《Web程序设计》课程实验指导

**实验六 响应式布局实验**

2018年1月

目录

[实验六 响应式布局实验 3](#_Toc503338444)

[实验目的 3](#_Toc503338445)

[实验环境（硬/软件要求） 3](#_Toc503338446)

[实验要求 3](#_Toc503338447)

[实验指导 4](#_Toc503338448)

[1 响应式布局概念介绍 4](#_Toc503338449)

[2 功能卡片响应式布局介绍 4](#_Toc503338450)

[3 功能卡片响应式布局开发指导 4](#_Toc503338450)

[实验后续任务 4](#_Toc503338451)

# 实验六 响应式布局实验

## 实验目的

1. 了解响应式布局概念
2. 完成功能卡片响应式布局

## 实验环境（硬/软件要求）

1. 本实验需要计算机设备一套，需要安装的软件有： MySQL数据库、PHP解释容器、Sublime、Chrome浏览器等。

## 实验要求

1. 了解响应式布局概念
2. 完成三种布局切换过程
3. 完成同一布局中大小变化自适应

## 实验指导

### 1 响应式布局概念介绍：

1. 为什么需要响应式布局：

随着互联网的不断发展，人们不再仅仅依赖电脑作为上网工具，手机、平板电脑甚至手表都能浏览网页。不同端设备分辨率不同，手机屏幕比较小平板次之而且不同型号电脑也有不同分辨率。同样的内容，要在大小迥异的屏幕上，都呈现出满意的效果，并不是一件容易的事。为了便捷的解决这个问题，我们希望“一次设计，普遍使用”，让一张网页自动适应不同大小的屏幕，根据屏幕大小，自动调整布局。

1. 什么是响应式布局：

指可以自动识别屏幕宽度、并做出相应调整的网页设计。

1. 响应式布局开发：

允许网页宽度自动调整

"自适应网页设计"到底是怎么做到的？其实并不难。首先，在网页代码的头部，加入一行viewport元标签。

<meta name="viewport" content="width=device-width, initial-scale=1" />

viewport是网页默认的宽度和高度，上面这行代码的意思是，网页宽度默认等于屏幕宽度（width=device-width），原始缩放比例（initial-scale=1）为1.0，即网页初始大小占屏幕面积的100%。所有主流浏览器都支持这个设置，包括IE9。对于那些老式浏览器（主要是IE6、7、8），需要使用css3-mediaqueries.js。

<!--[if lt IE 9]>

<script src="http://css3-mediaqueries-js.googlecode.com/svn/trunk/css3-mediaqueries.js"></script>

<![endif]-->

不使用绝对宽度

由于网页会根据屏幕宽度调整布局，所以不能使用绝对宽度的布局，也不能使用具有绝对宽度的元素。这一条非常重要。具体说，CSS代码不能指定像素宽度：

　　width:xxx px;

只能指定百分比宽度：

　　width: xx%;

或者

　　width:auto;

相对大小的字体

字体也不能使用绝对大小（px），而只能使用相对大小（em）。

　　body {

　　　　font: normal 100% Helvetica, Arial, sans-serif;

　　}

上面的代码指定，字体大小是页面默认大小的100%，即16像素。

　　h1 {

　　　　font-size: 1.5em;

　　}

然后，h1的大小是默认大小的1.5倍，即24像素（24/16=1.5）。

　　small {

　　　　font-size: 0.875em;

　　}

small元素的大小是默认大小的0.875倍，即14像素（14/16=0.875）。

流动布局（fluid grid）

"流动布局"的含义是，各个区块的位置都是浮动的，不是固定不变的。

　　.main {

　　　　float: right;

　　　　width: 70%;

　　}

　　.leftBar {

　　　　float: left;

　　　　width: 25%;

　　}

float的好处是，如果宽度太小，放不下两个元素，后面的元素会自动滚动到前面元素的下方，不会在水平方向overflow（溢出），避免了水平滚动条的出现。另外，绝对定位（position: absolute）的使用，也要非常小心。

选择加载CSS

"自适应网页设计"的核心，就是CSS3引入的Media Query模块。它的意思就是，自动探测屏幕宽度，然后加载相应的CSS文件。

　　<link rel="stylesheet" type="text/css"

　　　　media="screen and (max-device-width: 400px)"

　　　　href="tinyScreen.css" />

上面的代码意思是，如果屏幕宽度小于400像素（max-device-width: 400px），就加载tinyScreen.css文件。

　　<link rel="stylesheet" type="text/css"

　　　media="screen and (min-width: 400px) and (max-device-width: 600px)"

　　　　href="smallScreen.css" />

如果屏幕宽度在400像素到600像素之间，则加载smallScreen.css文件。除了用HTML标签加载CSS文件，还可以在现有CSS文件中加载。

　　@import url("tinyScreen.css") screen and (max-device-width: 400px);

CSS的@media规则

同一个CSS文件中，也可以根据不同的屏幕分辨率，选择应用不同的CSS规则。

　　@media screen and (max-device-width: 400px) {

　　　　.column {

　　　　　　float: none;

　　　　　　width:auto;

　　　　}

　　　　#sidebar {

　　　　　　display:none;

　　　　}

　　}

上面的代码意思是，如果屏幕宽度小于400像素，则column块取消浮动（float:none）、宽度自动调节（width:auto），sidebar块不显示（display:none）。

图片的自适应（fluid image）

除了布局和文本，"自适应网页设计"还必须实现图片的自动缩放。这只要一行CSS代码：

　　img { max-width: 100%;}

这行代码对于大多数嵌入网页的视频也有效，所以可以写成：

　　img, object { max-width: 100%;}

### 2 功能卡片响应式布局开发指导：

1. 功能介绍：

（1）拖动改变大小时，三种布局之间的切换

（2）同一布局范围内大小变化时内部元素的自适应

大布局：当宽度逐渐缩小时，隐藏部分信息；当宽度逐渐变大时，逐步显示完整信息。中或小布局：当宽度逐渐变小时，显示数量下降；当宽度逐渐变大时，显示数量增加

2. 开发思路：

在上次实验中我们完成了功能卡片的拖拽功能。今天实现拖拽功能下显示内容响应式布局。通过两个函数来完成这个过程：dragAble(ele)和eleWidthChange(div)

目前功能卡片中内容由我们直接写在HTML元素中，不考虑从数据库读取数据。在这种情况下，只需要在拖拽改变大小过程中判断当前功能卡片大小是否达到布局变换临界值和实现响应当前宽度的自适应

3. 拖动函数添加判断条件

当前宽度大于600px时使用大布局；在300-600px之间时使用中等布局；小于300px时使用小布局。同时调用eleWidthChange()函数使显示内容和当前宽度适应。大中布局切换使用hide或show来控制隐藏或显示。中小布局切换时，将展示区的内容替换。可借鉴大中布局切换一样通过隐藏来实现，也可以通过JS动态添加HTML元素实验。

4. 编写eleWidthChange函数和中小布局CSS属性改变。

函数功能:当宽度到达某个临界值是隐藏或显示某列

参数:需要隐藏的表格所在的可拖动div

返回值:无

在拖动函数之后执行。获取当前元素宽度，表格信息行、信息卡片容器。判断表格布局隐藏列。从大到小变化，宽度到达变换值时隐藏某一列信息，到布局转变临界值时从大布局转变成中等布局，隐藏大布局显示中等布局，调用填充函数，设置功能卡片高度和最小高度。从小到大变化，宽度到达变换值时显示某一列信息。

5. 中小布局CSS属性改写。

在布局设计时使用浮动布局(float : left | right)使元素可以在信息卡片容器中浮动显示，自适应功能卡片大小显示合理数量的信息卡片，隐藏多余卡片。在eleWidthChange只需要根据功能卡片大小变化调整信息卡片容器大小和判断布局变化情况。中布局考虑变成大布局和小布局。转变成大布局时显示大布局，隐藏中小布局同时设置最低高度；转变成小布局时,设置最小高度。

## 实验后续任务