script> 代码块**

为页面添加提示信息，并定义 <script> 代码块。代码如下所示：

1. **<!**DOCTYPE HTML**>**
2. **<**html**>**
3. **<**head**>**
4. **<**title**>**实现倒计时牌**</**title**>**
5. **<**meta http**-**equiv**=**"content-type" content**=**"text/html;charset=utf-8" **/>**
6. **<script type=**"text/javascript"**>**
7. **</script>**
8. **</**head**>**
9. **<**body**>**
10. **<h1>**
11. **距离2017年高考还有**
12. **<script type=**"text/javascript"**>**
13. **document.**write**(**getDateSpan**(2017, 6, 7))**
14. **</script>**
15. **天**
16. **</h1>**
17. **</**body**>**
18. **</**html**>**

**步骤三：定义函数 getDateSpan**

定义函数 getDateSpan，计算天数。代码如下所示：

1. **<!**DOCTYPE HTML**>**
2. **<**html**>**
3. **<**head**>**
4. **<**title**>**实现倒计时牌**</**title**>**
5. **<**meta http**-**equiv**=**"content-type" content**=**"text/html;charset=utf-8" **/>**
6. **<**script type**=**"text/javascript"**>**
7. //返回当前日期距离指定的日期还差多少天
8. **function** getDateSpan**(year, month, day) {**
9. **var now = new Date().**getTime**();** //获取当前系统时间的整形表示
10. **var target = new Date(year, month - 1, day).**getTime**();** //目标时间的整形表示
11. //若当前系统时间大于指定的时间，直接返回0
12. **if (now >= target) {**
13. **return 0;**
14. **} else {**
15. **return Math.**ceil**((target - now) / (1000 \* 3600 \* 24));** //不足1天算1天
16. **}**
17. **}**
18. **</**script**>**
19. **</**head**>**
20. **<**body**>**
21. **<**h1**>**
22. 距离**2017**年高考还有
23. **<**script type**=**"text/javascript"**>**
24. document**.**write**(**getDateSpan**(2017,** **6,** **7))**
25. **</**script**>**
26. 天
27. **</**h1**>**
28. **</**body**>**
29. **</**html**>**

文件exercise04\_01.html中的完整代码如下所示：

1. **<!**DOCTYPE HTML**>**
2. **<**html**>**
3. **<**head**>**
4. **<**title**>**实现倒计时牌**</**title**>**
5. **<**meta http**-**equiv**=**"content-type" content**=**"text/html;charset=utf-8" **/>**
6. **<**script type**=**"text/javascript"**>**
7. //返回当前日期距离指定的日期还差多少天
8. **function** getDateSpan**(**year**,** month**,** day**)** **{**
9. **var** now **=** **new** Date**().**getTime**();** //获取当前系统时间的整形表示
10. **var** target **=** **new** Date**(**year**,** month **-** **1,** day**).**getTime**();** //目标时间的整形表示
11. //若当前系统时间大于指定的时间，直接返回0
12. **if** **(**now **>=** target**)** **{**
13. **return** **0;**
14. **}** **else** **{**
15. **return** Math**.**ceil**((**target **-** now**)** **/** **(1000** **\*** **3600** **\*** **24));** //不足1天算1天
16. **}**
17. **}**
18. **</**script**>**
19. **</**head**>**
20. **<**body**>**
21. **<**h1**>**
22. 距离**2017**年高考还有
23. **<**script type**=**"text/javascript"**>**
24. document**.**write**(**getDateSpan**(2017,** **6,** **7))**
25. **</**script**>**
26. 天
27. **</**h1**>**
28. **</**body**>**
29. **</**html**>**

### 3 arguments属性的类型问题

Function对象的arguments属性中保存着函数当前传递进来的实际参数列表，由于实参可能有多个，那么arguments是Array类型的吗？

**参考答案**

虽然arguments属性有length属性，也可以像数组那样使用[index]的格式来获取第index个实参的值，但它并不是Array类型，如:

1. console**.**log**(**arguments **instanceof** Array**);**

将输出false；查看arguments的类型可知其是Object类型的，属于“类数组对象”。

### 4 统计字符串出现的次数(考虑参数异常)

题目基本要求见之前的练习，但要求对函数接收的实际参数进行检验，若不足两个(无法进行统计)，则抛出异常对象及异常提示消息。

**参考答案**

本案例的HTML页面完整代码如下所示：

1. **<!**DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN" "http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd"**>**
2. **<**html xmlns**=**"http://www.w3.org/1999/xhtml"**>**
3. **<**head**>**
4. **<**meta http**-**equiv**=**"Content-Type" content**=**"text/html; charset=utf-8" **/>**
5. **<**title**>**统计字符串出现的次数**</**title**>**
6. **<**script type**=**"text/javascript" src**=**"exercise09\_02.js"**></**script**>**
7. **</**head**>**
8. **<**body**>**
9. **<**p**>**统计下述达内公司接受文字中出现“达内”或者“Tarena”的总次数**(**忽略大小写**)</**p**>**
10. **<**button onclick**=**"console.log( getStringCount(src, '达内','tarena') )"**>**查看字符串出现次数**(**无异常**)</**button**>**
11. **<button onclick=**"console.log( getStringCount(src) )"**>查看字符串出现次数(有异常)</button>**
12. **<**p**>**达内IT培训集团（现已更名达内时代科技集团**-**Tarena Group）是中国高端IT培训机构，致力于培养面向电信和金融等领域的中高端IT人才，达**\**n内由美国国际数据集团IDG、集富亚洲JAFCOASIA和高盛集团共同投资，TARENA是国内首家获得国际风险投资的IT培训机构。**</**p**>**
13. **</**body**>**
14. **</**html**>**

本案例的JavaScript页面完整代码如下所示：

1. //下述字符串中有中文、大小写的英文的达内
2. **var** src**=**'达内IT培训集团（现已更名达内时代科技集团-Tarena Group）是中国高端IT培训机构，致力于培养面向电信和金融等领域的中高端IT人才，达内由美国国际数据集团IDG、集富亚洲JAFCOASIA和高盛集团共同投资，TARENA是国内首家获得国际风险投资的IT培训机构。'**;**
3. //统计一个字符串出现特定字符串的次数
4. **function** getStringCount**(**src**,**search**){**
5. **var** count **=** **0;**        //匹配计数器
7. //若参数个数不足2，则抛出异常对象及提示信息，函数终止执行
8. **if(arguments.length<2){**
9. **throw new Error(**'参数的个数不能少于2'**);**
10. **}**
12. //由于arguments对象不是Array实例，只是类数组对象，
13. //故取出其中欲查找的字符串组成正则表达式样式字符串放入一个字符串数组中
14. **var** argArr **=** **[];**
15. **for(var** i**=1;** i**<**arguments**.**length**;** i**++){**
16. argArr**[**i**-1]** **=** arguments**[**i**];**
17. **}**
18. **var** pattern **=** argArr**.**join**(**'|'**);**
19. **var** regexp **=** **new** RegExp**(**pattern**,** 'ig'**);**
21. //进行正则表达式匹配，返回所有匹配到的字符串
22. **while(** **(**regexp**.**exec**(**src**))!=null** **){**
23. count**++;**
24. **}**
25. **return** count**;**
26. **}**

**1 按钮在不同时间执行不同功能**

假设业务需求中有如下要求：在一个登录表单中有一个登录按钮，在上班时间（早9:00-晚17:00）执行的是员工登录；在非工作时间登录执行的是游客登录。请根据当前系统时间的不同，为同一个按钮的单击时间绑定不同的处理方法(输出不同的提示内容即可)。

**参考答案**

实现此案例需要按照如下步骤进行。

**步骤一：创建页面**

首先创建一个纯文本文件，并修改名称为 exercise05\_01.html，然后在文本编辑工具中打开此文件，为其添加 HTML 代码，以创建一个标准结构的 HTML 文档，并设置文档的标题以及编码格式。代码如下所示：

1. **<!**DOCTYPE html**>**
2. **<**html**>**
3. **<**head**>**
4. **<**meta http**-**equiv**=**"Content-Type" content**=**"text/html; charset=utf-8" **/>**
5. **<**title**>**按钮在不同时间执行不同功能**</**title**>**
6. **</**head**>**
7. **<**body**>**
8. **</**body**>**
9. **</**html**>**

**步骤二：添加提示信息，登录按钮（没有绑定事件处理方法），添加外部js文件引用**

没有在HTML中直接为登录按钮单击事件绑定处理方法，故当页面加载完成后，需要根据当前系统时间条件为按钮绑定不同的单击事件处理方法。

1. **<!**DOCTYPE html**>**
2. **<**html**>**
3. **<**head**>**
4. **<**meta http**-**equiv**=**"Content-Type" content**=**"text/html; charset=utf-8" **/>**
5. **<**title**>**按钮在不同时间执行不同功能**</**title**>**
6. **<script type=**"text/javascript" **src=**"exercise05\_01.js"**></script>**
7. **</**head**>**
8. **<**body onload**=**"bind()"**>**
9. **<p>同一个登录按钮在不同时间执行不同的函数</p>**
10. **<input id=**"btLogin" **type=**"button" **value=**"登录"**/>**
11. **</**body**>**
12. **</**html**>**

**步骤三：创建外部exercise05\_01.js文件，编写函数bind()**

1. //当页面加载完成后，为页面中的按钮绑定处理方法
2. **function** bind**(){**
3. //根据元素的ID值在当前HTML文档中查找唯一的元素
4. **var** bt **=** document**.**getElementById**(**'btLogin'**);**
6. **var** curHour **=** **new** Date**().**getHours**();**    //当前时间中的小时值
7. **if(**curHour**>=9** **&&** curHour**<17){**
8. bt**.**onclick **=** **function(){**    //为按钮的onclick事件绑定处理方式
9. console**.**log**(**'当前执行的是 员工 登录'**);**
10. **};**
11. **}else{**
12. bt**.**onclick **=** **function(){**    //为按钮的onclick事件绑定处理方式
13. console**.**log**(**'当前执行的是 游客 登录'**);**
14. **}**
15. **}**
16. **}**

**2 简述JS中的对象有哪些种类并举例**

JavaScript代码中可以使用的对象有很多，来源不同，也涉及到不同程度的浏览器兼容性问题，请详细的加以区分。

**参考答案**

(1)内置对象：或称为JavaScript原生对象，在ECMAScript标准定义的对象，由浏览器厂家来提供实现，由于标准统一，兼容性比较好。如：Object、Function、Array、String、Boolean、Number、Date、RegExp、Error、EvalError、RangeError、ReferenceError、SyntaxError、TypeError、URIError、Math、Global

(2)宿主对象：由浏览器厂家自主提供的对象，早期兼容性问题比较混乱，当前其中的一些对象在大部分浏览器中的功能大体相当，但仍需要注意细节上的兼容性。宿主对象又分为两种：DOM对象和BOM对象

(3)自定义对象：由用户自定义的对象，兼容性问题需要编写者加以注意。如Emp、Book、News等。

**3 使用面向对象思想进行需求分析与设计**

需求描述：新闻网项目中，有一个新闻列表页面，可以显示出新闻标题、编号、评论数量；还有一个新闻详情页面，可以显示某一条新闻的标题、发布时间、评论数量、来源、详细内容等信息，以及该新闻的所有评论。评论中显示出评论人注册名、评论时间、评论内容等信息。

根据上述需求描述信息，使用面向对象的分析与设计方法，模拟出需要的对象数据及对象间关系。

**参考答案**

根据需求描述，摘取其中的名词作为对象名或者对象的属性名，动词做为对象的方法；可以得到“新闻类型”和“评论类型”，其中每个新闻对象中应该包含编号、标题、来源、发布时间、评论列表等属性；而每个评论对象中应该包含评论人注册名、评论时间、评论内容等属性；注意由于每个新闻对象中都有评论列表属性，故评论数量直接声明为成员方法即可，否则若也声明为属性，就是“冗余属性”了。如图-1所示：

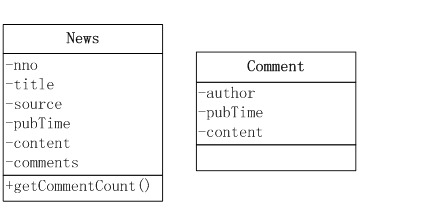


图-1

实现此案例需要按照如下步骤进行。

**步骤一：创建页面**

首先创建一个纯文本文件，并修改名称为 exercise05\_02.html，然后在文本编辑工具中打开此文件，为其添加 HTML 代码，以创建一个标准结构的 HTML 文档，并设置文档的标题以及编码格式。代码如下所示：

1. **<!**DOCTYPE html**>**
2. **<**html**>**
3. **<**head**>**
4. **<**meta http**-**equiv**=**"Content-Type" content**=**"text/html; charset=utf-8" **/>**
5. **<**title**>**使用面向对象思想进行需求分析与设计**</**title**>**
6. **</**head**>**
7. **<**body**>**
8. **</**body**>**
9. **</**html**>**

**步骤二：添加外部js文件引用，及提示文字**

1. **<!**DOCTYPE html**>**
2. **<**html**>**
3. **<**head**>**
4. **<**meta http**-**equiv**=**"Content-Type" content**=**"text/html; charset=utf-8" **/>**
5. **<**title**>**使用面向对象思想进行需求分析与设计**</**title**>**
6. **<script type=**"text/javascript" **src=**"exercise05\_02.js"**></script>**
7. **</**head**>**
8. **<**body**>**
9. **<p>新闻网项目中，有一个新闻列表页面，可以显示出新闻标题、编号、评论数量；还有一个新闻详情页面，可以显示某一条新闻的标题、发布时间、评论数量、来源、详细内容等信息，以及该新闻的所有评论</p>**
10. **</**body**>**
11. **</**html**>**

**步骤三：创建exercise05\_02.js文件，使用面向对象方式创建对象，及其成员属性和成员方法**

1. //创建一个新闻对象，及其所对应的评论对象
2. **var** news0 **=** **new** Object**();** /\*new Object()方式创建一个新对象\*/
3. news0**.**nno **=** **1001;** /\*为对象添加属性\*/
4. news0**.**title **=** '奥巴马蹲地打苍蝇被偷拍'**;**
5. news0**.**pubTime **=** **new** Date**(2014,** **11,** **25,** **10,** **25,** **33);**
6. news0**.**source **=** '新华社'**;**
7. news0**.**content **=** '这张照片是2014年5月6日在白宫椭圆形办公室内拍下的，一直“保密”到现在才“解密”。照片显示，奥巴马发现地上落了一只苍蝇后，立刻悄悄地卷起一本杂志，然后慢慢蹲下，手起书落，精准地打死落在地毯上的苍蝇，让在场的资深总统顾问波德斯达、幕僚长麦唐诺、白宫发言人欧内斯特等人忍不住大笑起来。'**;**
8. news0**.**comments **=** **[**
9. /\*对象直接量方式创建评论对象\*/
10. **{**
11. "author" **:** 'Tom&Jerry'**,**
12. "pubTime" **:** **new** Date**(2014,** **11,** **25,** **13,** **11,** **33),**
13. "content" **:** '白宫安保有漏洞'
14. **},** **{**
15. "author" **:** 'ASDF001'**,**
16. "pubTime" **:** **new** Date**(2014,** **11,** **25,** **18,** **21,** **33),**
17. "content" **:** '这才是真正的奥巴马，一个普通的人，为了美国，尽心尽力。'
18. **},** **{**
19. "author" **:** '隔壁老王'**,**
20. "pubTime" **:** **new** Date**(2014,** **11,** **26,** **09,** **10,** **10),**
21. "content" **:** '率真的形像！'
22. **}** **];**
23. news0**.**getCommentCount **=** **function()** **{** /\*为对象添加方法\*/
24. **return** **this.**comments**.**length**;**
25. **};**

类似的，再创建若干个类似的对象。

1. //创建一个新闻对象，及其所对应的评论对象
2. **var** news0 **=** **new** Object**();** /\*new Object()方式创建一个新对象\*/
3. news0**.**nno **=** **1001;** /\*为对象添加属性\*/
4. news0**.**title **=** '奥巴马蹲地打苍蝇被偷拍'**;**
5. news0**.**pubTime **=** **new** Date**(2014,** **11,** **25,** **10,** **25,** **33);**
6. news0**.**source **=** '新华社'**;**
7. news0**.**content **=** '这张照片是2014年5月6日在白宫椭圆形办公室内拍下的，一直“保密”到现在才“解密”。照片显示，奥巴马发现地上落了一只苍蝇后，立刻悄悄地卷起一本杂志，然后慢慢蹲下，手起书落，精准地打死落在地毯上的苍蝇，让在场的资深总统顾问波德斯达、幕僚长麦唐诺、白宫发言人欧内斯特等人忍不住大笑起来。'**;**
8. news0**.**comments **=** **[**
9. /\*对象直接量方式创建评论对象\*/
10. **{**
11. "author" **:** 'Tom&Jerry'**,**
12. "pubTime" **:** **new** Date**(2014,** **11,** **25,** **13,** **11,** **33),**
13. "content" **:** '白宫安保有漏洞'
14. **},** **{**
15. "author" **:** 'ASDF001'**,**
16. "pubTime" **:** **new** Date**(2014,** **11,** **25,** **18,** **21,** **33),**
17. "content" **:** '这才是真正的奥巴马，一个普通的人，为了美国，尽心尽力。'
18. **},** **{**
19. "author" **:** '隔壁老王'**,**
20. "pubTime" **:** **new** Date**(2014,** **11,** **26,** **09,** **10,** **10),**
21. "content" **:** '率真的形像！'
22. **}** **];**
23. news0**.**getCommentCount **=** **function()** **{** /\*为对象添加方法\*/
24. **return** **this.**comments**.**length**;**
25. **};**
26. //创建一个新闻对象，及其所对应的评论对象
27. **var news1 = new Object();** /\*new Object()方式创建一个新对象\*/
28. **news1.nno = 1015;** /\*为对象添加属性\*/
29. **news1.title =** '委内瑞拉总统今赴中国寻求援助'**;**
30. **news1.pubTime = new Date(2014, 11, 26, 12, 15, 53);**
31. **news1.source =** '21世纪财经网'**;**
32. **news1.content =** '据彭博社消息，委内瑞拉总统尼古拉·马杜罗将于今日前往中国，商谈资金援助事宜，并走访其他欧佩克成员国，以确定油价战略。'**;**
33. **news1.comments = [ {**
34. "author" **:** 'King'**,**
35. "pubTime" **: new Date(2014, 11, 26, 15, 47, 12),**
36. "content" **:** '对违约者要注意，要惩罚。'
37. **}, {**
38. "author" **:** 'DearTang'**,**
39. "pubTime" **: new Date(2014, 11, 26, 18, 57, 09),**
40. "content" **:** '要救，否则以后再南美就没有跳板了。'
41. **} ];**
42. **news1.getCommentCount = function() {** /\*为对象添加方法\*/
43. **return this.comments.length;**
44. **};**
45. //创建一个新闻对象，及其所对应的评论对象
46. **var news2 = new Object();** /\*new Object()方式创建一个新对象\*/
47. **news2.nno = 1026;** /\*为对象添加属性\*/
48. **news2.title =** '解放军挺进风雪高原'**;**
49. **news2.pubTime = new Date(2014, 11, 27, 9, 16, 27);**
50. **news2.source =** '新华社'**;**
51. **news2.content =** '新年伊始，从东北林海雪原到西北风雪达坂，从西藏高原到燕山腹地，冬季实战化训练热潮涌动军营。广大官兵深入学习贯彻习主席重要讲话精神，着眼强军目标，牢固树立战斗力标准，科学利用冬季严寒天气这一难得练兵时机，全方位提高严寒条件下的实战能力。'**;**
52. **news2.comments = [**
53. /\*对象直接量方式创建评论对象\*/
54. **{**
55. "author" **:** 'Tom&Jerry'**,**
56. "pubTime" **: new Date(2014, 11, 27, 13, 52, 33),**
57. "content" **:** '实战练兵，不打无准备的战争。'
58. **}, {**
59. "author" **:** 'DingDong'**,**
60. "pubTime" **: new Date(2014, 11, 27, 18, 16, 33),**
61. "content" **:** '最烦看八股文了'
62. **}, {**
63. "author" **:** '小明'**,**
64. "pubTime" **: new Date(2014, 11, 28, 09, 49, 10),**
65. "content" **:** '牛'
66. **} ];**
67. **news2.getCommentCount = function() {** /\*为对象添加方法\*/
68. **return this.comments.length;**
69. **};**

接下来，为了便于遍历所有新闻对象，把这些新闻对象封装入一个数组中

1. //将所有的新闻对象封装入一个数组
2. **var** newsList **=** **[];**
3. newsList**.**push**(**news0**);**
4. newsList**.**push**(**news1**);**
5. newsList**.**push**(**news2**);**

**步骤四：在exercise05\_02.html中，添加测试代码，以超链接形式列出所有的新闻**

1. **<!**DOCTYPE html**>**
2. **<**html**>**
3. **<**head**>**
4. **<**meta http**-**equiv**=**"Content-Type" content**=**"text/html; charset=utf-8" **/>**
5. **<**title**>**使用面向对象思想进行需求分析与设计**</**title**>**
6. **<**script type**=**"text/javascript" src**=**"exercise07\_02.js"**></**script**>**
7. **</**head**>**
8. **<**body**>**
9. **<**p**>**新闻网项目中，有一个新闻列表页面，可以显示出新闻标题、编号、评论数量；还有一个新闻详情页面，可以显示某一条新闻的标题、发布时间、评论数量、来源、详细内容等信息，以及该新闻的所有评论**</**p**>**
11. **<h2>新闻列表</h2>**
12. **<script type=**"text/javascript"**>**
13. **for(var i=0; i<newsList.length; i++){**
14. **document.**write**(**'<a href="newsdetail.do?nno='**+newsList[i].nno+**'">'**+newsList[i].title+**'</a> '**+newsList[i].**getCommentCount**()+**'<br/>'**);**
15. **}**
16. **</script>**
17. **</**body**>**
18. **</**html**>**

继续添加测试代码，显示某一个新闻对象的所有数据

1. **<!**DOCTYPE html**>**
2. **<**html**>**
3. **<**head**>**
4. **<**meta http**-**equiv**=**"Content-Type" content**=**"text/html; charset=utf-8" **/>**
5. **<**title**>**使用面向对象思想进行需求分析与设计**</**title**>**
6. **<**script type**=**"text/javascript" src**=**"exercise05\_02.js"**></**script**>**
7. **</**head**>**
8. **<**body**>**
9. **<**p**>**新闻网项目中，有一个新闻列表页面，可以显示出新闻标题、编号、评论数量；还有一个新闻详情页面，可以显示某一条新闻的标题、发布时间、评论数量、来源、详细内容等信息，以及该新闻的所有评论**</**p**>**
11. **<**h2**>**新闻列表**</**h2**>**
12. **<**script type**=**"text/javascript"**>**
13. **for(var** i**=0;** i**<**newsList**.**length**;** i**++){**
14. document**.**write**(**'<a href="newsdetail.do?nno='**+**newsList**[**i**].**nno**+**'">'**+**newsList**[**i**].**title**+**'</a> '**+**newsList**[**i**].**getCommentCount**()+**'<br/>'**);**
15. **}**
16. **</**script**>**
18. **<**h2**>**新闻详情**</**h2**>**
19. **<table width=**"80%" **cellspacing=**"5px"**>**
20. **<tr>**
21. **<td colspan=**"3"**><script>document.**write**(newsList[0].title)<**/script></**td>**
22. **</tr>**
23. **<tr>**
24. **<td>来源：<script>document.**write**(newsList[0].source)<**/script></**td>**
25. **<td>时间：<script>document.**write**(newsList[0].pubTime)<**/script></**td>**
26. **<td>评论：<script>document.**write**(newsList[0].**getCommentCount**())<**/script></**td>**
27. **</tr>**
28. **<tr>**
29. **<td colspan=**"3"**><script>document.**write**(newsList[0].content)<**/script></**td>**
30. **</tr>**
31. **</table>**
32. **</**body**>**
33. **</**html**>**

Exercise05\_02.html文件中的完整代码如下所示：

1. **<!**DOCTYPE html**>**
2. **<**html**>**
3. **<**head**>**
4. **<**meta http**-**equiv**=**"Content-Type" content**=**"text/html; charset=utf-8" **/>**
5. **<**title**>**使用面向对象思想进行需求分析与设计**</**title**>**
6. **<**script type**=**"text/javascript" src**=**"exercise05\_02.js"**></**script**>**
7. **</**head**>**
8. **<**body**>**
9. **<**p**>**新闻网项目中，有一个新闻列表页面，可以显示出新闻标题、编号、评论数量；还有一个新闻详情页面，可以显示某一条新闻的标题、发布时间、评论数量、来源、详细内容等信息，以及该新闻的所有评论**</**p**>**
11. **<**h2**>**新闻列表**</**h2**>**
12. **<**script type**=**"text/javascript"**>**
13. **for(var** i**=0;** i**<**newsList**.**length**;** i**++){**
14. document**.**write**(**'<a href="newsdetail.do?nno='**+**newsList**[**i**].**nno**+**'">'**+**newsList**[**i**].**title**+**'</a> '**+**newsList**[**i**].**getCommentCount**()+**'<br/>'**);**
15. **}**
16. **</**script**>**
18. **<**h2**>**新闻详情**</**h2**>**
19. **<**table width**=**"80%" cellspacing**=**"5px"**>**
20. **<**tr**>**
21. **<**td colspan**=**"3"**><**script**>**document**.**write**(**newsList**[0].**title**)<**/script></td**>**
22. **</**tr**>**
23. **<**tr**>**
24. **<**td**>**来源：**<**script**>**document**.**write**(**newsList**[0].**source**)<**/script></td**>**
25. **<**td**>**时间：**<**script**>**document**.**write**(**newsList**[0].**pubTime**)<**/script></td**>**
26. **<**td**>**评论：**<**script**>**document**.**write**(**newsList**[0].**getCommentCount**())<**/script></td**>**
27. **</**tr**>**
28. **<**tr**>**
29. **<**td colspan**=**"3"**><**script**>**document**.**write**(**newsList**[0].**content**)<**/script></td**>**
30. **</**tr**>**
31. **</**table**>**
32. **</**body**>**
33. **</**html**>**

Exercise05\_02.js文件中的完整代码如下所示：

1. //创建一个新闻对象，及其所对应的评论对象
2. **var** news0 **=** **new** Object**();** /\*new Object()方式创建一个新对象\*/
3. news0**.**nno **=** **1001;** /\*为对象添加属性\*/
4. news0**.**title **=** '奥巴马蹲地打苍蝇被偷拍'**;**
5. news0**.**pubTime **=** **new** Date**(2014,** **11,** **25,** **10,** **25,** **33);**
6. news0**.**source **=** '新华社'**;**
7. news0**.**content **=** '这张照片是2014年5月6日在白宫椭圆形办公室内拍下的，一直“保密”到现在才“解密”。照片显示，奥巴马发现地上落了一只苍蝇后，立刻悄悄地卷起一本杂志，然后慢慢蹲下，手起书落，精准地打死落在地毯上的苍蝇，让在场的资深总统顾问波德斯达、幕僚长麦唐诺、白宫发言人欧内斯特等人忍不住大笑起来。'**;**
8. news0**.**comments **=** **[**
9. /\*对象直接量方式创建评论对象\*/
10. **{**
11. "author" **:** 'Tom&Jerry'**,**
12. "pubTime" **:** **new** Date**(2014,** **11,** **25,** **13,** **11,** **33),**
13. "content" **:** '白宫安保有漏洞'
14. **},** **{**
15. "author" **:** 'ASDF001'**,**
16. "pubTime" **:** **new** Date**(2014,** **11,** **25,** **18,** **21,** **33),**
17. "content" **:** '这才是真正的奥巴马，一个普通的人，为了美国，尽心尽力。'
18. **},** **{**
19. "author" **:** '隔壁老王'**,**
20. "pubTime" **:** **new** Date**(2014,** **11,** **26,** **09,** **10,** **10),**
21. "content" **:** '率真的形像！'
22. **}** **];**
23. news0**.**getCommentCount **=** **function()** **{** /\*为对象添加方法\*/
24. **return** **this.**comments**.**length**;**
25. **};**
26. //创建一个新闻对象，及其所对应的评论对象
27. **var** news1 **=** **new** Object**();** /\*new Object()方式创建一个新对象\*/
28. news1**.**nno **=** **1015;** /\*为对象添加属性\*/
29. news1**.**title **=** '委内瑞拉总统今赴中国寻求援助'**;**
30. news1**.**pubTime **=** **new** Date**(2014,** **11,** **26,** **12,** **15,** **53);**
31. news1**.**source **=** '21世纪财经网'**;**
32. news1**.**content **=** '据彭博社消息，委内瑞拉总统尼古拉·马杜罗将于今日前往中国，商谈资金援助事宜，并走访其他欧佩克成员国，以确定油价战略。'**;**
33. news1**.**comments **=** **[** **{**
34. "author" **:** 'King'**,**
35. "pubTime" **:** **new** Date**(2014,** **11,** **26,** **15,** **47,** **12),**
36. "content" **:** '对违约者要注意，要惩罚。'
37. **},** **{**
38. "author" **:** 'DearTang'**,**
39. "pubTime" **:** **new** Date**(2014,** **11,** **26,** **18,** **57,** **09),**
40. "content" **:** '要救，否则以后再南美就没有跳板了。'
41. **}** **];**
42. news1**.**getCommentCount **=** **function()** **{** /\*为对象添加方法\*/
43. **return** **this.**comments**.**length**;**
44. **};**
45. //创建一个新闻对象，及其所对应的评论对象
46. **var** news2 **=** **new** Object**();** /\*new Object()方式创建一个新对象\*/
47. news2**.**nno **=** **1026;** /\*为对象添加属性\*/
48. news2**.**title **=** '解放军挺进风雪高原'**;**
49. news2**.**pubTime **=** **new** Date**(2014,** **11,** **27,** **9,** **16,** **27);**
50. news2**.**source **=** '新华社'**;**
51. news2**.**content **=** '新年伊始，从东北林海雪原到西北风雪达坂，从西藏高原到燕山腹地，冬季实战化训练热潮涌动军营。广大官兵深入学习贯彻习主席重要讲话精神，着眼强军目标，牢固树立战斗力标准，科学利用冬季严寒天气这一难得练兵时机，全方位提高严寒条件下的实战能力。'**;**
52. news2**.**comments **=** **[**
53. /\*对象直接量方式创建评论对象\*/
54. **{**
55. "author" **:** 'Tom&Jerry'**,**
56. "pubTime" **:** **new** Date**(2014,** **11,** **27,** **13,** **52,** **33),**
57. "content" **:** '实战练兵，不打无准备的战争。'
58. **},** **{**
59. "author" **:** 'DingDong'**,**
60. "pubTime" **:** **new** Date**(2014,** **11,** **27,** **18,** **16,** **33),**
61. "content" **:** '最烦看八股文了'
62. **},** **{**
63. "author" **:** '小明'**,**
64. "pubTime" **:** **new** Date**(2014,** **11,** **28,** **09,** **49,** **10),**
65. "content" **:** '牛'
66. **}** **];**
67. news2**.**getCommentCount **=** **function()** **{** /\*为对象添加方法\*/
68. **return** **this.**comments**.**length**;**
69. **};**
70. //将所有的新闻对象封装入一个数组
71. **var** newsList **=** **[];**
72. newsList**.**push**(**news0**);**
73. newsList**.**push**(**news1**);**
74. newsList**.**push**(**news2**);**

**1 使用模板定义对象**

需求描述：网上书城项目中，在书籍详情页面，需要展示如图-1所示的信息：

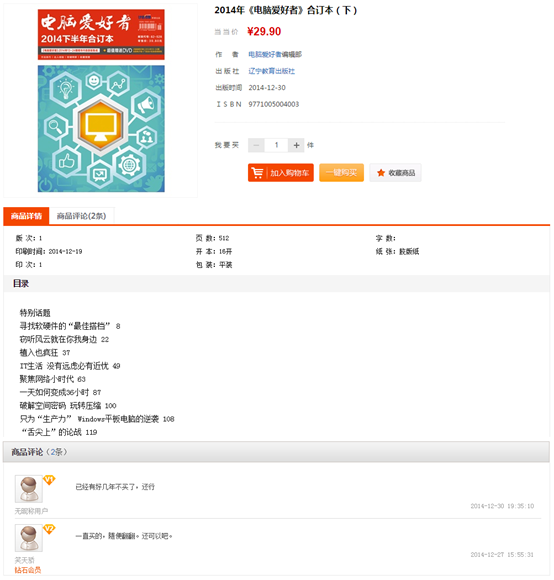


图-1

请使用模板方式定义书籍对象，及其所关联的对象。

**参考答案**

根据需求描述，书籍对象中有些属性是必需的，如书名、价格、作者、出版社、ISBN号、出版日期、内容目录等；此外还有些非必需或者不是那么重要的属性，如版次、页数、字数、开本等。上述需求原型中还涉及到了评论对象（内容和发布时间），以及用户对象(用户编号、用户名、头像、用户等级等)。

综上，本案例至少需要三类对象：书籍、用户、评论，其中书籍对象与评论对象形成一对多关联(一本书籍可以有多个评论，但一个评论只能针对某本书籍)，用户对象与评论对象形成一对多关联（一个用户可以发表多个评论，但一篇评论只能由一个用户发起）。

实现此案例需要按照如下步骤进行。

**步骤一：创建页面**

首先创建一个纯文本文件，并修改名称为 exercise06\_01.html，然后在文本编辑工具中打开此文件，为其添加 HTML 代码，以创建一个标准结构的 HTML 文档，并设置文档的标题以及编码格式。代码如下所示：

1. **<!**DOCTYPE html**>**
2. **<**html**>**
3. **<**head**>**
4. **<**meta http**-**equiv**=**"Content-Type" content**=**"text/html; charset=utf-8" **/>**
5. **<**title**>**使用模板定义对象**</**title**>**
6. **</**head**>**
7. **<**body**>**
8. **</**body**>**
9. **</**html**>**

**步骤二：添加外部js文件引用，及提示文字**

1. **<!**DOCTYPE html**>**
2. **<**html**>**
3. **<**head**>**
4. **<**meta http**-**equiv**=**"Content-Type" content**=**"text/html; charset=utf-8" **/>**
5. **<**title**>**使用模板定义对象**</**title**>**
6. **<script type=**"text/javascript" **src=**"exercise06\_01.js"**></script>**
7. **</**head**>**
8. **<**body**>**
9. **<p>使用模板定义书籍对象及其所关联的评论对象</p>**
11. **<h2>书籍详情</h2>**
12. **</**body**>**
13. **</**html**>**

**步骤三：创建exercise06\_01.js文件，定义对象模板，创建对象**

首先创建Book对象模板和Book对象：

1. //定义书籍对象模板
2. **function** Book**(**bno**,** bname**,** price**,** author**,** pubDate**,** isbn**,** content**){**
3. **this.**bno **=** bno**;**
4. **this.**bname **=** bname**;**
5. **this.**price **=** price**;**
6. **this.**author **=** author**;**
7. **this.**pubDate **=** pubDate**;**
8. **this.**isbn **=** isbn**;**
9. **this.**content **=** content**;**
10. **this.**getCommentCount **=** **function(){**
11. **if(this.**comments**){**
12. **return** **this.**comments**.**length**;**
13. **}else{**
14. **return** **0;**
15. **}**
16. **}**
17. **}**
18. //创建书籍对象
19. **var** b1002 **=** **new** Book**(1002,** '2014年《电脑爱好者》合订本（下）'**,** **29.9,** '电脑爱好者编辑部'**,** **new** Date**(2014,11,30),** '9771005004003'**,** '特别话题<br/>寻找软硬件的“最佳搭档”    8<br/>窃听风云就在你我身边    22<br/>植入也疯狂    37<br/>IT生活 没有远虑必有近忧    49<br/>聚焦网络小时代    63<br/>一天如何变成36小时    87'**);**
20. b1002**.**banCi **=** **1;**
21. b1002**.**yeShu **=** **512;**
22. b1002**.**yinShuaShiJian **=** **new** Date**(2014,11,19);**
23. b1002**.**kaiBen **=** '16开'**;**
24. b1002**.**zhiZhang **=** '胶版纸'**;**
25. b1002**.**yinCi **=** **1;**
26. b1002**.**baoZhuang **=** '平装'**;**

继续创建User模板，及User对象：

1. //定义用户对象模板
2. **function** User**(uno, uname, headPic, level){**
3. **this.uno = uno;**
4. **this.uname = uname;**
5. **this.headPic = headPic;**
6. **this.level = level;**
7. **}**
8. //创建用户对象
9. **var user1 = new** User**(1,** '无昵称用户'**,** 'img/defaulthp.jpg'**, 1);**
10. **var user2 = new** User**(2,** '笑天骄'**,** 'img/defaulthp.jpg'**, 2);**

再创建Comment模板及Comment对象：

1. //定义评论对象模板
2. **function** Comment**(**cno**,** user**,** content**,** pubTime**){**
3. **this.**cno **=** cno**;**
4. **this.**user **=** user**;**
5. **this.**content **=** content**;**
6. **this.**pubTime **=** pubTime**;**
7. **}**
8. //创建评论对象——每一个评论对象都要关联一个用户对象
9. **var** comment1 **=** **new** Comment**(1,** user1**,** '已经有好几年不买了，还行'**,** **new** Date**(2014,11,30,19,35,10));**
10. **var** comment2 **=** **new** Comment**(2,** user2**,** '一直买的，随便翻翻。还可以吧。'**,** **new** Date**(2014,11,27,15,55,31));**

最后，指定Book对象与Comment对象间的关联关系：

1. //创建书籍对象与评论对象的关联关系
2. b1002**.**comments **=** **[];**
3. b1002**.**comments**.**push**(**comment1**);**
4. b1002**.**comments**.**push**(**comment2**);**

**步骤四：在exercise06\_01.html中，添加测试代码，输出书籍对象**

正常页面中，应该将Book对象中的所有数据输出在页面中供客户端浏览，此案例中为了突出练习重点，省略了此步骤。

1. **<!**DOCTYPE html**>**
2. **<**html**>**
3. **<**head**>**
4. **<**meta http**-**equiv**=**"Content-Type" content**=**"text/html; charset=utf-8" **/>**
5. **<**title**>**使用模板定义对象**</**title**>**
6. **<**script type**=**"text/javascript" src**=**"exercise06\_01.js"**></**script**>**
7. **</**head**>**
8. **<**body**>**
9. **<**p**>**使用模板定义书籍对象及其所关联的评论对象**</**p**>**
11. **<**h2**>**书籍详情**</**h2**>**
12. **<script type=**"text/javascript"**>**
13. **console.**log**( b1002 );**//可以在控制台中查看该Book对象的成员
14. **console.**log**( b1002.**getCommentCount**() );**
15. //此处省略了JavaScript数据在页面中的显示
16. **</script>**
17. **</**body**>**
18. **</**html**>**

Exercise06\_01.js文件中的完整代码如下所示：

1. //定义书籍对象模板
2. **function** Book**(**bno**,** bname**,** price**,** author**,** pubDate**,** isbn**,** content**){**
3. **this.**bno **=** bno**;**
4. **this.**bname **=** bname**;**
5. **this.**price **=** price**;**
6. **this.**author **=** author**;**
7. **this.**pubDate **=** pubDate**;**
8. **this.**isbn **=** isbn**;**
9. **this.**content **=** content**;**
10. **this.**getCommentCount **=** **function(){**
11. **if(this.**comments**){**
12. **return** **this.**comments**.**length**;**
13. **}else{**
14. **return** **0;**
15. **}**
16. **}**
17. **}**
18. //创建书籍对象
19. **var** b1002 **=** **new** Book**(1002,** '2014年《电脑爱好者》合订本（下）'**,** **29.9,** '电脑爱好者编辑部'**,** **new** Date**(2014,11,30),** '9771005004003'**,** '特别话题<br/>寻找软硬件的“最佳搭档”    8<br/>窃听风云就在你我身边    22<br/>植入也疯狂    37<br/>IT生活 没有远虑必有近忧    49<br/>聚焦网络小时代    63<br/>一天如何变成36小时    87'**);**
20. b1002**.**banCi **=** **1;**
21. b1002**.**yeShu **=** **512;**
22. b1002**.**yinShuaShiJian **=** **new** Date**(2014,11,19);**
23. b1002**.**kaiBen **=** '16开'**;**
24. b1002**.**zhiZhang **=** '胶版纸'**;**
25. b1002**.**yinCi **=** **1;**
26. b1002**.**baoZhuang **=** '平装'**;**
27. //定义用户对象模板
28. **function** User**(**uno**,** uname**,** headPic**,** level**){**
29. **this.**uno **=** uno**;**
30. **this.**uname **=** uname**;**
31. **this.**headPic **=** headPic**;**
32. **this.**level **=** level**;**
33. **}**
34. //创建用户对象
35. **var** user1 **=** **new** User**(1,** '无昵称用户'**,** 'img/defaulthp.jpg'**,** **1);**
36. **var** user2 **=** **new** User**(2,** '笑天骄'**,** 'img/defaulthp.jpg'**,** **2);**
37. //定义评论对象模板
38. **function** Comment**(**cno**,** user**,** content**,** pubTime**){**
39. **this.**cno **=** cno**;**
40. **this.**user **=** user**;**
41. **this.**content **=** content**;**
42. **this.**pubTime **=** pubTime**;**
43. **}**
44. //创建评论对象——每一个评论对象都要关联一个用户对象
45. **var** comment1 **=** **new** Comment**(1,** user1**,** '已经有好几年不买了，还行'**,** **new** Date**(2014,11,30,19,35,10));**
46. **var** comment2 **=** **new** Comment**(2,** user2**,** '一直买的，随便翻翻。还可以吧。'**,** **new** Date**(2014,11,27,15,55,31));**
47. //创建书籍对象与评论对象的关联关系
48. b1002**.**comments **=** **[];**
49. b1002**.**comments**.**push**(**comment1**);**
50. b1002**.**comments**.**push**(**comment2**);**

**2 为Date对象添加格式化方法和解析方法**

为Date类型添加一个解析方法（命名为parse2()，以与标准的Date.parse()方法区分），可以将任意一个形如“yyyy-MM-dd HH:mm:ss”或者“yyyy-MM-dd”格式的字符串解析为一个Date对象，解决标准的Date.parse()方法的兼容性问题；

为页面中所有的Date对象添加一个format方法，可以将当前对象的属性值格式化为形如"yyyy-MM-dd HH:mm:ss"的字符串，注意补零问题，如1月应显示为01月，9点应显示为09点。

**参考答案**

为Date类型添加parse2()方法，使得可以使用类似Date.parse2(str)的形式(静态方法调用形式)将一个字符串解析为Date对象，直接为Date类型添加parse2()函数声明即可。

为每个Date对象添加一个format方法实现时间对象的格式化，需要使用Date的prototype属性，为Date的原型对象添加新的format()方法。

具体步骤如下所示：

**步骤一：创建页面**

首先创建一个纯文本文件，并修改名称为 exercise06\_02.html，然后在文本编辑工具中打开此文件，为其添加 HTML 代码，以创建一个标准结构的 HTML 文档，并设置文档的标题以及编码格式。代码如下所示：

1. **<!**DOCTYPE html**>**
2. **<**html**>**
3. **<**head**>**
4. **<**meta http**-**equiv**=**"Content-Type" content**=**"text/html; charset=utf-8" **/>**
5. **<**title**>**为Date对象添加格式化方法和解析方法**</**title**>**
6. **</**head**>**
7. **<**body**>**
8. **</**body**>**
9. **</**html**>**

**步骤二：添加测试按钮，以及外部JavaScript文件引用**

1. **<!**DOCTYPE html**>**
2. **<**html**>**
3. **<**head**>**
4. **<**meta http**-**equiv**=**"Content-Type" content**=**"text/html; charset=utf-8" **/>**
5. **<**title**>**为Date对象添加格式化方法和解析方法**</**title**>**
6. **<script type=**"text/javascript" **src=**"exercise06\_02.js"**></script>**
7. **</**head**>**
8. **<**body**>**
9. **<p>为Date对象添加格式化方法和解析方法</p>**
11. **<button onclick=**"console.log( Date.parse2('2014-12-25') )"**>测试Date.**parse2**()方法解析年月日</button>**
12. **<button onclick=**"console.log( Date.parse2('2014-12-25 8:9:10') )"**>测试Date.**parse2**()方法解析年月日时分秒</button>**
13. **<button onclick=**"console.log( new Date().format() )"**>测试dateObject.**format**()方法</button>**
14. **</**body**>**
15. **</**html**>**

**步骤三：创建外部exercise06\_02.js文件**

首先为Date类型添加静态方法parse()：

1. //为Date添加parse(str)方法
2. //str: 要求待解析的字符串形如“yyyy-MM-dd HH:mm:ss”或者“yyyy-MM-dd”
3. Date**.**parse2 **=** **function(**str**)** **{**
4. // 使用连续的空白字符对原字符串进行拆分
5. **var** arr **=** str**.**split**(**/\s+/**);**
6. // arr.length为1说明没有指定时间部分；为2说明指定了日期和时间两部分；否则就是格式错误
7. **if** **(**arr**.**length **!=** **1** **&&** arr**.**length **!=** **2)** **{** // 简单的格式检验
8. **throw** **new** Error**(**'格式错误:' **+** str**);**
9. **}**
10. **var** result **=** **new** Date**();** // 待返回的结果Date对象
11. // 保存日期中的“年-月-日”部分
12. **var** dateArr **=** arr**[0].**split**(**'-'**);**
13. **if** **(**dateArr**.**length **!=** **3)** **{**
14. **throw** **new** Error**(**'日期格式错误：' **+** str**);**
15. **}**
16. result**.**setFullYear**(**parseInt**(**dateArr**[0]),** parseInt**(**dateArr**[1])** **-** **1,**
17. parseInt**(**dateArr**[2]));**
18. // 保存日期中的“时:分:秒”部分
19. **if** **(**arr**[1])** **{**
20. **var** timeArr **=** arr**[1].**split**(**':'**);**
21. **if** **(**timeArr**.**length **!=** **3)** **{**
22. **throw** **new** Erro**(**'时间格式错误：' **+** str**);**
23. **}**
24. result**.**setHours**(**parseInt**(**timeArr**[0]));**
25. result**.**setMinutes**(**parseInt**(**timeArr**[1]));**
26. result**.**setSeconds**(**parseInt**(**timeArr**[2]));**
27. **}** **else** **{** // 若不指定此else，时分秒部分会被初始化为当前系统时间
28. result**.**setHours**(0);**
29. result**.**setMinutes**(0);**
30. result**.**setSeconds**(0);**
31. **}**
32. **return** result**;**
33. **}**

接下来，为所有的Date对象添加用于格式化的format()方法，需要使用到Date类型的原型属性：

1. // 为所有的Date对象添加format()方法
2. Date**.prototype.**format **=** **function()** **{**
3. **var** year **=** **this.**getFullYear**();**
4. year **=** year**<10** **?** '0'**+**year **:** year**;**
5. **var** month **=** **this.**getMonth**()+1;**
6. month **=** month**<10** **?** '0'**+**month **:** month**;**
7. **var** date **=** **this.**getDate**();**
8. date **=** date**<10** **?** '0'**+**date **:** date**;**
9. **var** hour **=** **this.**getHours**();**
10. hour **=** hour**<10** **?** '0'**+**hour **:** hour**;**
11. **var** minute **=** **this.**getMinutes**();**
12. minute **=** minute**<10** **?** '0'**+**minute **:** minute**;**
13. **var** second **=** **this.**getSeconds**();**
14. second **=** second**<10** **?** '0'**+**second **:** second**;**
15. **return** year **+** "-" **+** month **+** "-" **+** date
16. **+** " " **+** hour **+** ":" **+** minute **+** ":" **+** second**;**
17. **}**

文件exercise06\_02.js的完整内容如下：

1. //为Date添加parse(str)方法
2. //str: 要求待解析的字符串形如“yyyy-MM-dd HH:mm:ss”或者“yyyy-MM-dd”
3. Date**.**parse2 **=** **function(**str**)** **{**
4. // 使用连续的空白字符对原字符串进行拆分
5. **var** arr **=** str**.**split**(**/\s+/**);**
6. // arr.length为1说明没有指定时间部分；为2说明指定了日期和时间两部分；否则就是格式错误
7. **if** **(**arr**.**length **!=** **1** **&&** arr**.**length **!=** **2)** **{** // 简单的格式检验
8. **throw** **new** Error**(**'格式错误:' **+** str**);**
9. **}**
10. **var** result **=** **new** Date**();** // 待返回的结果Date对象
11. // 保存日期中的“年-月-日”部分
12. **var** dateArr **=** arr**[0].**split**(**'-'**);**
13. **if** **(**dateArr**.**length **!=** **3)** **{**
14. **throw** **new** Error**(**'日期格式错误：' **+** str**);**
15. **}**
16. result**.**setFullYear**(**parseInt**(**dateArr**[0]),** parseInt**(**dateArr**[1])** **-** **1,**
17. parseInt**(**dateArr**[2]));**
18. // 保存日期中的“时:分:秒”部分
19. **if** **(**arr**[1])** **{**
20. **var** timeArr **=** arr**[1].**split**(**':'**);**
21. **if** **(**timeArr**.**length **!=** **3)** **{**
22. **throw** **new** Erro**(**'时间格式错误：' **+** str**);**
23. **}**
24. result**.**setHours**(**parseInt**(**timeArr**[0]));**
25. result**.**setMinutes**(**parseInt**(**timeArr**[1]));**
26. result**.**setSeconds**(**parseInt**(**timeArr**[2]));**
27. **}** **else** **{** // 若不指定此else，时分秒部分会被初始化为当前系统时间
28. result**.**setHours**(0);**
29. result**.**setMinutes**(0);**
30. result**.**setSeconds**(0);**
31. **}**
32. **return** result**;**
33. **}**
34. // 为所有的Date对象添加format()方法
35. Date**.prototype.**format **=** **function()** **{**
36. **var** year **=** **this.**getFullYear**();**
37. year **=** year**<10** **?** '0'**+**year **:** year**;**
38. **var** month **=** **this.**getMonth**()+1;**
39. month **=** month**<10** **?** '0'**+**month **:** month**;**
40. **var** date **=** **this.**getDate**();**
41. date **=** date**<10** **?** '0'**+**date **:** date**;**
42. **var** hour **=** **this.**getHours**();**
43. hour **=** hour**<10** **?** '0'**+**hour **:** hour**;**
44. **var** minute **=** **this.**getMinutes**();**
45. minute **=** minute**<10** **?** '0'**+**minute **:** minute**;**
46. **var** second **=** **this.**getSeconds**();**
47. second **=** second**<10** **?** '0'**+**second **:** second**;**
48. **return** year **+** "-" **+** month **+** "-" **+** date
49. **+** " " **+** hour **+** ":" **+** minute **+** ":" **+** second**;**
50. **}**

**1 数组去重复的各种方法总结与性能分析**

需求描述：为Array类型添加公共方法Unique，去掉给定数组中的重复元素。

**参考答案**

方法一：创建另一个空数组。然后依次遍历原数组中每个元素。如果新数组中不包含正在遍历的元素，就将正在遍历的当前元素加入新数组中。

**步骤一：创建页面**

首先创建一个exercise07\_01.html，然后在文本编辑工具中打开此文件，为其添加 HTML 代码，以创建一个标准结构的 HTML 文档（使用过渡型文档类型声明），并设置文档的标题以及编码格式。代码如下所示：

1. **<!**DOCTYPE html**>**
2. **<**html**>**
3. **<**head**>**
4. **<**meta http**-**equiv**=**"Content-Type" content**=**"text/html; charset=utf-8" **/>**
5. **<**title**>**数组去重复的各种方法总结与性能分析**</**title**>**
6. **</**head**>**
7. **<**body**>**
8. **</**body**>**
9. **</**html**>**

**步骤二：添加外部js文件引用，同时为页面添加一个div用作输出结果**

1. **<!**DOCTYPE html**>**
2. **<**html**>**
3. **<**head**>**
4. **<**meta http**-**equiv**=**"Content-Type" content**=**"text/html; charset=utf-8" **/>**
5. **<**title**>**数组去重复的各种方法总结与性能分析**</**title**>**
6. **<script type=**"text/javascript" **src=**"exercise07\_01.js"**></script>**
7. **</**head**>**
8. **<**body**>**
9. **<div id=**"out"**></div>**
10. **</**body**>**
11. **</**html**>**

**步骤三：创建exercise07\_01.js文件**

1. /\*方法一\*/
2. //为Array类型的prototype对象添加unique1方法
3. Array**.prototype.**unique1**=function(){**
4. //遍历当前数组对象中的每个元素，同时创建新空数组对象，保存在变量n中
5. **for(var** i**=0,**n**=[];**i**<this.**length**;**i**++){**
6. n**.**indexOf**(this[**i**])!=-1&&**n**.**push**(this[**i**]);**
7. **}**
8. **return** n**;**
9. **}**

**步骤四：在exercise07\_01.js中，添加测试代码**

1. /\*测试代码：\*/
2. window**.**onload**=function(){**
3. //获得输出结果的div
4. **var** out**=**document**.**getElementById**(**"out"**);**
5. //随机生成10000个0~10000之间的随机数，放入数组arr中
6. **for(var** i**=0,**arr**=[];**i**<10000;**i**++){**
7. arr**.**push**(**Math**.**floor**(**Math**.**random**()\*10000));**
8. **}**
9. //获得开始时间：
10. **var** start**=new** Date**();**
11. //调用数组类型的去重复方法1，获得新数组保存在变量re中
12. **var** re**=**arr**.**unique1**();**
13. //获得结束时间：
14. **var** end**=new** Date**();**
15. //在控制台输出数组内容：
16. console**.**log**(**re**);**
17. //在输出div中写入方法和耗时
18. out**.**innerHTML**+=**"方法1 耗时 "**+(**end**-**start**)+**" ms<br>"**;**
19. **}**

**步骤五：在exercise07\_01.js文件中，为Array类型的prototype定义第二种数组去重复的方法**

1. /\*方法一\*/
2. //为Array类型的prototype对象添加unique1方法
3. Array**.prototype.**unique1**=function(){**
4. … …
5. **}**
6. /\*方法二\*/
7. //为Array类型的prototype对象添加unique2方法
8. Array**.prototype.**unique2**=function(){**
9. //遍历当前数组中每个元素，同时创建一个hash对象和一个结果数组
10. **for(var** i**=0,**hash**={},**r**=[];**i**<this.**length**;**i**++){**
11. **!**hash**[this[**i**]]&&(**hash**[this[**i**]]=null,**r**.**push**(this[**i**]));**
12. **}**
13. **return** r**;**
14. **}**
15. /\*测试代码：\*/
16. window**.**onload**=function(){**
17. … …
18. **}**

**步骤六：在exercise07\_01.js中,继续添加测试代码**

1. /\*测试代码：\*/
2. window**.**onload**=function(){**
3. … …
4. //在输出div中写入方法和耗时
5. out**.**innerHTML**+=**"方法1 耗时 "**+(**end**-**start**)+**" ms<br>"**;**
6. //获得开始时间：
7. start**=new** Date**();**
8. //调用数组类型的去重复方法2，获得新数组保存在变量re中
9. re**=**arr**.**unique2**();**
10. //获得结束时间：
11. end**=new** Date**();**
12. //在控制台输出数组内容：
13. console**.**log**(**re**);**
14. //在输出div中写入方法和耗时
15. out**.**innerHTML**+=**"方法2 耗时 "**+(**end**-**start**)+**" ms<br>"**;**
16. **}**

执行，查看结果。对比前两种方法：明显第二种方法效率更高。主要原因：数组类型的indexOf方法其实也需要遍历数组中每个元素，才能找到匹配的元素。这就大幅增加了循环和比较的次数。而方法二中利用了JavaScript语言中对象底层其实是hash数组的特性。hash数组中，按照key查找元素，无需遍历。所以自然在循环次数上优于方法一。但是方法二却多创建了一个对象。多占用了空间，却换来了执行效率上的优化，这种现象也称为用空间换时间。

**步骤七：在exercise07\_01.js文件中，为Array类型的prototype定义第三种数组去重复的方法**

1. /\*方法一\*/
2. //为Array类型的prototype对象添加unique1方法
3. Array**.prototype.**unique1**=function(){**
4. … …
5. **}**
6. /\*方法二\*/
7. //为Array类型的prototype对象添加unique2方法
8. Array**.prototype.**unique2**=function(){**
9. … …
10. **}**
11. /\*方法三\*/
12. //为Array类型的prototype对象添加unique3方法
13. Array**.prototype.**unique3**=function(){**
14. //先将数组元素默认按升序排列
15. **this.**sort**();**
16. //遍历数组中每个元素，同时创建结果数组r，先放入原数组第一个元素。
17. **for(var** i**=1,**r**=[this[0]];**i**<this.**length**;**i**++){**
18. //用原数组当前元素和结果数组最后一个元素比较，如果不相等，就将原数组当前元素压入结果数组
19. **this[**i**]!==**r**[**r**.**length**-1]&&**r**.**push**(this[**i**]);**
20. **}**
21. **return** r**;**
22. **};**
23. /\*测试代码：\*/
24. window**.**onload**=function(){**
25. … …
26. **}**

**步骤八：在exercise07\_01.js中,继续添加测试代码**

1. /\*测试代码：\*/
2. window**.**onload**=function(){**
3. … …
4. //在输出div中写入方法和耗时
5. out**.**innerHTML**+=**"方法2 耗时 "**+(**end**-**start**)+**" ms<br>"**;**
6. //获得开始时间：
7. start**=new** Date**();**
8. //调用数组类型的去重复方法3，获得新数组保存在变量re中
9. re**=**arr**.**unique3**();**
10. //获得结束时间：
11. end**=new** Date**();**
12. //在控制台输出数组内容：
13. console**.**log**(**re**);**
14. //在输出div中写入方法和耗时
15. out**.**innerHTML**+=**"方法3 耗时 "**+(**end**-**start**)+**" ms<br>"**;**
16. **}**

执行，观察结果，方法三比方法二效率稍低。因为方法三需要先对数组排序。无论采用何种方法排序，都会增加遍历和比较的次数。但即使如此，方法三远优于方法一。因为数组的sort方法采用的排序算法通常都经过了特别的优化，比indexOf这样的遍历方法效率要高。

**步骤九：在exercise07\_01.js文件中，为Array类型的prototype定义第四种数组去重复的方法**

1. /\*方法一\*/
2. //为Array类型的prototype对象添加unique1方法
3. Array**.prototype.**unique1**=function(){**
4. … …
5. **}**
6. /\*方法二\*/
7. //为Array类型的prototype对象添加unique2方法
8. Array**.prototype.**unique2**=function(){**
9. … …
10. **}**
11. /\*方法三\*/
12. //为Array类型的prototype对象添加unique3方法
13. Array**.prototype.**unique3**=function(){**
14. … …
15. **};**
16. /\*方法四：\*/
17. //为Array类型的prototype对象添加unique4方法
18. Array**.prototype.**unique4**=function(){**
19. //以空数组作为基础，依次迭代处理数组中每个元素。如果当前元素curr在prev数组中未出现过，则将当前元素curr压入prev数组。
20. **return** **this.**reduce**(function(**prev**,** curr**)** **{**
21. prev**.**indexOf**(**curr**)<0&&**prev**.**push**(**curr**);**
22. **return** prev**;**
23. **},** **[]);**
24. **};**
25. /\*测试代码：\*/
26. window**.**onload**=function(){**
27. … …
28. **}**

**步骤十：在exercise07\_01.js中,继续添加测试代码**

1. /\*测试代码：\*/
2. window**.**onload**=function(){**
3. … …
4. //在输出div中写入方法和耗时
5. out**.**innerHTML**+=**"方法3 耗时 "**+(**end**-**start**)+**" ms<br>"**;**
6. //获得开始时间：
7. start**=new** Date**();**
8. //调用数组类型的去重复方法4，获得新数组保存在变量re中
9. re**=**arr**.**unique4**();**
10. //获得结束时间：
11. end**=new** Date**();**
12. //在控制台输出数组内容：
13. console**.**log**(**re**);**
14. //在输出div中写入方法和耗时
15. out**.**innerHTML**+=**"方法4 耗时 "**+(**end**-**start**)+**" ms<br>"**;**
16. **}**

执行，观察结果，方法四的执行效率与方法一相当。因为都采用了indexOf方法，增大了遍历和比较的次数。

**步骤十一：在exercise07\_01.js文件中，为Array类型的prototype定义第五种数组去重复的方法**

1. /\*方法一\*/
2. //为Array类型的prototype对象添加unique1方法
3. Array**.prototype.**unique1**=function(){**
4. … …
5. **}**
6. /\*方法二\*/
7. //为Array类型的prototype对象添加unique2方法
8. Array**.prototype.**unique2**=function(){**
9. … …
10. **}**
11. /\*方法三\*/
12. //为Array类型的prototype对象添加unique3方法
13. Array**.prototype.**unique3**=function(){**
14. … …
15. **};**
16. /\*方法四：\*/
17. //为Array类型的prototype对象添加unique4方法
18. Array**.prototype.**unique4**=function(){**
19. … …
20. **};**
21. /\*方法五：\*/
22. //为Array类型的prototype对象添加unique5方法
23. Array**.prototype.**unique5 **=** **function** **()** **{**
24. //先将数组排序，再用两个逗号拼接数组元素，再用正则表达式将多个逗号分隔的相同的子字符串替换为仅剩一个第一个。最后按两个逗号为分隔，将字符串再切割成数组。
25. **return** **this.**sort**().**join**(**",,"**).**replace**(**/(,|^)([^,]+)(,,\2)+(,|$)/g**,**"$1$2$4"**).**split**(**",,"**);**
26. **}**
27. /\*测试代码：\*/
28. window**.**onload**=function(){**
29. … …
30. **}**

**步骤十二：在exercise07\_01.js中,继续添加测试代码**

1. /\*测试代码：\*/
2. window**.**onload**=function(){**
3. … …
4. //在输出div中写入方法和耗时
5. out**.**innerHTML**+=**"方法4 耗时 "**+(**end**-**start**)+**" ms<br>"**;**
6. //获得开始时间：
7. start**=new** Date**();**
8. //调用数组类型的去重复方法5，获得新数组保存在变量re中
9. re**=**arr**.**unique5**();**
10. //获得结束时间：
11. end**=new** Date**();**
12. //在控制台输出数组内容：
13. console**.**log**(**re**);**
14. //在输出div中写入方法和耗时
15. out**.**innerHTML**+=**"方法5 耗时 "**+(**end**-**start**)+**" ms<br>"**;**
16. **}**

执行，观察结果，方法五的执行效率与方法三相当。因为都采用了sort方法先排序再处理。另外，由此还可看出正则表达式处理字符串的效率和数组遍历的效率几乎相当。可见正则表达式的效率也是非常高的。

结论：查找最快的数据结构是hash数组。经常用于统计元素数量，去除重复元素等需要反复遍历和比较的操作中。而indexOf方法的效率不高，不推荐作为首选方法。

Exercise07\_01.html文件中的完整代码如下所示：

1. **<!**DOCTYPE html**>**
2. **<**html**>**
3. **<**head**>**
4. **<**meta http**-**equiv**=**"Content-Type" content**=**"text/html; charset=utf-8" **/>**
5. **<**title**>**数组去重复的各种方法总结与性能分析**</**title**>**
6. **</**head**>**
7. **<**body**>**
8. **</**body**>**
9. **</**html**>**

Exercise07\_01.js文件中的完整代码如下所示：

1. /\*方法一\*/
2. //为Array类型的prototype对象添加unique1方法
3. Array**.prototype.**unique1**=function(){**
4. //遍历当前数组对象中的每个元素，同时创建新空数组对象，保存在变量n中
5. **for(var** i**=0,**n**=[];**i**<this.**length**;**i**++){**
6. n**.**indexOf**(this[**i**])==-1&&**n**.**push**(this[**i**]);**
7. **}**
8. **return** n**;**
9. **}**
10. /\*方法二\*/
11. //为Array类型的prototype对象添加unique2方法
12. Array**.prototype.**unique2**=function(){**
13. //遍历当前数组中每个元素，同时创建一个hash对象和一个结果数组
14. **for(var** i**=0,**hash**={},**r**=[];**i**<this.**length**;**i**++){**
15. **!**hash**[this[**i**]]&&(**hash**[this[**i**]]=null,**r**.**push**(this[**i**]));**
16. **}**
17. **return** r**;**
18. **}**
19. /\*方法三\*/
20. //为Array类型的prototype对象添加unique3方法
21. Array**.prototype.**unique3**=function(){**
22. //先将数组元素默认按升序排列
23. **this.**sort**();**
24. //遍历数组中每个元素，同时创建结果数组r，先放入原数组第一个元素。
25. **for(var** i**=1,**r**=[this[0]];**i**<this.**length**;**i**++){**
26. //用原数组当前元素和结果数组最后一个元素比较，如果不相等，就将原数组当前元素压入结果数组
27. **this[**i**]!==**r**[**r**.**length**-1]&&**r**.**push**(this[**i**]);**
28. **}**
29. **return** r**;**
30. **};**
31. /\*方法四：\*/
32. //为Array类型的prototype对象添加unique4方法
33. Array**.prototype.**unique4**=function(){**
34. //以空数组作为基础，依次迭代处理数组中每个元素。如果当前元素c在p数组中未出现过，则将当前元素c压入p数组。
35. **return** **this.**reduce**(function(**prev**,** curr**)** **{**
36. prev**.**indexOf**(**curr**)<0&&**prev**.**push**(**curr**);**
37. **return** prev**;**
38. **},** **[]);**
39. **};**
40. /\*方法五：\*/
41. //为Array类型的prototype对象添加unique5方法
42. Array**.prototype.**unique5 **=** **function** **()** **{**
43. //先将数组排序，再用两个逗号拼接数组元素，再用正则表达式将多个逗号分隔的相同的子字符串替换为仅剩一个第一个。最后按两个逗号为分隔，将字符串再切割成数组。
44. **return** **this.**sort**().**join**(**",,"**).**replace**(**/(,|^)([^,]+)(,,\2)+(,|$)/g**,**"$1$2$4"**).**split**(**",,"**);**
45. **}**
46. /\*测试代码：\*/
47. window**.**onload**=function(){**
48. //获得输出结果的div
49. **var** out**=**document**.**getElementById**(**"out"**);**
50. //随机生成10000个0~10000之间的随机数，放入数组arr中
51. **for(var** i**=0,**arr**=[];**i**<10000;**i**++){**
52. arr**.**push**(**Math**.**floor**(**Math**.**random**()\*10000));**
53. **}**
54. //获得开始时间：
55. **var** start**=new** Date**();**
56. //调用数组类型的去重复方法1，获得新数组保存在变量re中
57. **var** re**=**arr**.**unique1**();**
58. //获得结束时间：
59. **var** end**=new** Date**();**
60. //在控制台输出数组内容：
61. console**.**log**(**re**);**
62. //在输出div中写入方法和耗时
63. out**.**innerHTML**+=**"方法1 耗时 "**+(**end**-**start**)+**" ms<br>"**;**
64. //获得开始时间：
65. start**=new** Date**();**
66. //调用数组类型的去重复方法2，获得新数组保存在变量re中
67. re**=**arr**.**unique2**();**
68. //获得结束时间：
69. end**=new** Date**();**
70. //在控制台输出数组内容：
71. console**.**log**(**re**);**
72. //在输出div中写入方法和耗时
73. out**.**innerHTML**+=**"方法2 耗时 "**+(**end**-**start**)+**" ms<br>"**;**
74. //获得开始时间：
75. start**=new** Date**();**
76. //调用数组类型的去重复方法3，获得新数组保存在变量re中
77. re**=**arr**.**unique3**();**
78. //获得结束时间：
79. end**=new** Date**();**
80. //在控制台输出数组内容：
81. console**.**log**(**re**);**
82. //在输出div中写入方法和耗时
83. out**.**innerHTML**+=**"方法3 耗时 "**+(**end**-**start**)+**" ms<br>"**;**
84. //获得开始时间：
85. start**=new** Date**();**
86. //调用数组类型的去重复方法4，获得新数组保存在变量re中
87. re**=**arr**.**unique4**();**
88. //获得结束时间：
89. end**=new** Date**();**
90. //在控制台输出数组内容：
91. console**.**log**(**re**);**
92. //在输出div中写入方法和耗时
93. out**.**innerHTML**+=**"方法4 耗时 "**+(**end**-**start**)+**" ms<br>"**;**
94. //获得开始时间：
95. start**=new** Date**();**
96. //调用数组类型的去重复方法5，获得新数组保存在变量re中
97. re**=**arr**.**unique5**();**
98. //获得结束时间：
99. end**=new** Date**();**
100. //在控制台输出数组内容：
101. console**.**log**(**re**);**
102. //在输出div中写入方法和耗时
103. out**.**innerHTML**+=**"方法5 耗时 "**+(**end**-**start**)+**" ms<br>"**;**
104. **}**

**2 简述call，callee，apply，bind四个方法的区别：**

**参考答案**

call,apply,bind都是函数对象的方法。

call方法：在调用函数时，更换函数执行上下文对象中的this引用的对象地址。

通常用于一个对象的方法，被另一个对象临时借用时使用。

call方法不但可以在调用函数时更换this引用的对象，而且，如果调用的函数如果需要参数，还可以同时传入参数值。

apply方法的作用和call方法完全相同。都是在调用函数时，临时更换执行上下文的中的this引用的对象。唯一的差别就是call方法在传入参数值时，要求每个参数值作为独立的参数传入，用逗号分隔。而apply则要求所有传入的参数值必须放在一个数组中，作为一个整体传入函数。

bind方法的作用是基于现有函数，创建一个新的函数副本。创建新函数副本的同时，为新函数副本绑定this引用的对象。绑定的结果是，新函数无论何时调用，this都引用绑定时指定的对象。属于持久绑定。

bind方法不但可以持久绑定一个对象到函数的this。而且还可以在调用前提前绑定个别参数的值。调用时，已经绑定的参数可不必传入。

bind方法和call以及apply的差别是：

显著差别是：call和apply是调用一个函数对象执行任务，而bind则是创建一个新的函数对象副本。bind方法不负责调用函数。

另外，call和apply方法，仅在调用时才能更换this引用的对象以及传入函数需要的参数值。一旦执行完就释放了。再调用时，还需要重新指定this引用的对象和传入参数。

而bind方法可将对象持久绑定在一个函数对象上。同时提前指定的参数值也可持久绑定在函数对象上。即使反复调用函数对象，this引用的对象和提前绑定的属性值依然与函数对象持久绑定。

bind方法基本实现原理模拟如下：

1. Function**.prototype.**bind **=** **function** **(**scope**)** **{**
2. **var** fn **=** **this;** // this 指向调用bind方法的一个函数对象
3. **return** **function** **()** **{**
4. //使用闭包将原函数对象fn，绑定给this的对象scope和参数对象arguments牢牢保护在新函数对象内。达到持久保存的目的。
5. **return** fn**.**apply**(**scope**,** arguments**);**
6. **}**
7. **};**

需要强调的是,因为闭包的原因，bind返回的函数对象中绑定的this对象的引用，无法用call或apply在调用时临时改变。bind返回的函数对象执行时的this，始终只会引用最初绑定的对象。

Callee是arguments对象的一个属性，引用了arguments所在的函数对象本身。

Callee属性和call以及apply方法的差别是，callee不会修改this的引用。Callee只是简单的引用当前arguments所在的函数对象而已。而call和apply反而不关心函数具体定义，而只负责修改调用函数时的this引用的对象。

Callee属性主要用于实现函数定义与外部的分离。比如实现匿名函数递归调用等操作。

总结：call和apply负责在调用函数时，临时更换函数执行上下文的this引用的对象。bind则专门用于基于现有函数定义，返回一个新函数对象，并提前绑定函数对象执行时的this引用。且是持久绑定，无法被call和apply替换。Callee是arguments的一个属性，与this无关。仅引用arguments所在的函数对象本身。

**3 判断下列代码的输出结果：**

1. **function** Person**(){**
2. **var** prop**=**"私有属性"**;**
3. **this.**prop**=**"公有属性"**;**
4. **var** fun**=function(){**
5. console**.**log**(**"私有方法"**);**
6. console**.**log**(**prop**);**
7. **};**
8. **this.**fun**=function(){**
9. console**.**log**(**"公有方法"**);**
10. fun**();**
11. console**.**log**(**prop**);**
12. console**.**log**(this.**prop**);**
13. **}**
14. **}**
15. **var** lilei**=new** Person**();**
16. lilei**.**fun**();**

**参考答案**

调用lilei的fun方法，首先执行Person类型的公有方法fun。其中第一句话首先输出“公有方法”。第二句话公有方法fun调用了私有方法fun，首先输出“私有方法“。私有方法fun中又输出了私有属性prop的值”私有属性“。

公有方法执行完私有fun方法后，继续输出私有属性的值“私有属性“，然后又输出了当前对象的公有属性prop的值”公有属性“。

所以，输出结果：

公有方法

私有方法

私有属性

私有属性

公有属性