



任务1 主要产品 (8 小时)

1.1 预期效果



图 1-1 主要产品效果图

1.2 掌握技能

表 1-1 任务说明

任务名称	子任务	任务内容
主要产品	1	主要产品的 html+css 布局
	2	主要产品的 js 动画效果

1.3 任务分解

1. 界面的 html+css 布局

(1) 在轮播图代码后面，接着编写主要产品包裹层及内部第一层 html

```
<div class="mokuai chanpin">
```



```

<h1 style="margin-top:60px; line-height:60px;">主要产品</h1>
<h2>Our Works</h2>

<div class="content_wrap" id="chanping01"></div>
</div>
```

包裹层设置 Mokuai 类用于设置模块公用样式，. Chanpin 用于设置产品模块特殊样式

由于第一层上面的几个标签样式和页面下面的其他模块相似度很大，所以，只在 CSS 中定义公用样式，特殊样式直接写到标签内部

.content_wrap 为一个中部内容的包裹层，设置 id 是为了以后写 js 效果的时候用

(2) 设置样式

```
.mokuai{
    width:1100px;
    margin:0 auto;
}
.mokuai > img{
    margin:0 auto;
    display:block;
}
.mokuai > h1, .mokuai > h2{
    text-align:center;
    font-size:54px;
    color:#311759;
}

.mokuai > h2{
    font-family:Arial, Helvetica, sans-serif;
    font-weight:normal;
    font-size:18px;
    letter-spacing:6px;
}
.chanpin .content_wrap{
    height:540px;
    position:relative;
}
```



.mokuai 定义宽度和 margin 以设置居中

.mokuai > img 对第一层的 img 标签设置居中

设置 h1/h2 的文字居中, 字体大小, 颜色

h2 设置字体 (英文的, 用 arial 比较好看), 重置字体大小, 设置 letter-spacing 以设置每个字母间的间距。

.content_wrap 设置高度和相对定位 (后面的子集会相对于它去定位)。

(3) 在 .content_wrap 中加入左边步骤线的 html

```
<div class="now_line">
  <div class="now_linebtn_one now"><i></i><span></span></div>
  <div class="now_linebtn_one"><i></i><span></span></div>
  <div class="now_linebtn_one"><i></i><span></span></div>
  <div class="now_linebtn_one"><i></i><span></span></div>
  <div class="now_linebtn_one"><i></i><span></span></div>
  <div class="now_linebtn_one"><i></i><span></span></div>
</div>
```

一共 6 个步骤, 每个步骤使用了 i 和 span 两个标签, 一个现实步骤, 一个显示当前步骤。

(4) 设置步骤线的样式 step1 (样式设置 css 较多, 我们分成两步来写)

```
.chanpin .now_line{
  width:19px;
  margin-right:51px;
  float:left;
  height:380px;
  background:url(../images/chanpin_nowline_bgline.png) repeat-y 9px
  20px;
  margin-top:22px;
}
.chanpin .now_linebtn_one{
  text-align:center;
  height:11px;
  margin-bottom:63px;
}
.chanpin .now_linebtn_one span{
```



```
display:block;
width:11px;
height:11px;
margin:0 auto;
background:url(../images/chanpin_nowline_xiaoquan.png);
cursor:pointer;
}
```

.now_line 设置宽高浮动，设置 margin 值以设置距离。设置背景（那条又上至下的线）

.now_linebtn_one 设置每一个小点包裹层的样式，text-align:center 是为了让其内部的点左右居中显示。

对里面的 span 设置块儿显示，设置宽高和居中，设置背景。

(5) 设置步骤线的样式 step2

```
.chanpin .now_linebtn_one i{
display:none;
width:19px;
height:19px;
background:url(../images/chanpin_nowline_daquan.png);
cursor:pointer;
position:relative;
left:0px;
top:-4px;
float:left;
}
.chanpin .now_linebtn_one.now i{
display:block;
transition: all 0.5s ease 0s;
}
.chanpin .now_linebtn_one.now span{
display:none;
}
.chanpin .now_linebtn_one span:hover{
background:url(../images/chanpin_nowline_xiaoquan.png) 0 11px;
transition: all 0.5s ease 0s;
}
```



ps: 设置 i 的样式的时候, 为了方便查看, 可以先将 span 隐藏, 等编写 i 的样式, 在设置 span 的显示, 并隐藏 i。

对 i 设置宽高背景和定位, 之后设置隐藏

对当前步骤下的 i 设置显示, transition: all 0.5s ease 0s; 设置一个 css3 的过渡效果。显示隐藏很生硬。

对当前步骤下的 span 设置隐藏,

对 span 设置鼠标移入效果, 切换背景, 并设置过渡的 transition

(6) 在 .now_line 的同级加入内容的包裹层

```
<div class="contents">
  <div class="content_one" style="display:block;">
  </div>
  <div class="content_one">
  </div>
  <div class="content_one">
  </div>
</div>
```

包裹层为 .contents

里面的每个内容为 .content_one, 因为多个内容需要切换, 所有后面 css 会设置其隐藏, 这里在页面中设置第一个显示

(7) 设置样式

```
.chanpin .contents{
  width:1020px;
  float:left;
}
.chanpin .contents .content_one{
  display:none;
}
```

contents 设置宽度和浮动, 以和左边的步骤线排成一行



隐藏所有的 content_one

(8) 在第一个 .content_one 中写入内部内容

```
<div class="left_content"></div>
<div class="right_content">
  <h1>小鸟掌学<span class="pen"></span></h1>
  <div class="content_body"></div>
</div>
```

分为左侧和右侧两个部分，左侧为一个图，右侧内容包括一个头部和下面的主要内容

(9) 设置样式

```
.chanpin .left_content{
  float:left;
}
.chanpin .right_content{
  width:613px;
  float:right;
}

.chanpin .right_content > h1{
  color:#311759;
  font-size:30px;
  background:url(../images/pen_line.png) no-repeat left bottom;
  height:60px;
  position:relative;
}

.chanpin .right_content .pen{
  width:61px;
  height:54px;
  background:url(../images/pen.png) no-repeat;
  position:absolute;
  right:0px;
  bottom:-1px;
}

.chanpin .content_body{
  color:#565559;
```



```
font-weight:500;
letter-spacing:0px;
}
```

设置左右部分 content 宽度和浮动

h1 设置颜色, 字体大小, 背景, 高度, 和定位

对 h1 下边的 .pen 设置宽高, 背景和定位。

content_body 设置颜色, 字体加粗度, 字间距。

(10) . content_body 内部 html

```
<div class="p1">
  
  <h2>概述</h2>
  <p>小鸟掌学创建于2014年初, 起源服务于四川省专业技术人员继续教育基地的在线
  培训平台, 受到客户与社会的密切关注与欢迎。 卓新思创科技有限公司负责小鸟掌学的运
  营、推广和研发等工作。</p>
</div>
<div class="p2">
  <p>我们的愿景: 成为中国最强、最大的中小微企业定制在线学习服务提供商! </p>
  <p>我们做的是“新学习、新媒体”! </p>
  <p>帮助企业建立自己的企业学校, 实现: 全民全学、全民全校、全民全智( 比拼式学
  习)! </p>
</div>
<div class="tip">
  
  <div>学我所用、学以致用</div>
  
</div>
```

.tip 内的第一个 img 标签设置了 margin-top, 是为了设置距离 (这里先不设置, 等 css 设置的时候再添上)

(11) 设置样式, 分两步, setp1, 设置 p1 p2 的样式

```
.chanpin .content_body p{
  font-family:KaiTi_GB2312;
}
```



```
.chanpin .content_body .p1{
    overflow:hidden;
    margin:35px 0 90px 10px;
}
.chanpin .content_body .p1 *{
    float:left;
}
.chanpin .content_body .p1 img{
    margin-right:25px;
    margin-top:5px;
}
.chanpin .content_body .p1 h2{
    font-size:24px;
    width:470px;
}
.chanpin .content_body .p1 p{
    width:480px;
    line-height:24px;
}
.chanpin .content_body .p2 p{
    line-height:24px;
}
```

对段落设置字体 KaiTi_GB2312 (此楷体字体偏细, 小字体的时候更容易识别, 对于电脑上装有楷体的才有用, 没安装的就使用默认字体吧)

.p1 设置 overflow:hidden 是为了在子集均浮动的时候, 自身能保持相对高度。

.p1 *设置让 p1 的所有子集浮动。

.p1 img 设置 margin 来设置间距。

h2 设置字体, 宽度

p 设置宽度和行高

(12) 设置样式, 分两步, setp2, 设置 .tip 的样式

```
.chanpin .tip{
    margin-top:30px;
}
.chanpin .tip *{
```




```
float:right;
}
.chanpin .tip div{
margin-top:14px;
font-size:18px;
}
```

.tip 设置 margin 以设置间距，将其下的所有元素右浮动，以向右边排列
对其下边的 div 设置顶部的 margin 值和字体大小

(13) 写入其他.content_one 的 html 内容

```
//其他.content_one 的内容
```

由于内容太长，写到任务书里占篇幅太大，这里就不写代码了，大家把第一个.content_one 复制 5 份，按照示例的 html 修改里面的内容即可。
其中需要调整样式的地方，以行内样式编写。

(14) 在 .content_wrap 末尾加上左右切换按钮的 html

```
<div class="change_line">
    <span class="prev"></span>
    <span class="next"></span>
</div>
```

只有左右切换的按钮

(15) 设置样式

```
.change_line{
position:absolute;
left:53%;
```



```

margin-left:-108px;
top:232px;
z-index:20;
height:29px;
width:610px;
background:url(../images/next&prev_line.png) repeat-x left center;
}
.change_line > span{
position:absolute;
top:0;
height:29px;
width:29px;
background:url(../images/prev_jiantou.png) no-repeat;
cursor:pointer;
}
.change_line > .next{
right:0;
background:url(../images/next_jiantou.png) no-repeat;
}

```

. change_line 作定位，用 margin-left 以设置居中，设置 z-index 以浮动到表层，设置宽高和背景。

对前后箭头设置定位和背景。(hover 效果就不用设置了，因为我们做 banner 轮播的时候设置的前后箭头的 hover 效果可以在这里被共用)

主要产品的 html + css 编写完成！

2. 主要产品的 js 动画效果

(1) 在 index.js 的 \$(function() {}) 中写入代码，获取主要产品全部的 DOM

```

(function(){
    var oDiv = $('#chanping01');
    var oPrev = oDiv.find(".prev");
    var oNext = oDiv.find(".next");
    var aDot = oDiv.find(".now_linebtn_one");
    var aSpan = oDiv.find(".now_line span");
    var aContents = oDiv.find(".content_one");
    var nowIndex = 0;
    //本效果后续代码写在此处

```



```
})();
```

为了避免和其他代码之间变量的影响，这里将轮播的 js 写成函数自执行的形式

```
(function() {
```

```
//此处执行代码
```

```
})();
```

oDiv 是最外面的 banner 包裹层 DOM

oPrev 是前一页 DOM

oNext 是下一页 DOM

aSpan 是左侧切换位置的小点 DOM

aDot 是切换位置小点的父级 DOM

aContents 是中间需要切换的模块的合集

nowIndex 代表当前要显示的 content_one 的 index，初始值为 0;

(2) 点击下一页

```
oNext.click(function(){
    nowIndex++;
    if(nowIndex>=aDot.length){
        nowIndex = 0;
    }
    doFade("fadeInRight");
});
```

1、每点击一次，nowIndex 增加 1，如果超出了最大长度，就把 nowIndex 设置为 0，让第一个 content_one 显示。

2、执行运动函数，运动函数下一步中编写。传入参数“fadeInRight”是从右进入 css3 动画的 class

(3) doFade 方法

```
function doFade(action){

    aDot.removeClass("now").eq(nowIndex).addClass("now");
    aContents.fadeOut(0).eq(nowIndex).fadeIn(200);
```



```
aContents.eq(nowIndex).find("h1, p,
img").attr("class", "").addClass("animated "+action);
}
```

- 1、aDot 设置当前项
- 2、aContents 显示当前项
- 3、设置要显示的 aContents 里面的 h1, p, img 的动画效果, 先清除其 class, 在设置 css3 动画类。(先清除 class 的意义为: 如果第二次切换到此 content 的时候, 以前加的 css3 动画类会互相影响, 先清除就不会影响了)

(4) 点击上一页

```
oPrev.click(function(){
    nowIndex--;
    if(nowIndex<0){
        nowIndex = aDot.length-1;
    }
    doFade("fadeInLeft");
});
```

- 1、每点击一次, nowIndex 减 1, 如果小于 0, 就把 nowIndex 设置为最大的 index, 让最后一个 content_one 显示。
- 2、执行运动函数。传入参数 “fadeInLeft” 是从左进入 css 动画的 class

(5) 点击左侧切换位置的小点

```
aSpan.click(function(){
    var index = aDot.index($(this).parent());
    var action = (nowIndex>index)?"fadeInLeft":"fadeInRight";
    nowIndex = index;
    doFade(action);
});
```

- 1、获取当前点击的点的父级在其兄弟节点中的 index, 此 index 就是需要显示的项的 index。



2、取得动画执行方向，如果点击的是原来项前边的点，即 $\text{nowIndex} > \text{index}$ ，则让动画从左进入，否则从右进入。

3、赋值 nowIndex 并执行动画函数 Fade

1.4 参考资料

无

1.5 扩展练习

教师自行补充