

《微信小程序开发》实验报告三

网页基础—— JavaScript

班级 计科 1902 学号 姓名 电话

1. 实验目的

通过本次实验掌握 JavaScript 的基础知识，能使用 JavaScript 编写程序，进行程序设计。

掌握 JavaScript 和 DOM 及 BOM 网页技术，了解 JavaScript 与网页和浏览器交互的方式，能通过 JavaScript 操作网页弹窗和修改网页元素。

通过实践掌握 JavaScript 在网页设计中的常见应用之一——表单校验：对表单输入值合法性进行相应的校验，并能够输出提示信息。

学会响应式网页设计，通过学习掌握 JavaScript 事件，通过编写事件处理函数对用户的点击、悬浮等事件进行处理，使静态网页能够对用户事件进行响应。

能够编写响应式的 JavaScript 组件模块，通过对 HTML、CSS、JavaScript 的综合运用，编写一个能够独立完成复杂功能的模块。

2. 实验内容及要求

准备工作：为了方便后续的学习和实践，建议同学们自带电脑，在自己电脑上进行实验内容，并完成实验报告。本次实验为基础内容，认真完成本次实验内容即可掌握开发微信小程序的基本工具使用。

（一）实验内容

- 1、学习 JavaScript 的相关基础语法，掌握变量类型、变量声明等的语法格式；
- 2、学习 JavaScript 编写条件、判断、循环语句的方式，能熟练写出相应语句；
- 3、学习 JavaScript 操作 HTML 界面的方式—— DOM 的相关知识；
- 4、学习 JavaScript 操作浏览器窗口的方式 —— BOM 的相关知识；
- 5、使用 JavaScript 完成表单校验的学习和编写
- 6、综合使用 JavaScript 结合 BOM 和 DOM 对页面进行修改
- 7、学习 JavaScript 进行组件化开发，一步一步完成独立的组件

（二）实验步骤（仔细阅读，按照步骤做完）

- 1、通过学习 JavaScript 语法，循序渐进的学习 JavaScript

- (1) 打开 <https://www.w3school.com.cn/js/index.asp> 学习 JavaScript 基础语法；

JavaScript 语法繁多，实际上我们遇到的 80% 的问题只需要 20% 的知识就能够解决了，主要学习的内容包括：

- js 中的主要数据类型
- js 变量的申明及如何赋值
- js 中的运算符
- 函数
- 数字日期字符串
- 选择循环及判断语句的写法等

- (2) 打开 https://www.w3school.com.cn/js/js_htmlDOM.asp 学习 JavaScript HTML DOM，掌握如何通过 JavaScript 访问和改变 HTML 文档的元素和属性

- (3) 打开 https://www.w3school.com.cn/js/js_window.asp 学习 JavaScript Browser BOM，掌握如何通过 JavaScript 使用浏览器与用户对话

2、综合运用 JavaScript 实现对网页内容的修改

- (1) 新建一个 index.html 测试文件，写好文件的基本框架，并添加一个 `<h1>标题</h1>` 标签作为标题
- (2) 在测试文件目录下创建一个名为 scripts 的文件夹，然后在其中创建一个名为 main.js 的文件
- (3) 在 index.html 文件中，`</body>` 标签后添加以下代码：
`<script src="scripts/main.js"></script>`
- (4) 打开 main.js 文件，在其中添加如下内容：
- ```
let myHeading = document.querySelector('h1');

myHeading.textContent = 'Hello world!';
```
- (5) 最后分别将两个文件保存后预览，用浏览器打开 index.html，可以预览到 index.html 已显示了标题，为标题和背景添加样式之后，效果如下所示：



(注：样式请自行修改成自己想要的样子)

### 3、通过 JavaScript 完成对表单的校验功能

上次作业的表单如下图所示：

用户 名：

密 码：  真实姓名：

确认密码：  手机号码：

性 别： ☐男 ☒女

民 族：  邮 箱：

爱 好： ☐篮球 ☐足球 ☐抽烟 (等等自行设计)

国 家：  城 市：

现有新要求如下：

1. (选做) 运用弹性布局(流动布局)使得输入框左边的描述汉字也能够左右对齐
2. 使用 JavaScript 逻辑对表单的输入项进行校验，如：
  - a) 密码和确认密码必须一致
  - b) 手机号码必须为 11 位数字
  - c) 邮箱必须是 xxxxxxxx@xxx 的形式等

如果有不满足要求的情况下进行提交，则浏览器弹框警告错误原因，并提示用户重新输入

### 4、使用 js 完成简单的动态网页效果

- (1) 新建一个 index.html 文件，在该文件 body 中引入一张图片
- (2) 为图片添加内联样式，设置定位方式为绝对定位，宽度设为 200px，高度也设为 200px
- (3) 通过设置图片的定位的 left 和 top 属性，为图片显示位置指定一个初始值

- (4) 通过 JavaScript 使用随机函数为图片设置初始值，每次刷新应看到位置都不同，可参考如下代码：

```
// 获取图片 dom 节点对象
var zju = document.getElementById('zju');
// 生成图片一开始的随机位置
var width = Math.random()*w_width;
var height = Math.random()*w_height;
zju.style.left = width + "px";
zju.style.top = height + 'px';
// 拿到浏览器窗口宽高
const w_width = window.screen.availWidth - 200;
const w_height = window.screen.availHeight - 200;
// 生成图片一开始的随机位置
var width = Math.random()*w_width;
var height = Math.random()*w_height;
zju.style.left = width + "px";
zju.style.top = height + 'px';
```

- (5) 通过使用 setInterval 函数使得图片能够每秒改变一次位置（此题会留为作业）
- (6) 通过使用 setInterval 函数使得图片能够在浏览器窗口中自由漂浮，可以参考以下代码：

```
<script>
// 获取图片dom节点对象
var zju = document.getElementById('zju');
// 拿到浏览器窗口宽高
const w_width = window.screen.availWidth - 200;
const w_height = window.screen.availHeight - 200;
// 生成图片一开始的随机位置
var width = Math.random()*w_width;
var height = Math.random()*w_height;
zju.style.left = width + "px";
zju.style.top = height + 'px';
// 图片移动的速度控制
var vx = 30;
var vy = 30;
// 改变图片位置的函数
function changePos() {
 if ((width + vx) >= w_width || (width + vx) <= 0){
 vx = -vx;
 }
 width = width + vx;
 if ((height + vy) >= w_height || (height + vy) <= 0){
 vy = -vy;
 }
 height = height + vy;
 zju.style.left = width + "px";
 zju.style.top = height + "px";
}
// 使得图片每秒变动一次位置
setInterval(changePos,1000);
</script>
```

## 5、使用 JavaScript 完成一个猜数字游戏编写

最终完成的效果如下图所示：

### 猜数字游戏

我刚才随机选定了一个100以内的自然数。看你能否在 10 次以内猜中它。每次我都会告诉你所猜的结果是高了还是低了。

请猜数:

我们希望它能实现这样的功能：

我想让你开发一个猜数字游戏。游戏应随机选择一个 100 以内的自然数，然后邀请玩家在 10 轮以内猜出这个数字。每轮后都应告知玩家的答案正确与否，如果出错了，则告诉他数字是低了还是高了。并且应显示出玩家前一轮所猜的数字。一旦玩家猜对，或者用尽所有机会，游戏将结束。游戏结束后，可以让玩家选择再次开始。

我们现在一步一步来开始：

- (1) 首先，我们需要有猜数字游戏的界面，于是我们编好相应的 html 代码（你可以参考我下方的代码）

```
<body>
<h1>猜数字游戏</h1>

<p>我刚才随机选定了一个100以内的自然数。看你能否在 10 次以内猜中它。每次我都会告诉你所猜的结果是高了还是低了。</p>

<div class="form">
 <label for="guessField">请猜数: </label>
 <input type="text" id="guessField" class="guessField">
 <input type="submit" value="确定" id="guessSubmit" class="guessSubmit">
</div>

<div class="resultParas">
 <p id="guesses" class="guesses"></p>
 <p id="lastResult" class="lastResult"></p>
 <p id="lowOrHigh" class="lowOrHigh"></p>
</div>
</body>
```

- (2) 为了让游戏界面显得更加美观，我们编写 css 为它添加一些样式（我下面的样式修改了字体并实现了居中显示，你可以添加背景色等其他效果）。

```

<style>
 html {
 font-family: sans-serif;
 }
 body {
 width: 50%;
 max-width: 800px;
 min-width: 480px;
 margin: 0 auto;
 }
 .lastResult {
 color: white;
 padding: 3px;
 }
</style>

```

(3) 为了完成猜数字游戏的功能，我们简单梳理一下代码编写时需要注意的逻辑：

- a) 首先生成一个随机数字
- b) 得到用户输入数字并与初始随机数比较大小
- c) 用户得到比较结果反馈，同时输入框重置以便于用户再次输入

处理清代码逻辑之后，我们再代码编写过程中还需要注意一些其他的细节：

- (1) 为了用户友好，我们应该保存并显示每次用户的输入记录
- (2) 用户每次输入结果都应该显示出来作为提示
- (3) 若 10 次都没有猜中，应重置游戏让用户重新开始

(4) 我们按照步骤依次进行，先生成随机数，并获得输入框 dom 节点，拿到用户输入：

```

let randomNumber = Math.floor(Math.random() * 100) + 1;
let counts = 1;

const guessField = document.getElementById("guessField");

```

(5) 为输入框绑定一个事件，使得我们每次点确定就能拿到用户输入与随机数判断大小

```

<input type="submit" value="确定" id="guessSubmit" class="guessSubmit" onclick="guess()">

```

(6) 编写判断用户输入与随机数大小判断的函数

```

function guess() {
 let value = guessField.value;
 if (value > randomNumber){
 window.alert("猜高了");
 } else if (value < randomNumber){
 window.alert("猜低了");
 } else {
 window.alert("恭喜你猜对了")
 }
}

```

(7) 对提示信息显示方式进行修改，将浏览器弹窗改为在页面上的提示



```
function guess() {
 let value = guessField.value;
 if (value > randomNumber){
 // window.alert("猜高了");
 addLastResult("你猜错了", "red");
 addLowOrHigh("猜高了");
 } else if (value < randomNumber){
 // window.alert("猜低了");
 addLastResult("你猜错了", "red");
 addLowOrHigh("猜低了");
 } else {
 // window.alert("恭喜你猜对了")
 addLastResult("你猜对了", "green");
 addLowOrHigh("猜对了");
 }
 guessField.value = null;

 addGuesses(value);
}
```

```
function addGuesses(value) {
 if (counts == 1){
 guesses.textContent = '上次猜的数: ';
 }
 guesses.textContent += value + " ";
}

function addLastResult(message, color) {
 lastResult.textContent = message;
 lastResult.style.backgroundColor = color;
}

function addLowOrHigh(message) {
 lowOrHigh.textContent = message;
}
```

- (8) 修改完善代码，游戏应该在结束后能够重置，编写重置函数并在相应的位置调用

```
function resetGame() {
 guessField.value = null;
 counts = 1;
 guesses.textContent = null;
 lastResult.textContent = null;
 lastResult.style.backgroundColor = null;
 lowOrHigh.textContent = null;
}
```

- (9) 完整的示例代码参考如下

```
<script>
 let randomNumber = Math.floor(Math.random() * 100) + 1;
 let counts = 1;

 const guessField = document.getElementById("guessField");
 const guesses = document.getElementById("guesses");
 const lastResult = document.getElementById("lastResult");
 const lowOrHigh = document.getElementById("lowOrHigh");

 function guess() {
 if (counts > 10){
 resetGame();
 window.alert("你的次数已经用完了! \n请重新开始游戏");
 return;
 }
 let value = guessField.value;
 if (value > randomNumber){
 // window.alert("猜高了");
 addLastResult("你猜错了", "red");
 addLowOrHigh("猜高了");
 } else if (value < randomNumber){
 // window.alert("猜低了");
 addLastResult("你猜错了", "red");
 addLowOrHigh("猜低了");
 } else {
 // window.alert("恭喜你猜对了")
 addLastResult("你猜对了", "green");
 addLowOrHigh("猜对了");
 }
 guessField.value = null;

 addGuesses(value);
 counts ++;
 }

 function addGuesses(value) {
 if (counts === 1){
 guesses.textContent = '上次猜的数: ';
 }
 guesses.textContent += value + " ";
 }

 function addLastResult(message, color) {
 lastResult.textContent = message;
 lastResult.style.backgroundColor = color;
 }

 function addLowOrHigh(message) {
 lowOrHigh.textContent = message;
 }

 function resetGame() {
 guessField.value = null;
 counts = 1;
 guesses.textContent = null;
 lastResult.textContent = null;
 lastResult.style.backgroundColor = null;
 lowOrHigh.textContent = null;
 }
}
</script>
```



(10) 进行测试并查看效果，游戏胜利的结果应该如下图所示：

## 猜数字游戏

我刚才随机选定了了一个100以内的自然数。看你能否在 10 次以内猜中它。每次我都会告诉你所猜的结果是高了还是低了。

请猜数:

上次猜的数: 10 20 30 40 35 34 33 32 31

你猜对了

猜对了

作业上交内容与事项：

1. 实验报告文档：填写（或截图）「实验记录和作业」后，上交原实验报告；
2. 程序代码文件：上交「实验记录和作业」中所要求的源代码文件；
3. 汇总 1、2 所有文件并打包成压缩包，上传到实验 2 下。

本期实验作业上期限：

请在实验设置的截止日期内提交实验报告结果，若逾期提交，成绩会适当被打折。

### 本次作业上交内容：

注：本次实验报告上交格式为 PDF 文件，具体的上交内容包括「实验记录和作业」中所要求的：

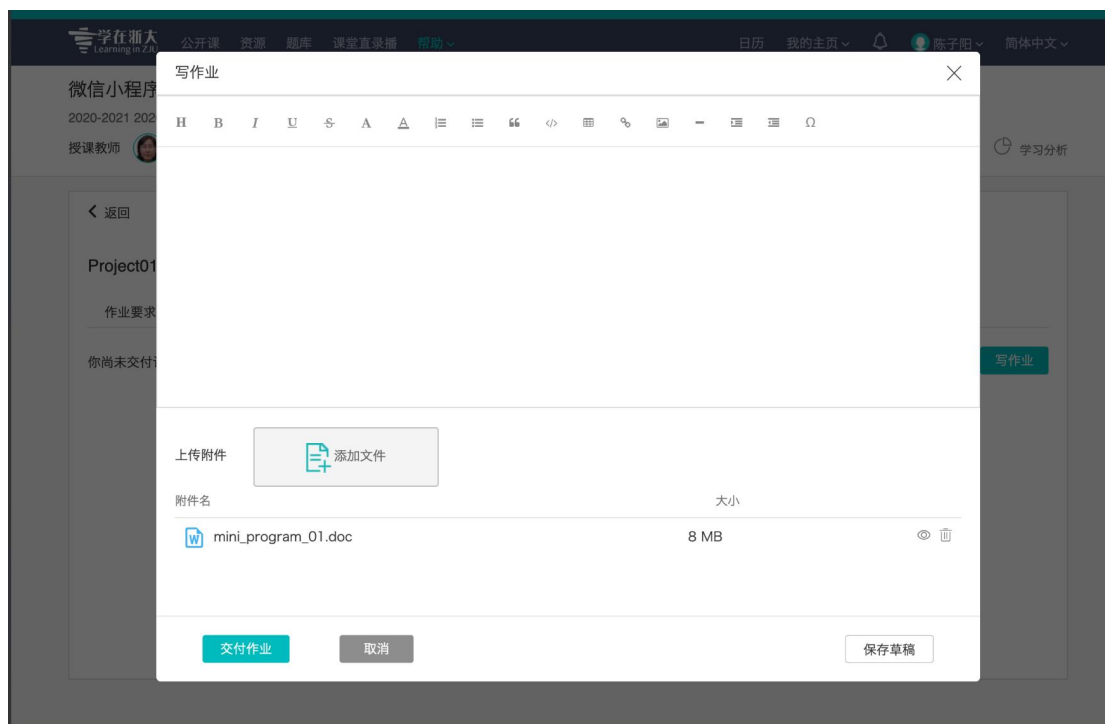
1. 问题解答
2. 要求的效果截图
3. 要求的源代码截图

按作业序号作答并写到指定位置，最终保存为 PDF 文件并上传

## 3. 实验感受及记录

**1.实验感受**（本次实验遇到的疑难问题及你的主要收获，即写你掌握了什么，问题还有哪些）

说明：请将此部分内容写到“写作业”下方的文本框中。



## 2. 实验记录和作业（实验作业完成好的源程序及运行结果放在此处）

(1) 认真完成实验步骤 1，回答以下问题：

- js 中主要有哪几种数据类型？
- js 变量可以声明为哪几种形式，有什么区别？
- js 中如何判断「相等」和「不相等」？
- js 中的 '+' 运算符有什么特殊的地方？

答：

1. 数据类型主要有 number、string、boolean、null、undefined、object 等，其中 object 又包括 array、function 等类型。

2. let、const、var 等三种。var 声明是函数范围的，声明之后，变量是没有值的。let 声明是块范围的，在赋值之前无法访问，不能在同一范围内重新声明。const 声明是块范围的，在赋值之前无法访问，不能在同一范围内重新声明，无法重新赋值。

3. == 运算符比较两个变量的值，若相等则返回 true。=== 运算符比较两个变量的值和类型，若都相等则返回 true。

4. 可用于对数字相加，也可用于对字符串进行相加（此时称为级联运算符），对一个数字和一个字符串相加将返回一个字符串。

(2) 认真完成实验步骤 1，回答以下问题：

- JavaScript 中获取一个真实的节点有哪些办法？
- 获取到一个节点（假设该节点名字为 node）后我想将它的背景色改为“blue”，应该通过什么 js 语句

答:

1. 通过 id 查找 HTML 元素: `document.getElementById`; 通过标签名查找 HTML 元素: `document.getElementsByTagName`; 通过类名查找 HTML 元素: `document.getElementsByClassName`; 通过 CSS 选择器查找 HTML 元素: `document.querySelector`; 通过 HTML 对象集合查找 HTML 元素: `document.forms`。

2. `node.background-color="blue";`

(3) 认真完成实验步骤 1, 回答以下问题:

- i. JavaScript 中控制浏览器弹窗有几种形式, 各是什么?
- ii. 他们的区别是什么, 我如何使弹出的信息换行显示?

答:

1. `alert`、`confirm`、`prompt`

2. `alert`: 使用 `alert` 弹框提示信息, 最后都会被转化为字符串输出; `confirm`: 在 `alert` 基础上增加了让用户选择性的操作, 提供确定和取消按钮; `prompt`: 在 `confirm` 的基础上增加让用户输入的效果。

在弹窗显示的字符串末尾加上 `\n`

(4) 完成实验步骤 2, 提交相关的代码文件

(5) 实现实验步骤 3, 提交相关的代码文件

(6) 完成实验步骤 4 (5), 同时要求当鼠标悬停在图片上的时候, 图片停止运动, 请将这个功能添加在上次实验中的个人网页中作为悬浮的小助手, 提交相关的源代码文件。(提示, 通过清除定时器)

由于没有想到个人主页中的悬浮小助手能加入什么功能, 我在猜数字页面增加了一个悬浮的重置游戏的按钮, 以使得它有实际的功能而不是个人主页的摆设, 也实现了鼠标悬停等功能。

(7) 实现实验步骤 5, 提交相关的代码文件