

软件工程第四次实验课

参考资料

菜鸟教程网址: [JavaScript 测试 jQuery | 菜鸟教程 \(runoob.com\)](https://www.runoob.com)

相关课程参考博主: B 站 up 主: 黑马 pink 讲前端

[JavaScript 前端课程-dom-bom-js-es6 新语法-jQuery-数据可视化 echarts 黑马 pink 老师前端入门基础视频教程\(500 多集\)持续 哔哩哔哩_bilibili](#)

相关教程: [第二篇: 前端 VSCode 常用插件-快捷键-以及常用技巧-pink 老师 - 哔哩哔哩 \(bilibili.com\)](#)

一、 初识 javascript

1.1 web 端构成

上节课我们说到每一个网页文件就是 html 文件, 但我们所看到的网页一般都是由 html、css 和 javascript 封装而成。三者缺一不可。那么根据上节课的学习, 我们已经可以初步写出一个静态的网页了, 那我们知道我们平时上网的网页都会有交互功能, 比如输入用户名密码, 就登录跳转界面。这个操作就需要我们的 javascript 了。

1.2 Javascript

- JavaScript 是世界上最流行的语言之一, 是一种运行在客户端的脚本语言
- 脚本语言: 不需要编译, 运行过程中由 js 解释器(js 引擎)逐行来进行解释并执行
- 现在也可以基于 Node.js 技术进行服务器端编程

1.3 JavaScript 的作用

- 表单动态校验(密码强度检测) (JS 产生最初的目的)
- 网页特效
- 服务端开发(Node.js)
- 桌面程序(Electron)
- App(Cordova)
- 控制硬件-物联网(Ruff)
- 游戏开发(cocos2d-js)

1.4 HTML/CSS/JS 的关系

1.4.1 HTML/CSS 标记语言--描述类语言

- HTML 决定网页结构和内容(决定看到什么), 相当于人的身体。
- CSS 决定网页呈现给用户的模样(决定好不好看), 相当于给人穿衣服、化妆。

1.4.2 JS 脚本语言--编程类语言

- 实现业务逻辑和页面控制(决定功能), 相当于人的各种动作

1.5 Javascript 的简单使用

由于我们时间有限, 大家也有一定的编程基础, 语言不同但是底层逻辑是类似的。所以就只简单介绍一些本堂课所需要的内容

1.5.1 变量

白话：变量就是一个装东西的盒子。

通俗：变量是用于存放数据的容器。我们通过变量名获取数据，甚至数据可以修改。

那么 javascript 怎么应用到 html 里面呢？非常简单，我们上节课说到，在 vscode 中输出感叹号+tab 键会自动生成一段代码，如下：

```
<!DOCTYPE html>
<html lang="en">

<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <meta http-equiv="X-UA-Compatible" content="ie=edge">
  <title>Document</title>
</head>
<body>

</body>
</html>
```

我们在红色箭头位置加入<script>xxxxx</script>。xxxxx 的部位就是我们写具体代码的地方我们先生成一个变量 age，在 javascript 中，是用 var + 变量名 = 变量值。类似 c 语言的 c 语言的 int age = 18 完整的代码如下：

```
<!DOCTYPE html>
<html lang="en">

<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <meta http-equiv="X-UA-Compatible" content="ie=edge">
  <title>Document</title>
  <script>
    // 1. 声明了一个 age 的变量
    var age;
    // 2. 赋值 把值存入这个变量中
    age = 18;
  </script>
</head>

<body>

</body>
</html>
```

1.5.2 网页控制台输出信息

那么赋完值，怎么看结果呢，c 语言是用 printf，javascript 用的是 console.log

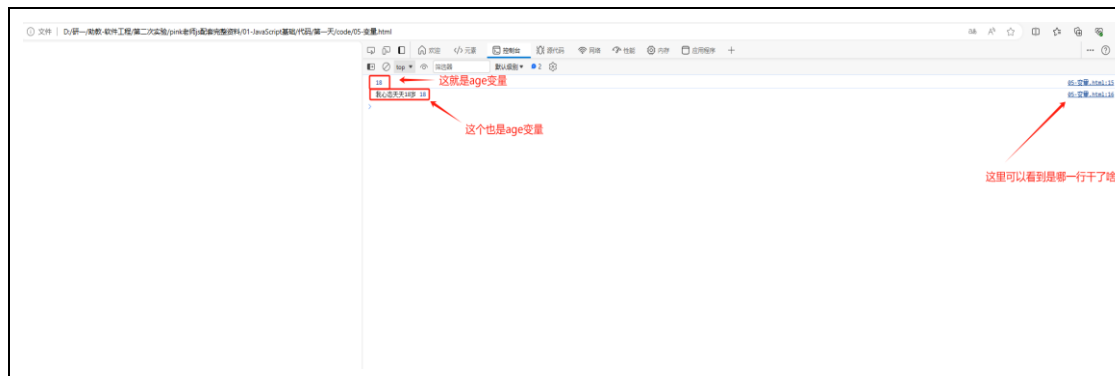
比如在上面的代码声明了 age = 18，那么输出就是 console.log(age) 然后 ctrl+s 保存有点麻烦的是，我们需要打开网页才能看到输出的结果。

具体的代码如下：

```
<!DOCTYPE html>
<html lang="en">

<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <meta http-equiv="X-UA-Compatible" content="ie=edge">
  <title>Document</title>
  <script>
    // 1. 声明了一个 age 的变量
    var age;
    // 2. 赋值 把值存入这个变量中
    age = 18;
    // 3. 输出结果
    console.log(age);
    console.log('我心态天天 18 岁',age)
  </script>
</head>
<body>
</body>
</html>
```

保存后我们打开网页，按 Fn+F12 会出现如下界面



声明的内容也可以是字符串，同 c 语言都是类似的。

```
<script>
  // 1. 声明了一个 age 的变量
  var age;
  // 2. 赋值 把值存入这个变量中
  age = 18;
  // 3. 输出结果
  console.log(age);
  // 4. 变量的初始化
  var myname = 'pink 老师';
  console.log(myname);
</script>
```

1.6 JavaScript 函数的使用

同样和 c 语言类似，javascript 的函数声明是：

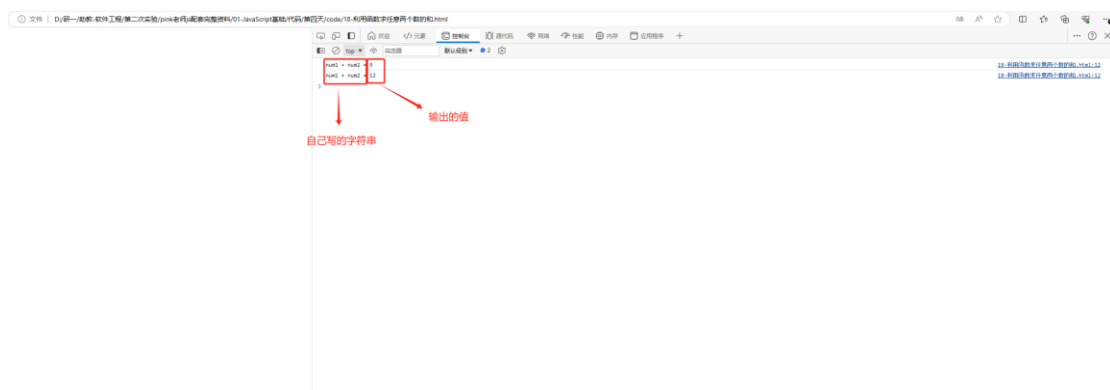
```
function (函数名) {  
    //函数体代码  
}
```

- function 是声明函数的关键字,必须小写

- 由于函数一般是为了实现某个功能才定义的，并且通常都是在交互中，获得什么东西，所以通常我们将函数名命名为动词，比如 getSum

简单写一个求和函数：

```
<!DOCTYPE html>  
<html lang="en">  
  
<head>  
    <meta charset="UTF-8">  
    <meta name="viewport" content="width=device-width, initial-scale=1.0">  
    <meta http-equiv="X-UA-Compatible" content="ie=edge">  
    <title>Document</title>  
    <script>  
        // 1. 利用函数求任意两个数的和  
        function getSum(num1, num2) {  
            console.log('num1 + num2 =', num1 + num2);  
  
        }  
        getSum(1, 8);  
        getSum(3, 9);  
    </script>  
</head>  
<body>  
</body>  
</html>
```



1.7 函数小练习：利用函数计算 1-100 之间的累加和，结果 console.log 出来

自己搜索一下相关的循环语法，写一个函数，并保存。

需要提交

目前的效果如下：

← ↻ 🏠 🔍 ⓘ 文件 | D:/研一/助教-软件工程/第二次

- 多个里面筛选几个
 - 多个里面筛选几个
 - 多个里面筛选几个
 - 多个里面筛选几个
 - 多个里面筛选几个
 - 多个里面筛选几个
1. 多个里面筛选几个
 2. 多个里面筛选几个
 3. 多个里面筛选几个
 4. 多个里面筛选几个
 5. 多个里面筛选几个
 6. 多个里面筛选几个

那我们的 jQuery 要怎么选择呢？

首先，要先写基本元素：

```
<script>
    $(function() {
    })
</script>
```

这个“\$”就表示调用的是 jQuery 库，这个库的文件已经下载好会发给你们，因此具体的调用 jQuery 库，和使用 jQuery 的代码如下：

```
<!DOCTYPE html>
<html lang="en">

<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <meta http-equiv="X-UA-Compatible" content="ie=edge">
    <title>Document</title>
    <script src="jquery.min.js"></script> <!-- 调用 jQuery 库 -->
</head>
<body>
    <ul>
        <li>多个里面筛选几个</li>
        <li>多个里面筛选几个</li>
        <li>多个里面筛选几个</li>
        <li>多个里面筛选几个</li>
        <li>多个里面筛选几个</li>
        <li>多个里面筛选几个</li>
    </ul>
    <ol>
        <li>多个里面筛选几个</li>
        <li>多个里面筛选几个</li>
        <li>多个里面筛选几个</li>
        <li>多个里面筛选几个</li>
        <li>多个里面筛选几个</li>
    </ol>
```

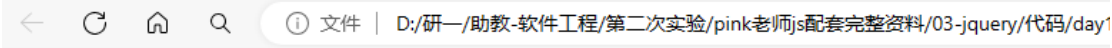
```
<li>多个里面筛选几个</li>
</ol>
<script>
    $(function() { // 使用 jQuery 库
        $("ul li:first").css("color", "red");
        $("ul li:eq(2)").css("color", "blue");
        $("ol li:odd").css("color", "skyblue");
        $("ol li:even").css("color", "pink");
    })
</script>
</body>

</html>
```

下面我们着重分析这四句话：

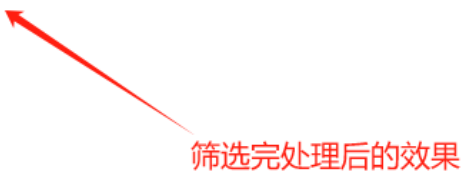
`$("ul li:first").css("color", "red");` 从 ul 里面的 li 的第一个修改 css，将颜色变成红色
`$("ul li:eq(2)").css("color", "blue");` 从 ul 的 li 中的索引 2 也就是第三个元素变成蓝色
`$("ol li:odd").css("color", "skyblue");` 从 ol 获取到的 li 元素中，选择索引号为奇数的元素
`$("ol li:even").css("color", "pink");` 从 ol 获取到的 li 元素中，选择索引号为偶数的元素

大家自己尝试一下，在网页端看看效果。目前应该已经有一点了解了，就是选择某些元素，对这些元素进行一些功能的定义。



- 多个里面筛选几个
- 多个里面筛选几个
- 多个里面筛选几个
- 多个里面筛选几个
- 多个里面筛选几个
- 多个里面筛选几个

1. 多个里面筛选几个
2. 多个里面筛选几个
3. 多个里面筛选几个
4. 多个里面筛选几个
5. 多个里面筛选几个
6. 多个里面筛选几个



筛选完处理后的效果

2.2 学院官网下拉列表的应用

下面我们来练习如何使用 jQuery 来简单的达到这一效果



2.2.1 补充知识（ul 和 li）



我们补充一个 html 的无序列表的知识， 标签表示 HTML 页面中项目的无序列表，一般会以项目符号呈现列表项，而列表项使用 标签定义。

无序列表的基本语法格式如下：

```
<ul class="nav">
  <li>
    <a href="#">学院概况</a>

  </li>
  <li>
    <a href="#">师资力量</a>

  </li>
  <li>
    <a href="#">教务教学</a>

  </li>
  <li>
    <a href="#">学生工作</a>

  </li>
</ul>
```

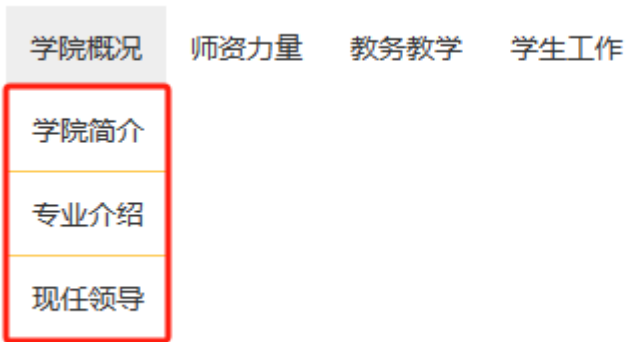
结合 css 的语法：

```
.nav>li {
    position: relative;
    float: left; /* 定义这个列表是横着排列 */
    width: 80px;
    height: 41px;
    text-align: center;
}
```

意思其实就是 ul 表示我有一个表，li 就是下面的具体元素，效果如下：

学院概况 师资力量 教务教学 学生工作

那么接下来，我们要补充下拉列表的元素，效果图如下：



在本节最前面的图，我们箭头白色字的说明了，其实这也是一个列表，他是嵌套在学院概况的这个 li 下面，所以我们只需要在对应的学院概况、师资力量等等的 li 标签下，嵌套 ul 和 li，补充学院简介、专业介绍、现任领导这几个文字到表中，代码如下：

```
<ul class="nav">
  <li>
    <a href="#">学院概况</a>
    <ul>
      <li>
        <a href="">学院简介</a>
      </li>
      <li>
        <a href="">专业介绍</a>
      </li>
      <li>
        <a href="">现任领导</a>
      </li>
    </ul>
  </li>
  <li>
    <a href="#">师资力量</a>
    <ul>
      <li>
        <a href="">教授</a>
      </li>
      <li>
        <a href="">副教授</a>
      </li>
      <li>
        <a href="">助理教授</a>
      </li>
    </ul>
  </li>
</ul>
```

```

</li>
  <a href="#">教务教学</a>
  <ul>
    <li>
      <a href="">教学通知</a>
    </li>
    <li>
      <a href="">本科生教务</a>
    </li>
    <li>
      <a href="">研究生教务</a>
    </li>
  </ul>
</li>
<li>
  <a href="#">学生工作</a>
  <ul>
    <li>
      <a href="">规章制度</a>
    </li>
    <li>
      <a href="">学工通知</a>
    </li>
  </ul>
</li>
</ul>

```

这时候的效果如下，同时和大家说明，下拉列表的元素，属于是学院概况这些元素的子代

学院概况	师资力量	教务教学	学生工作
学院简介	教授	教学通知	规章制度
专业介绍	副教授	本科生教务	学工通知
现任领导	助理教授	研究生教务	

但是我们是要下拉列表，不应该是直接显示，而是鼠标经过的时候才显示，所以我们加一个 css 语法，定义，下拉列表里面的元素的 display 为 none。

```

.nav ul {
    display: none;
    position: absolute;
}

```

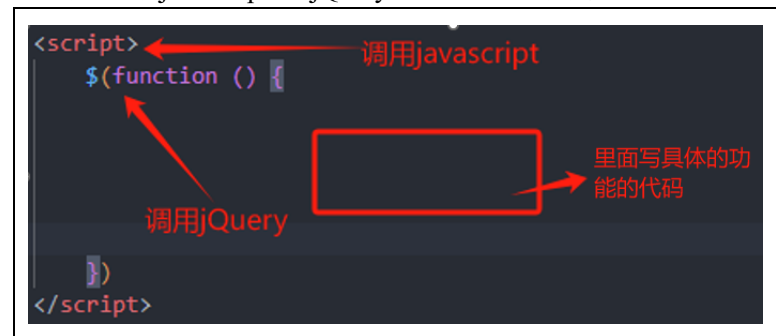
这个代码的意思是，在 nav 类（最开始最外层的 ul 是给了 class: nav 的）里面的 ul，也就是 nav 下的所有 ul，意思就是下拉列表里面的这些元素，默认展示方式是不显示。那如何显示他们，怎样的情况下显示，就看 JavaScript 的了。

2.2.2 jQuery 的简单应用

目前的话，我们已经定义了一个横着的表，同时每个元素还包含了一个下拉列表，只是没把他显示。那如果想显示隐藏的下拉列表我们先思考功能：

- 1、当鼠标经过触碰到顶部的列表，弹出下拉列表
- 2、当鼠标离开，继续隐藏下拉列表。

我们先写好 javascript 和 jQuery 的调用的基本代码：



接着实现 1：

```
<script>
    $(function () {
        // 鼠标经过
        $(".nav>li").mouseover(function () {
            // $(this) jQuery 当前元素 this 不要加引号
            // show() 显示元素 hide() 隐藏元素
            $(this).children("ul").show();
        });
    })
</script>
```

因为是第一次学习，这里逐个字进行讲解：

`$(".nav>li")` → 表示选择 class 为 nav 里面的 li 标签
`.mouseover` → 如何选中这些选中的元素？定义 `.mouseover` 意思是当鼠标经过的时候，后面跟 `(function())` 表示当鼠标经过，此时已经找到了对应的元素之后执行操作。
`function () {`
 `$(this).children("ul").show();`
 `}` → `this` 就是表示选中的元素，因为我们是要展示下拉元素，是 li 里面的 ul，也就是此时的 li 的子代，我们用 `children` 表示，然后定义 `.show` 表示原本是 `display: none`，现在表现出来，就会达到弹出下拉列表的效果了

`$(this).children("ul").show();`：简洁来说，就是选中横着的表的 li 里面的子代的 ul 元素，并展现他们。

效果如下：

学院概况	师资力量	教务教学	学生工作
学院简介	教授		
专业介绍	副教授		
现任领导	助理教授		

那么现在的效果就是，当鼠标经过他会弹出下拉列表，但是当鼠标离开，下拉列表不会消失了，因为他的 display 被定义成 show 了，那么我们执行第二步

```
// 鼠标离开
$(".nav>li").mouseout(function () {
    $(this).children("ul").hide();
});
```

同样的道理这个代码表示当鼠标离开（mouseout）nav 下面的 li 时，该 li 的子代中的 ul 隐藏起来（.hide）

整个 JavaScript 块的代码如下：

```
<script>
$(function () {
    // 鼠标经过
    $(".nav>li").mouseover(function () {
        // $(this) jQuery 当前元素 this 不要加引号
        // show() 显示元素 hide() 隐藏元素
        $(this).children("ul").show();
    });
    // 鼠标离开
    $(".nav>li").mouseout(function () {
        $(this).children("ul").hide();
    });
});
</script>
```

大家可以自行熟悉，这节的代码放在 2.2 练习发给大家，大家自行熟悉感受，无需提交。

三、 暴打灰太狼网页小游戏制作

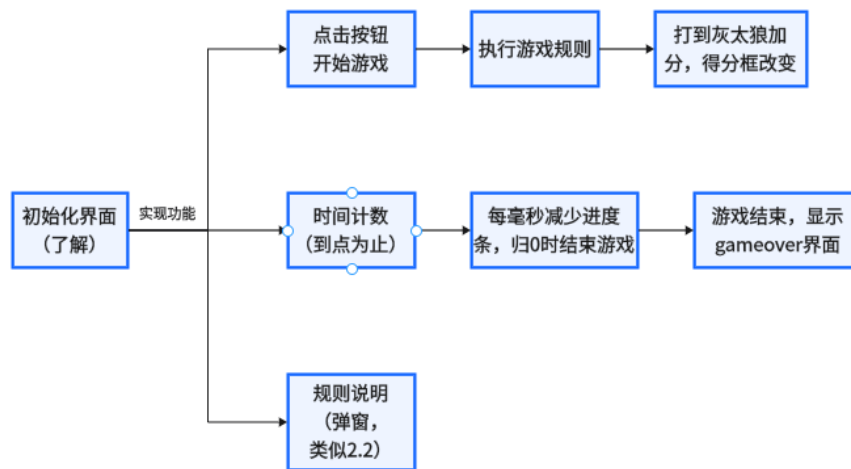
本节，我们会简单介绍一个小游戏的制作，具体的样式如下面两幅图所示：



游戏中，灰太狼会随机出现，当我们在规定时间范围内点击灰太狼，就会加十分。

3.1 梳理软件“生命周期”

下面用流程图进行了简单的讲解：

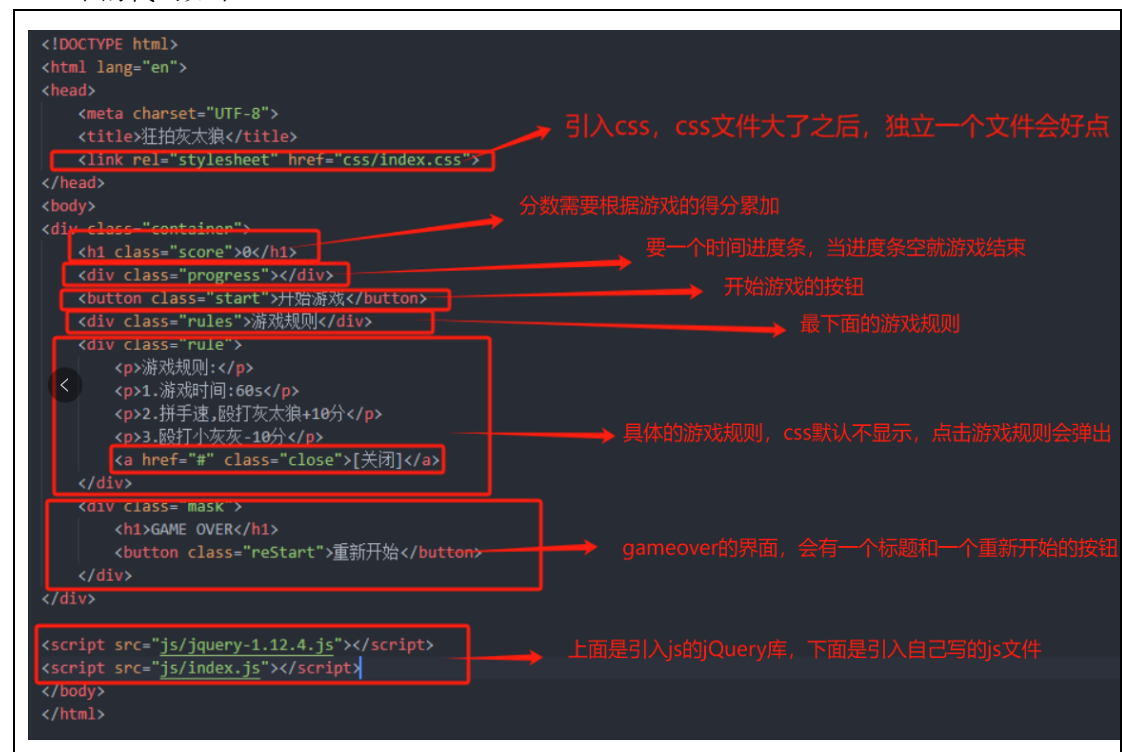


3.2 初始化界面

首先的初始化界面，由于本节课重点在于 javascript 的掌握，所以不作要求，给大家介绍基本的要素，后面代码会发给大家。



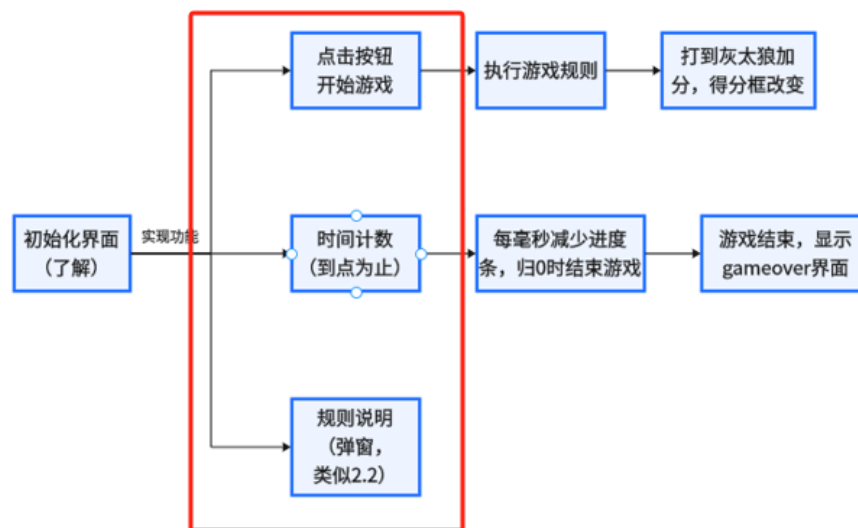
首先上图是我们拿到的背景图片，我们需要补充的东西红色字已经作了说明，所以我们的html 中的代码如下：



Css 部分因为是美观画面，我们就不做过多介绍了，代码都会给到大家，大家拿到的就是已经美化过的界面了。

下面我们开始 javascript 的实验。

3.3 基本功能实现（为了让大家积极练习，这里不给文档的代码，只给图片，大家自己输入进去，感受效果）



我们按流程图先实现红色框的部分。

由于我们无论是点击按钮开始游戏还是说点击规则出现弹窗，我们都应该用淡入淡出的功能会不显得突兀，这个功能 jQuery 已经封装好了，首先我们先写调用 jQuery 的代码

```
$(function () {  
  //输入我们需要的代码  
});
```

3.3.1 点击按钮开始游戏

我们先实现第一个功能，当我们点击开始按钮就开始游戏，上面我们说了 html 中定义了“开始游戏”的按钮的 class 是 start，那我们就选择这个类然后执行开始游戏的操作：

```
// 监听游戏开始  
$(".start").click(function () {  
  // 选类  
  // 鼠标点击时  
  // 函数：执行里面的这些东西  
  $(".start").stop().fadeOut(100);  
  // 游戏开始时，这个按钮就不要显示了，fadeout 淡出，stop 就是停止他的显示，不用扣原理  
  // 调用进度条方法  
  progressHandler();  
  // 开启灰太狼动画  
  startWolfAnimation();  
});
```

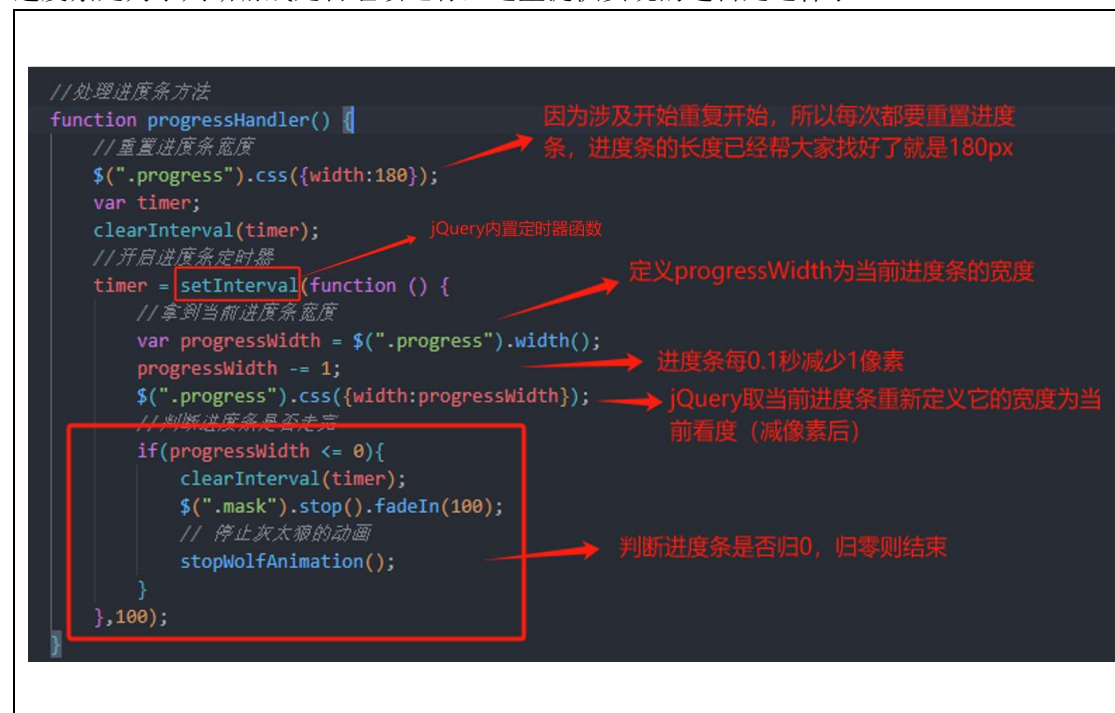
进度条和游戏的函数我们后面说。

3.3.2 点击规则弹出规则

```
// 监听游戏规则  
$(".rules").click(function () {  
  // 同样的逻辑，并且这里不用调其他函数  
  $(".rule").stop().fadeIn(100);  
});  
// 监听游戏规则关闭  
$(".close").click(function () {  
  $(".rule").stop().fadeOut(100);  
});
```

3.3.3 进度条函数 progressHandler

进度条是为了判断游戏是否继续运行，这里提供实现的逻辑是这样子：



这里补充说明：`setInterval()`是 JavaScript 中的一个函数，用于按照指定的时间间隔（以毫秒为单位）重复执行指定的代码或函数。

具体用法：`setInterval(function, milliseconds);`

`function` 为执行的功能，`milliseconds` 是重复执行的时间间隔，以毫秒为单位。这里定义 100 就是 100 毫秒也就是 0.1 秒。

3.3.4 制定规则

我们在 html 中定义了两个类，一个是 class 为 `rules` 的，是图片最下面可以点击的灰色框，一个是具体规则 class 为 `rule`。



复习一下第二节讲述的，如果一开始不显示的内容，我们在 css 中就应该设置为 none，大家如果以后开发软件会经常用到。所以这里具体的规则类为 rule 的就如下设置

```
.rule{
  width: 100%;
  height: 100%;
  background-color: rgba(0,0,0,.5);
  position: absolute;
  left: 0;
  top: 0;
  padding-top: 100px;
  box-sizing: border-box;
  text-align: center;
  display: none;
}
```

那么在 js 中我们就需要和第二节一样的，让 rule 显示出来：

```
// 监听游戏规则
$(".rules").click(function () {
  $(".rule").stop().fadeIn(100);
});
// 监听游戏规则关闭
$(".close").click(function () {
  $(".rule").stop().fadeOut(100);
});
```

——> 当我们点击，规则（灰色框）的时候
——> 对应函数执行具体规则淡入显示

此时界面如下：



那么规则弹出后肯定也要关闭，所以我们要写一个函数，上面图片写了功能了：

```
// 监听游戏规则
```

```
$(".rules").click(function () {  
    $(".rule").stop().fadeIn(100);  
});
```

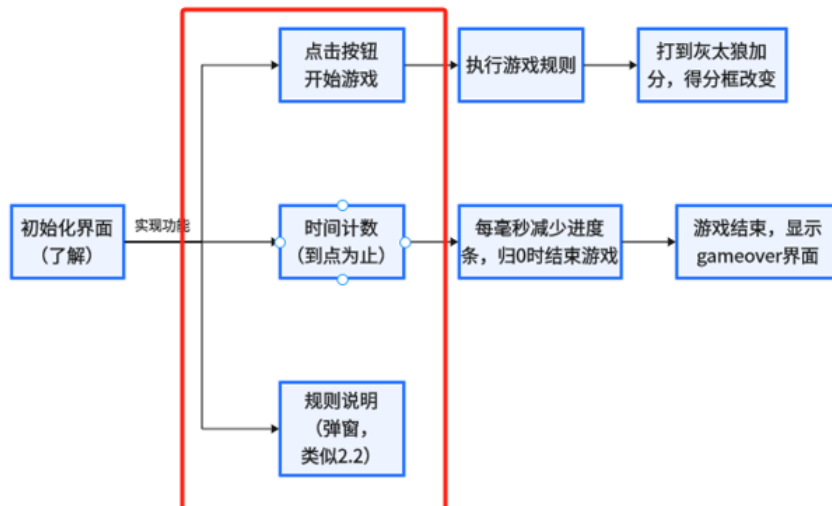
```
// 监听游戏规则关闭
```

```
$(".close").click(function () {  
    $(".rule").stop().fadeOut(100);  
});
```

我们定义了这个关闭class为close

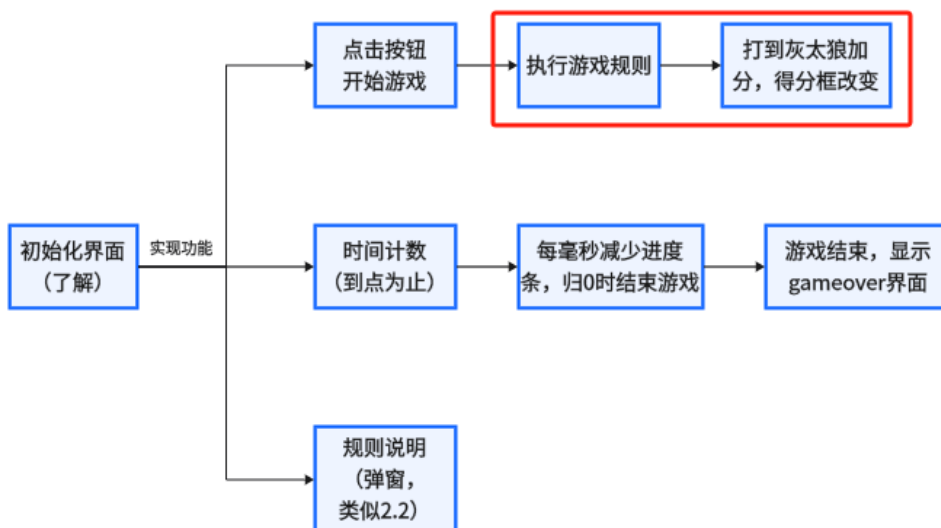
规则对应就淡出fadeOut即可

那么到这一步，我们的流程图的红色框就已经完成了。



3.3.5 执行游戏规则

下面我们来实现这一块：

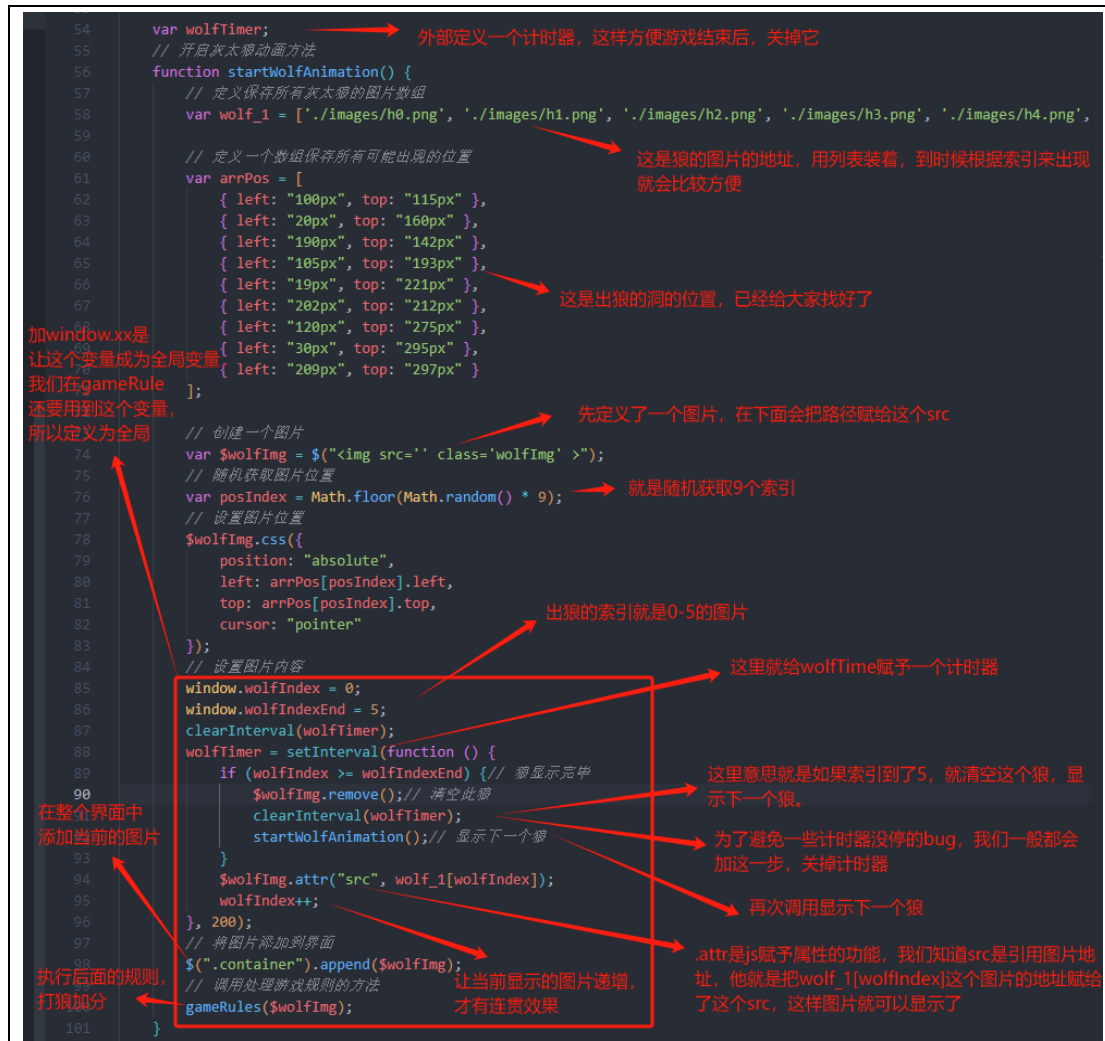


我们在 3.3.1 中开始按钮，执行后，除了调用了时间计数函数外，还要开启游戏，就是里面写的 `startWolfAnimation()` 函数，这个是执行出狼的函数。

先理清一下我们的思路，由于我们是简单制作，为了出狼连贯我们就用 6 张图片依次显示来达到这个效果，图片已准备好，左边 6 个是出狼，就是一个头先出到全部展示的过程。



具体这部分的代码在图片中给大家解释了，大家自己复现一遍就会有所了解

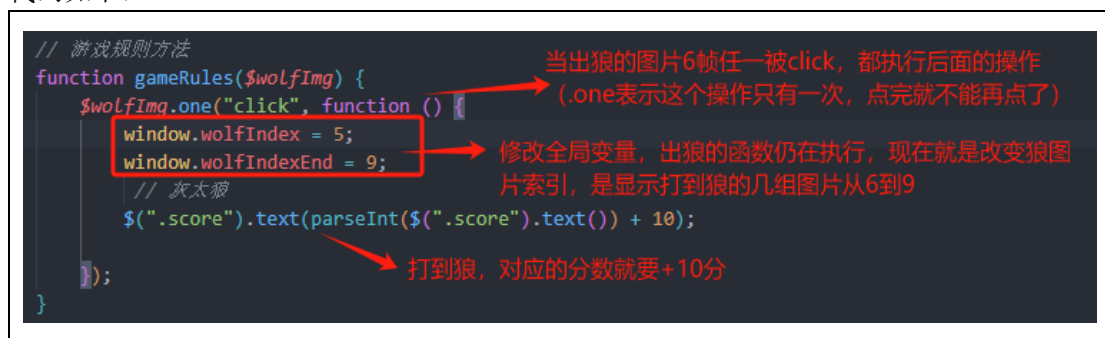


3.3.6 游戏规则

我们是当狼出现的时候，点击他就会加十分，所以需要的功能有：

- 1、点击狼，要切换图片，是狼被锅打的图片，已经给大家准备好了就是索引 6-9 的四张图
- 2、因为打到狼我们就加十分，但是因为我们的狼出现会有 6 帧，所以在这 6 帧内达到都算数，都要加分。

代码如下：



那么到这里，我们除了游戏结束的操作没做完之外，基本的游戏已经可以运行了。

3.3.7 完成最后一步（Game Over 设置）

其实最后一步就是在时间条走完之后实现的功能，这一步我们就补充在进度条方法的地方没有介绍的。

```
// 处理进度条方法
function progressHandler() {
    // 重置进度条宽度
    $(".progress").css({ width: 180 });
    var timer;
    clearInterval(timer);
    // 开启进度条定时器
    timer = setInterval(function () {
        // 拿到当前进度条宽度
        var progressWidth = $(".progress").width();
        progressWidth -= 1;
        $(".progress").css({ width: progressWidth });
        // 判断进度条是否走完
        if (progressWidth <= 0) {
            clearInterval(timer);
            $(".mask").stop().fadeIn(100);
            // 停止灰太狼的动画
            stopWolfAnimation();
        }, 100);
    }

    // 停止灰太狼动画
    function stopWolfAnimation() {
        $(".wolfImg").remove();
        clearInterval(wolfTimer);
    }
}
```

当进度条走完之后，我们关闭计时器，同时html定义的mask类淡入

移除动画图片，关闭计时器

此时的效果如下：



此时应该是这个界面，我们就提供一个功能点击重新开始，就重新运行出狼的函数即可

那个给重新开始的函数，就当做练习，大家自己写了，和开始是类似的，但注意要把分数归零。

本堂课作业

一、1.7 的 javascript 函数练习

二、小游戏实现：

- 1、把小游戏代码复现，并添加重新开始的函数
- 2、我把小灰灰的图片也给到大家了，和灰太狼一样的操作，大家增加一个功能，就是随机弹出灰太狼和小灰灰，灰太狼+10 分，小灰灰扣 10 分。（提示：出狼函数加多小灰灰数组，规则函数设置奖惩机制）
- 3、修改进度条时长到 30s，当前是 18s，进度条的像素 180px 是定死的，可以从 setInterval 计时器入手，不要求进度条减少得很顺滑。
(js 中如果每次减少 0.几他可能默认会四舍五入，所以尽量不要 progressWidth 这里减去一个 0.几的数字)
- 4、（挑战题）设置 bonus，当我们的得分每到 50 分就加 1s。

注意，小游戏实现的 javascript 代码在 js 文件下的 index 文件，框架已经做好，在里面填充即可。

