

## 《微信小程序综合实践》实验报告三

### 网页基础——JavaScript

姓名 俞仲炜 专 业 混合  
学号 3220104929 联系方式 zhongweiy@zju.edu.cn

#### 一、实验目的

1. 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；
2. 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；
3. 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；
4. 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；
5. 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；

#### 二、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, 按照步骤完成实验)

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

- a) 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;

- b) 打开 [https://www.w3school.com.cn/js/js\\_htmlDOM.asp](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;

- c) 打开 [https://www.w3school.com.cn/js/js\\_window.asp](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

- a) 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;
- b) 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;
- c) 在 index.html 文件中, `</body>` 标签后添加以下代码:  
`<script src="scripts/main.js"></script>`
- d) 打开 main.js 文件, 在其中添加如下内容:  
`let myHeading = document.querySelector('h1');`  
`myHeading.textContent = 'Hello world!';`

- e) 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

- a) 上次作业的表单如下图所示：

#### 注册信息

基本信息	用户姓名：	<input type="text"/>		
	真实姓名：	<input type="text"/>		
	密 码：	<input type="password"/>		
	确认密码：	<input type="password"/>		
	性 别：	<input checked="" type="radio"/> 男 <input type="radio"/> 女		
通信信息				
手机号码：	<input type="text"/>	电子邮箱：	<input type="text"/>	
居住地址：	<input type="text"/>			
额外信息				
爱 好：	<input type="checkbox"/> 篮球 <input type="checkbox"/> 足球 <input type="checkbox"/> 乒乓球			

- b) 现有新要求如下：
- （选做）尝试使用弹性布局（流动布局）使得输入框左边的描述汉字也能够左右对齐；
  - 使用 JavaScript 逻辑对表单的输入项进行校验，如：
    - 密码和确认密码必须一致；
    - 手机号码必须为 11 位数字；
    - 邮箱必须是 xxxxxxx@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

- a) 新建一个 index.html 文件，在该文件 body 中引入一张图片；

- b) 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；
- c) 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；
- d) 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
const zju = document.getElementById('zju');

// 拿到浏览器窗口宽高
const w_width = window.screen.availWidth;
const w_height = window.screen.availHeight;

// 生成图片的随机位置
let width = Math.random() * w_width;
let height = Math.random() * w_height;
zju.style.left = width + 'px';
zju.style.top = height + 'px';
```

- e) 通过使用 setInterval 函数使得图片能够每秒改变一次位置；
- f) 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：

```
// 初始位置
let width = 200;
let height = 200;
// 图片移动速度
let vx = 30;
let vy = 30;
// 改变图片位置
function movePos() {
    width = (width + vx) % (w_width-300);
    height = (height + vy) % (w_height-300);

    zju.style.left = width + 'px';
    zju.style.top = height + 'px';
}

setInterval(movePos, 1000);
```

## 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（可以参考下方的代码）

```
<h1>猜数字游戏</h1>
<p>我刚才随机选定了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">
  <input type="submit" value="确定" id="guessSubmit" onclick="guess()">
  <input type="submit" value="重置" id="resetSubmit" onclick="resetGame()">
</div>
<div class="result">
  <p id="guesses"></p>
  <p id="lastResult"></p>
  <p id="lowOrHigh"></p>
</div>
```

- (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（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）

```
html {
  font-family: sans-serif;
}
body {
  width: 50%;
  max-width: 800px;
  min-width: 480px;
  margin: 0 auto;
}
#lastResult {
  color: white;
  padding: 3px;
}
```

(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：

- 首先生成一个随机数字；
  - 得到用户输入数字并与初始随机数比较大小；
  - 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：

- 为了用户友好，我们应该保存并显示每次用户的输入记录；
- 用户每次输入结果都应该显示出来作为提示；
- 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;
let 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" 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( message: "你猜错了", color: "red");
    addLowOrHigh("猜高了");
  }else if(value < randomNumber) {
    // window.alert("猜低了");
    addLastResult( message: "你猜错了", color: "red");
    addLowOrHigh("猜低了");
  }else{
    // window.alert("恭喜你猜对了");
    addLastResult( message: "恭喜你猜对了", color: "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() {
  randomNumber = Math.floor( Math.random() * 100) + 1;
  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：

```

let randomNumber = Math.floor(Math.random() * 100) + 1;
let counts = 1;
let guessField = document.getElementById( elementId: "guessField");
let guesses = document.getElementById( elementId: "guesses");
let lastResult = document.getElementById( elementId: "lastResult");
let lowOrHigh = document.getElementById( elementId: "lowOrHigh");

function guess() {
  if(counts > 10) {
    resetGame();
    window.alert("你的次数已用完! \n请重新开始游戏。");
    return
  }

  let value = guessField.value;
  if (value > randomNumber) {
    // window.alert("猜高了");
    addLastResult( message: "你猜错了", color: "red");
    addLowOrHigh("猜高了");
  }else if(value < randomNumber) {
    // window.alert("猜低了");
    addLastResult( message: "你猜错了", color: "red");
    addLowOrHigh("猜低了");
  }else{
    // window.alert("恭喜你猜对了");
    addLastResult( message: "恭喜你猜对了", color: "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() {
  randomNumber = Math.floor(Math.random() * 100) + 1;
  guessField.value = null;
  counts = 1;
  guesses.textContent = null;
  lastResult.textContent = null;
  lastResult.style.backgroundColor = null;
  lowOrHigh.textContent = null;
}

```



(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: 50 75 65 60 55 52

恭喜你猜对了

猜对了

作业上交内容与事项：

1. 实验报告文档：按照要求完成实验并将关键步骤实验结果进行截图记录。

注意文档工整；

2. 程序代码：如果有程序代码或其他相关材料，可汇总压缩所有文件并上传压缩包。注意文件命名；

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，若逾期提交，成绩会适当被打折。

本次作业上交内容：

- 实验报告文档(.docx)
- 程序代码压缩文档(.zip)

## 三、实验感受及记录

(一) 实验感受（本次实验遇到的问题、主要收获等内容）

本次实验整体还算顺利，只有遇到了些小问题，搞了很久也没能解决。比如步骤三的左右对齐，步骤四刚加载进去图片不会自动开始移动，步骤五的js文件有个变量名打错了，都浪费了很多时间去解决。不过这也算是学习过程中必然得付出的吧，毕竟现在还不是很熟练，也算增长经验教训了。有些问题自己真的解决不了最后去找了助教，助教也很耐心地解答了，感觉收获满满。

(二) 实验记录（实验过程中关键步骤截图记录及文字描述）

2.1 认真完成实验步骤 1，回答以下问题：

- 1) js 中主要有哪几种数据类型？

1. 数值（number）：整数或者浮点数
2. 字符串（string）：一串表示文本值的字符序列
3. 布尔值（boolean）：true 或 false
4. 空值（null）：表示一个空对象指针
5. 未定义（undefined）：表示未定义的值
6. Symbol：唯一且不可变的值
7. 对象（object）：无序键值对的集合，或者是无序属性的集合
8. 数组（array）：按顺序排列的值的列表，也是对象的一种特殊形式

2) js 变量可以声明为哪几种形式，有什么区别？

在 JavaScript 中，变量可以声明为三种形式：var、let 和 const。

var 声明的变量作用域是函数作用域（Function Scope），不受块级作用域（Block Scope）的影响。在同一个函数内部，无论声明多少次都只会被视为一次声明，并且会进行变量提升（hoisting）。即变量会被提升至函数顶部进行声明，但赋值操作仍然在原来的位置。var 声明的变量可以被重新赋值，并且如果没有初始化，会被默认赋值为 undefined。

let 声明的变量只在当前代码块中有效，不受函数作用域的影响。在同一个块级作用域内，不能重复声明同一个变量。let 声明的变量可以被重新赋值。

const 声明的变量不能重新赋值，一旦声明就不能改变其值。在同一个块级作用域内，不能重复声明同一个常量。const 声明的变量必须在声明时进行初始化，否则会报错。

使用 let 和 const 可以更好地避免变量污染和误操作，并且符合更严格的编程规范。同时，let 和 const 的性能也比 var 更好。

3) js 中如何判断「相等」和「不相等」？

1. 「相等」：“===”用于判断两者是否完全相等
2. 「不相等」：“!==”用于判断两者是否不一样

4) js 中的 ‘+’ 运算符有什么特殊的地方？

JavaScript 中 ‘+’ 的功能不仅仅局限于数学运算。

1. 数字相加：‘+’ 运算符可以用于对数字进行相加操作。

2. 字符串拼接：‘+’ 运算符还可以用于将字符串进行拼接操作。如果其中一个操作数是字符串类型，那么它会将另一个操作数转换为字符串，然后进行拼接。
3. 字符串和其他类型的隐式转换：当‘+’ 运算符的操作数中包含字符串和其他类型（如数字、布尔值等）时，JavaScript 会进行隐式类型转换。它会将其他类型的值转换为字符串，然后进行拼接。
4. 数组拼接：‘+’ 运算符可以用于将两个数组进行拼接操作，生成一个新的数组。需要注意的是，数组拼接的结果是将两个数组的元素转换为字符串后进行拼接，而不是将两个数组合并成一个新的数组。

## 2.2 认真完成实验步骤 1，回答以下问题：

- 1) JavaScript 中获取一个真实的节点有哪些办法？
  - a) 通过 ID 获取节点：可以使用 ‘getElementById’ 方法通过节点的 ID 属性来获取节点。这个方法会返回具有指定 ID 的节点，如果找不到匹配的节点，则返回 ‘null’。
  - b) 通过类名获取节点：可以使用 ‘getElementsByClassName’ 方法通过节点的类名来获取节点。这个方法会返回一个 HTMLCollection 对象，其中包含具有指定类名的所有节点。
  - c) 通过标签名获取节点：可以使用 ‘getElementsByTagName’ 方法通过节点的标签名来获取节点。这个方法会返回一个 HTMLCollection 对象，其中包含具有指定标签名的所有节点。
  - d) 通过选择器获取节点：可以使用 ‘querySelector’ 或 ‘querySelectorAll’ 方法通过 CSS 选择器来获取节点。‘querySelector’ 方法返回匹配选择器的第一个节点，而 ‘querySelectorAll’ 方法返回一个 NodeList 对象，其中包含所有匹配选择器的节点。
  - e) 通过父节点获取子节点：如果已经获取到一个父节点，可以使用 ‘childNodes’、‘children’、‘firstChild’、‘lastChild’ 等属性来获取其子节点。

- 2) 获取到一个节点（假设该节点名字为 `node`）后想将它的背景色改为“blue”，应该通过什么 js 语句实现？

```
node.style.backgroundColor = "blue";
```

### 2.3 认真完成实验步骤 1，回答以下问题：

- 1) JavaScript 中控制浏览器弹窗有几种形式，各是什么？
  - a) ‘alert’: ‘alert’ 是最简单的浏览器弹窗形式之一。它会在浏览器中显示一个包含文本消息和一个确认按钮的对话框。使用 ‘alert’ 可以向用户显示一条简单的信息，但无法定制对话框的外观或按钮行为。
  - b) ‘confirm’: ‘confirm’ 用于显示一个包含文本消息、确认按钮和取消按钮的对话框。用户可以通过点击确认按钮或取消按钮来做出选择。‘confirm’ 返回一个布尔值，表示用户是否点击了确认按钮。
  - c) ‘prompt’: ‘prompt’ 用于显示一个包含文本消息、输入框、确认按钮和取消按钮的对话框。用户可以在输入框中输入文本，并通过点击确认按钮或取消按钮来做出选择。‘prompt’ 返回一个字符串，表示用户输入的文本，或者返回 ‘null’，表示用户点击了取消按钮。
- 2) 他们的区别是什么，如何使弹出的信息换行显示？

区别：

  - a) ‘alert’ 用于显示一个简单的消息框，只包含文本消息和一个确认按钮。无法定制对话框的外观或按钮行为。无法换行显示消息文本。
  - b) ‘confirm’ 用于显示一个包含文本消息、确认按钮和取消按钮的对话框。用户可以点击确认按钮或取消按钮来做出选择。返回一个布尔值，表示用户是否点击了确认按钮。无法换行显示消息文本。
  - c) ‘prompt’ 用于显示一个包含文本消息、输入框、确认按钮和取消按钮的对话框。用户可以在输入框中输入文本，并点击确认按钮或取消按钮来做出选择。返回一个字符串，表示用户输入的文本，或者返回 ‘null’，表示用户点击了取消按钮。无法换行显示消息文本。

换行：可以在文本中使用换行符 ‘\n’ 或者 ‘<br/>’

2.4 完成实验步骤 2，在下方附上页面截图并提交相关的代码文件。

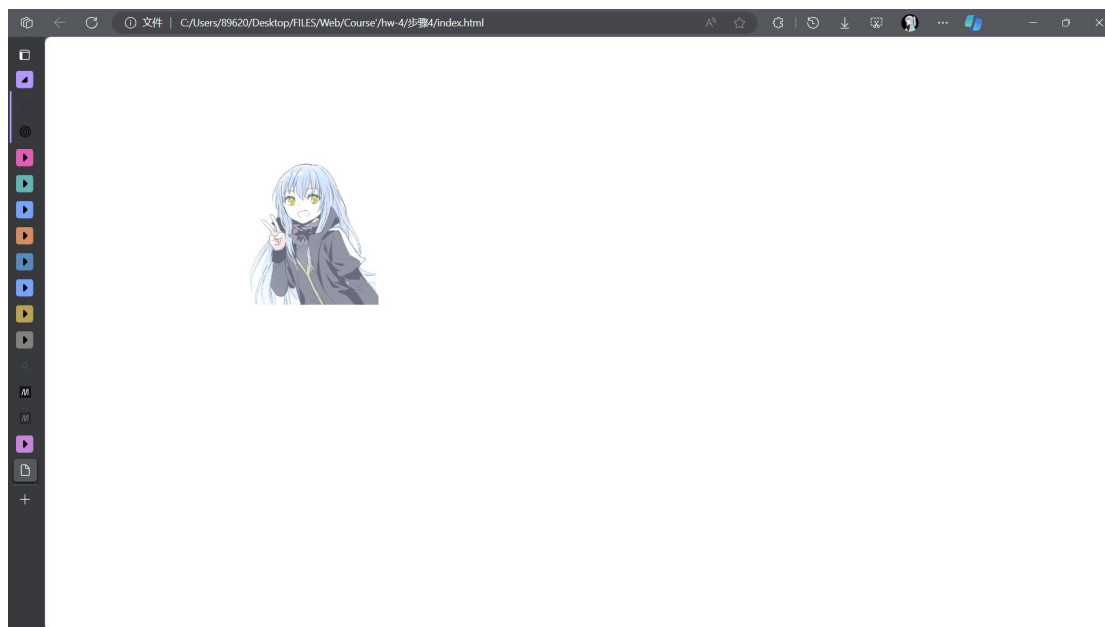


2.5 实现实验步骤 3，在下方附上页面截图并提交相关的代码文件。

注册信息

基本信息	用户姓名:	<input type="text"/>	
	真实姓名:	<input type="text"/>	
	密码:	<input type="password"/>	
	确认密码:	<input type="password"/>	
	性别:	<input type="radio"/> 男 <input type="radio"/> 女	
通信信息			
手机号码:	<input type="text"/>	电子邮箱:	<input type="text"/>
居住地址:	<input type="text"/>		
额外信息			
爱好	<input type="checkbox"/> 篮球 <input type="checkbox"/> 足球 <input type="checkbox"/> 乒乓球		
头像	<input type="button" value="选择文件"/> 未选择文件		
个人说明	<input type="text"/>		

2.6 完成实验步骤 4，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，在下方附上页面截图并提交相关的源代码文件。（提示：清除定时器）



2.7 完成实验步骤 5，在下方附上页面截图并提交相关的代码文件。

## Number guessing game

We have selected a random number between 1 and 100. See if you can guess it in 10 turns or fewer. We'll tell you if your guess was too high or too low.

Enter a guess:

Previous Guesses: 5 50 25 15 20 17 18 19

**Congratulations! You got it right!**