1. 第一天
   1. git的使用：版本控制工具，可以进行团队开发



* 1. html：是用来描述网页的一种语言。
* HTML 指的是超文本标记语言 (**H**yper **T**ext **M**arkup **L**anguage)
* HTML 不是一种编程语言，而是一种**标记语言** (markup language)
* 标记语言是一套**标记标签** (markup tag)
* HTML 使用**标记标签**来描述网页

HTML 标记标签通常被称为 HTML 标签 (HTML tag)。

* HTML 标签是由**尖括号**包围的关键词，比如 <html>
* HTML 标签通常是**成对出现**的，比如 <b> 和 </b>
* 标签对中的第一个标签是**开始标签**，第二个标签是**结束标签**
* 开始和结束标签也被称为**开放标签**和**闭合标签**

1. 第二天
2. css: 选择器

选择器种类：

标签选择器

分组选择器

ID选择器

Class选择器

属性选择器

后代选择器

伪类选择器

选择器的优先级：!important > 内联 > id > class > tag|伪类|属性 > 通配符

选择器的权重值：内联 1000，id 100，class 10|伪类 10|属性 10，标签 1，通配符0，权重值相同的情况下，后面定义的样式会覆盖前面的，不同的会层叠

1. 水平导航菜单
2. HTML：
   1. ul无序列表
   2. li列表
   3. a链接：anchor, href属性是用来指代链接的地址
   4. href，a的属性，用来指定链接的地址
   5. 其它一些重要知识点：
   6. doctype：文档类型，document type，必须要写，否则可能出现兼容性问题
   7. 标签(tag)：也称之为元素，标签是成对的
   8. utf-8：统一字符集，一共有65536个字符
3. 第三天：多级导航菜单

javascript：

1. 变量

* 使用关键字var声明一个变量
* 变量的名称是区分大小写的
* 第一个字符必须是字母、下划线（\_）或美圆符号（$）
* 余下的字符可以是下划线、美圆符号或任何字母或数字字符。

1. 获取元素：
   1. getElementById()根据元素的id查找元素，返回一个dom对象
   2. getElementsByTagName()根据元素的标签名查找元素，返回数组
   3. 一个好的变量命名习惯，对象用o开头，字符串用s开头，数组用a开头等等
   4. 用js修改元素的样式，通过element.style.xxx = yyy来进行，其中xxx是css的属性，命名规范和css的命名规范几乎一样，除了将css中的"-"去掉，后面的字母变大写，如：background-color => backgroundColor
   5. js中所有的事件都是以on开头，如：onclick单击，ondblclick双击，onmouseover鼠标划过，onmouseout鼠标划出等
   6. this是对当前对象的引用，this出现在哪个对象的方法中，this就指向这个对象
2. 第四天：折叠菜单

CSS：

1. 选择器：
   1. 标签(tag)选择器：如 ul等
   2. id选择器：如#main-menu
   3. \*通配符，选择所有元素
   4. 后代选择器 如：#main-menu h3

JavaScript：

1. 获取元素：
   1. getElementById()根据元素的id查找元素，返回一个dom对象
   2. getElementsByTagName()根据元素的标签名查找元素，返回数组
   3. this是对当前对象的引用，this出现在哪个对象的方法中，this就指向这个对象

 核心代码：

<**script**>

**var *oMainMenu*** = **document**.getElementById(**'main-menu'**);

**var *aH3*** = ***oMainMenu***.getElementsByTagName(**'h3'**);

**var *aSubMenu*** = ***oMainMenu***.getElementsByTagName(**'ul'**);

*// var bShow = false;*

**for**(**var *i***=0; ***i***<***aH3***.**length**; ***i***++){

***aH3***[***i***].**index** = ***i***;

***aH3***[***i***].**bShow** = **false**;

***aH3***[***i***].onclick = **function**(){

**if**(**this**.**bShow**){

***aSubMenu***[**this**.**index**].**style**.**display** = **'none'**;

}**else**{

***aSubMenu***[**this**.**index**].**style**.**display** = **'block'**;

}

**this**.**bShow** = !**this**.**bShow**;

};

}

</**script**>

1. 第五天：选项卡
2. 知识点
   1. HTML： ul无序列表
      1. li列表
      2. div可定义文档中的分区或节

1. CSS
   1. 选择器： id选择器：如#container
      1. #tab li 后代选择器
      2. #container{margin: 100px auto; }
         1. margin:100px auto 上下margin值是100px; 左右居中显示
         2. margin：0 auto；不居中可能有以下两个的原因:
            1. 没有设置宽度
            2. 没声明DOCTYPE
      3. cursor: pointer; 鼠标滑过为手状;
2. JavaScript
   1. 获取元素：
      1. 用js修改元素的class，通过element.className = xxx 来进行，其中xxx是class的名
      2. innerHTML 属性设置或返回标签的 HTML 。
      3. js中所有的事件都是以on开头，如：onclick单击，ondblclick双击，onmouseover鼠标划过，onmouseout鼠标划出等
      4. children 非标准属性，它返回指定元素的子元素集合。但它只返回HTML节点，甚至不返回文本节点，虽然不是标准的DOM属性，但它和innerHTML方法一样，得到了几乎所有浏览器的支持。

核心代码：

**var** oMenu = document.getElementById("menu");  
**var** aLi = oMenu.getElementsByTagName("li");  
**var** oContent = document.getElementById("content");  
**var** aDiv = oContent.getElementsByTagName("div");  
**for**(**var** i=0; i<aLi.length; i++){  
 aLi[i].index = i;//给每个li对象自定义属性index，作用是保存住当时的i值，其实就是索引  
 aLi[i].onclick = **function**(){  
 **for**(**var** i=0; i<aLi.length; i++){  
 aLi[i].className = "";  
 aDiv[i].className = "";  
 }  
 **this**.className = "selected";  
 aDiv[**this**.index].className = "selected";  
 };

1. 第六天：轮播图
2. 知识点
   1. HTML：

ul无序列表

* + 1. li列表
    2. img标签
    3. span标签
    4. div标签
    5. href，a的属性，用来指定链接的地址
  1. CSS
     1. 选择器
     2. display属性
     3. z-index属性
     4. cursor: pointer;鼠标变手状
     5. position属性

1. 选择器：
   1. 标签(tag)选择器：如h1, p, ul等
   2. id选择器：如#menu
   3. \*通配符，选择所有元素
   4. \*{margin:0; padding:0;}是样式重置，去掉所有元素的内外边距，因为不同的浏览器的内外边距可能不统一
   5. ul,ol{list-style:none;}去掉列表的小点样式
   6. a{text-decoration:none;}去掉a的默认下划线
   7. 将line-height行高设置为和height一样大小，可以实现单行文字垂直居中效果
   8. 元素浮动后空间释放，没有元素撑着父元素，父元素就没有高度了，这就是经典的塌陷问题，解决这个问题的办法有：
      1. 父元素overflow:hidden;
      2. 父元素也float;
      3. 给父元素直接加高度
      4. 在浮动的元素后面插入一个空div，设置div的clear:both;
      5. 添加一个伪类：.clearfix（这种方法ie6不支持after）

.clearfix:after{

content: '';

display: block;

clear: both;

overflow: hidden;

\*zoom:1;

}

1. JavaScript
   1. 利用for循环给多个dom对象加事件
   2. 函数定义，函数参数，函数调用
   3. setInterval定时器应用和清除
   4. dom对象的children属性
2. 第七天：淡入淡出轮播图
3. 知识点
   1. HTML：
      1. ul无序列表
      2. li列表
      3. img标签
      4. span标签
      5. div标签
   2. CSS
      1. 选择器：
      2. display属性
      3. cursor: pointer;鼠标变手状
      4. position属性
      5. opacity,透明度属性,ie浏览器需要利用filter: "alpha(opacity=0)"做兼容性;

lmg . style. opacity 
olmg. style. filter 
'alpha(opacity 0) 

1. JavaScript
   1. 利用for循环给多个dom对象加事件
   2. 函数定义，函数参数，函数调用
   3. setInterval定时器应用和清除
   4. 元素的offsetWidth属性，指dom对象自身的宽度，整型，单位像素。
   5. dom对象的children属性
2. 第八天：无缝滚动
3. 关键点

要实现无缝滚动，需要在四个div的后面再复制一遍四个div

1. 知识点
   1. HTML：
      1. ul无序列表
      2. li列表
   2. CSS
      1. [定位](onenote:#CSS&section-id={6E40BDA6-BD68-3345-87F1-F7DD70C7044E}&page-id={CE51A6CE-E960-CD4F-B4C8-38C0C4A85C18}&object-id={64C610FD-CEE0-4C49-9E82-C78C7FBCF8B9}&18&base-path=https://d.docs.live.net/2d6214c820ba0730/文档/前端课程体系/基础知识.one)：父元素先相对定位，子元素再绝对定位
      2. [Display](onenote:#CSS&section-id={6E40BDA6-BD68-3345-87F1-F7DD70C7044E}&page-id={CE51A6CE-E960-CD4F-B4C8-38C0C4A85C18}&object-id={004828B0-3771-9607-378F-A294F56C3E17}&C&base-path=https://d.docs.live.net/2d6214c820ba0730/文档/前端课程体系/基础知识.one):inline-block，可以让几个div横过来，但问题是中间会产生空隙，这个空隙是由于div之间的换行产生的。解决办法：1、手动删掉div中间的换行和空格2、浮动
      3. 父元素overflow:hidden 移除隐藏
      4. div中文字水平和垂直居中：text-align:center line-height:div高度
   3. JavaScript
      1. 定时器：setInterval()和setTimtout()的区别
      2. 复制div时，使用innerHTML代替append，这样效率更高
      3. 鼠标的划入事件onmouseover
      4. 左右方向使用一个变量的正负值来控制
2. 第九天：无缝滚动的轮播图
3. 知识点
   1. HTML：
      1. ul无序列表
      2. li列表
      3. img标签
      4. span标签
      5. div标签
   2. CSS
      1. 选择器：
      2. display属性
      3. cursor: pointer;鼠标变手状
      4. position属性
   3. JavaScript
      1. 元素的offsetWidth属性，指dom对象自身的宽度，整型，单位像素。通过.style.width可以设置宽度，但是只能获取获取出内联样式的width

切换图片的方法：

**function** *changeImg*(index){

**for**(**var** i=0; i<***aLi***.**length**; i++){

***aLi***[i].**className** = **''**;

}

*// elem.className = 'selected';*

***aLi***[index].**className** = **'selected'**;

animate(***oContent***, {**left**: -index\****aImg***[0].**offsetWidth**});

}

1. 第十天：拖拽
2. 知识点
   1. HTML：
      1. div标签
   2. CSS
      1. 选择器：
      2. position属性
   3. JavaScript
      1. 事件onmousedown，onmousemove，onmouseup
      2. 鼠标事件event,兼容IE(e = e || window.event)
      3. 通过event获取鼠标点击坐标（e.clientX,e.clientY）
      4. offsetLeft:元素到浏览器左边的距离，整型，单位像素
      5. offsetTop:元素到浏览器上边的距离，整型，单位像素
      6. 获取浏览器的宽度（document.documentElement.clientWidth）
      7. 获取浏览器的高度（document.documentElement.clientHeight）

核心代码：

<**script**>

**var *oDiv*** = **document**.getElementById(**'div1'**);

***oDiv***.onmousedown = **function**(e){

e = e || **window**.**event**;

**var** disX = e.**clientX** - **this**.**offsetLeft**;

**var** disY = e.**clientY** - **this**.**offsetTop**;

*//alert(disX+","+disY);*

**document**.onmousemove = **function**(e){

e = e || **window**.**event**;

**var** ileft = e.**clientX** - disX;

**var** itop = e.**clientY** - disY;

**if**(ileft <= 0){

ileft = 0;

}

**if**(ileft >= **document**.**documentElement**.**clientWidth** - ***oDiv***.**offsetWidth**){

ileft = **document**.**documentElement**.**clientWidth** - ***oDiv***.**offsetWidth**;

}

**if**(itop <= 0){

itop = 0;

}

**if**(itop >= **document**.**documentElement**.**clientHeight** - ***oDiv***.**offsetHeight**){

itop = **document**.**documentElement**.**clientHeight** - ***oDiv***.**offsetHeight**;

}

***oDiv***.**style**.**left** = ileft + **'px'**;

***oDiv***.**style**.**top** = itop + **'px'**;

};

***oDiv***.onmouseup = **function**(){

**document**.**onmousemove** = **null**;*//取消事件*

};

}

</**script**>

1. 第十一天：自定义滚动条
2. 知识点
   1. HTML：
   2. CSS
      1. 选择器：
         1. overflow: ;当内容溢出元素框时发生的事情，可能的值：

 
visible 
hidden 
scroll 
auto 
inherit 
描 述 
默 认 值 。 内 容 不 会 被 修 剪 ， 会 呈 现 在 元 素 框 之 外 。 
内 容 会 被 修 剪 ， 并 且 其 余 内 容 是 不 可 见 的 。 
内 容 会 被 修 剪 ， 但 是 浏 览 器 会 显 示 滚 动 条 以 便 查 看 其 余 的 内 容 。 
如 果 内 容 被 修 剪 ， 则 浏 览 器 会 显 示 滚 动 条 以 便 查 看 其 余 的 内 容 。 
规 定 应 该 从 父 元 素 继 承 overflow 属 性 的 值 。 

1. JavaScript
   1. 事件onmousedown（鼠标按下），onmousemove（鼠标移动），onmouseup（鼠标抬起）
   2. 事件event, ie浏览器通过window.event获取，兼容IE写法(e = e || window.event)
   3. 通过event获取鼠标点击坐标（e.clientX,e.clientY）,相对于浏览器左上角
   4. 元素的offsetLeft与offsetTop属性返回元素在页面中相对于父元素的坐标
   5. offsetHeight属性用来取得对象在页面中的实际所占区域高度
   6. onselectstart='return false;' 阻止默认的选中状态
2. 第十二天：放大镜
3. 关键点
   1. 在小图片上面覆盖一个透明的div，把事件绑定到该div上
   2. 找出小图和大图之间的比例关系

1. 知识点
   1. HTML：
   2. CSS
      1. [定位](onenote:#CSS&section-id={6E40BDA6-BD68-3345-87F1-F7DD70C7044E}&page-id={CE51A6CE-E960-CD4F-B4C8-38C0C4A85C18}&object-id={64C610FD-CEE0-4C49-9E82-C78C7FBCF8B9}&18&base-path=https://d.docs.live.net/2d6214c820ba0730/文档/前端课程体系/基础知识.one)：父元素先相对定位，子元素再绝对定位
      2. Z-index 定位后调整层级
      3. 父元素overflow:hidden 移除隐藏
      4. div中文字水平和垂直居中：text-align:center line-height:div高度
      5. 透明度兼容性：

**opacity**:0;

**filter**:**alpha**(**opacity**=0);

1. JavaScript
   1. 鼠标事件：onmouseover、onmouseout、onmousemove
   2. 事件：e=e||event，通过e.clientX和e.clientY获取当前鼠标的坐标
   3. js中控制样式：obj.style.xx
   4. 改变top和left的值时要先定位

*//定位大图片*

**var** iScaleX = (e.**clientX** - ***oDrag***.**offsetWidth** / 2) / (***oSmallPic***.**offsetWidth** - ***oDrag***.**offsetWidth**);

**var** iScaleY = (e.**clientY** - ***oDrag***.**offsetHeight** / 2) / (***oSmallPic***.**offsetHeight** - ***oDrag***.**offsetHeight**);

***oBigImg***.**style**.**left** = -iScaleX \* (***oBigImg***.**offsetWidth** - ***oBigPic***.**offsetWidth**) + **'px'**;

***oBigImg***.**style**.**top** = -iScaleY \* (***oBigImg***.**offsetHeight** - ***oBigPic***.**offsetHeight**) + **'px'**;

1. 第十三天：图片轮换播放

在轮播图基础上实现一个更加复杂的轮播图，上面是8张大图片淡入淡入切换，下面对应8个缩略图，中间会显示总的图片数量以及现在显示图片的索引。

使用到animate（）函数，该函数共三个参数：做动画的对象、做动画的属性、动画做完时的回调函数。

核心代码：

**function** changImg(){  
 //当前要是图片淡入淡出方式显示  
 aBigImgs[iNow].style.opacity = 0;  
 aBigImgs[iNow].style.filter = "alpha(opacity=0)";  
 aBigImgs[iNow].style.zIndex = zIndex++;  
 oInfo.style.zIndex = oPrev.style.zIndex = oNext.style.zIndex = zIndex++;  
 animate(aBigImgs[iNow], {  
 opacity : 100  
 });  
  
 //info里面的信息切换  
 oCurrentPage.innerHTML = iNow + 1;  
  
 //小图片所有都透明，当前图片不透明  
 **for**(**var** i=0; i<aSmallImgs.length; i++){  
 aSmallImgs[i].style.opacity = 0.3;  
 aSmallImgs[i].style.filter = "alpha(opacity=30)";  
 }  
 aSmallImgs[iNow].style.opacity = 1;  
 aSmallImgs[iNow].style.filter = "alpha(opacity=100)";  
  
 **var** left = 0;  
 **if**(iNow == 0 || iNow == 1){  
 left= 0;  
 }**else if**(iNow == aSmallImgs.length - 1 || iNow == aSmallImgs.length - 2){  
 left = aSmallImgs.length / 2;//4  
 }**else**{  
 left = iNow - 1;  
 }  
 animate(oSmallImg, {  
 left: -left \* aSmallImgs[0].offsetWidth  
 });  
  
}

1. 第十四天：定时器

在javascritp中，有两个关于定时器的专用函数，分别为：

1.倒计定时器：timename=setTimeout("function();",delaytime);  
2.循环定时器：timename=setInterval("function();",delaytime);

第一个参数“function()”是定时器触发时要执行的动作，可以是一个函数，也可以是几个函数，函数间用“；”隔开即可。比如要弹出两个警告窗口，便可将“function();”换成  
“alert('第一个警告窗口!');alert('第二个警告窗口!');”；而第二个参数“delaytime”则是间隔的时间，以毫秒为单位，即填写“5000”，就表示5秒钟。  
　　倒计时定时器是在指定时间到达后触发事件，而循环定时器就是在间隔时间到来时反复触发事件，两者的区别在于：前者只是作用一次，而后者则不停地作用。  
比如你打开一个页面后，想间隔几秒自动跳转到另一个页面，则你就需要采用倒计定时器“setTimeout("function();",delaytime)” ，而如果想将某一句话设置成一个一个字的出现，  
则需要用到循环定时器“setInterval("function();",delaytime)” 。

setTimeout("function",time) 设置一个超时对象

setInterval(expression,delayTime),每个DelayTime,都将执行Expression.常常可用于刷新表达式.  
setInterval("function",time) 设置一个超时对象

SetInterval为自动重复，setTimeout不会重复。

clearTimeout(对象) 清除已设置的setTimeout对象  
clearInterval(对象) 清除已设置的setInterval对象

1. 第十五天：复习：
2. html常见标签

常见标签的用法， 具体用法如下：

<head>标签的子标签

<title>标签：设置标题，设置完成后可以显示到IE窗口的标题上

<style>标签：引入css文件

<script>标签：引入[JavaScript](http://lib.csdn.net/base/javascript)文件

标题字体-<h1~6>

可以取值1,2,3,4,5,6

标题显示为黑体字

这些标记自动插入一个空行，不必用 <p> 标记再加空行

数字越小字体越大

size—1,2,3,4,5,6,7

color—字体颜色， color为 16 进制 RGB 数字，或者是下列预定义色彩,blue/yellow/red……

文字布局：

<div>…</div> <span>…</span>

文字的分区显示2个div之间会自动换行

<ul><li>…</ul>

无序号列表

<ol><li>…</ol>

有序号列表

图片-<img>:

src--图片路径，最好使用相对路径

alt--当鼠标放到图片上显示出提示信息

width—图片宽度，如果不指定按原始宽度

height—图片高度，如果不指定按原始高度

CSS：

1. 盒模型（box model）

eight 
border 
margin 
padding 
element 
W3School.com.cn 

盒模型：

1. w3c标准盒模型：width=content
2. 定位
   1. relative 相对定位，相对于自己的初始位置 空间不释放

op : 20px 
left:30px 

1. absolute 绝对定位，相对于最近的已定位祖先元素，如果元素没有已定位的祖先元素，那么它的位置相对于最初的包含块(body)

op:20px 
left:3üpx 
框 2 
相 对 定 位 的 袓 兂 元 素 

1. fixed  固定定位 ,相对于浏览器 ，ie6不支持
2. static 默认定位方式 无特殊定位，对象遵循正常文档流

1. 浮动

|  |  |
| --- | --- |
| **值** | **描述** |
| left | 元素向左浮动。 |
| right | 元素向右浮动。 |
| none | 默认值。元素不浮动，并会显示在其在文本中出现的位置。 |
| inherit | 规定应该从父元素继承 float 属性的值。 |

例子：水平导航菜单

ink one 
Link two 
Link three 
Link four 

1. display,常用的几个：

|  |  |
| --- | --- |
| **值** | **描述** |
| none | 此元素不会被显示。 |
| block | 此元素将显示为块级元素，此元素前后会带有换行符。 |
| inline | 默认。此元素会被显示为内联元素，元素前后没有换行符。 |
| inline-block | 行内块元素。（CSS2.1 新增的值） |

1. 行级元素和块级元素的区别

1.行内元素与块级元素直观上的区别

    行内元素会在一条直线上排列，都是同一行的，水平方向排列

    块级元素各占据一行，垂直方向排列。块级元素从新行开始结束接着一个断行。

2.块级元素可以包含行内元素和块级元素。行内元素不能包含块级元素。

3.行内元素与块级元素属性的不同，主要是盒模型属性上

行内元素设置width无效，height无效(可以设置line-height)，margin上下无效，padding上下无效

1. 第十六天：复习js
2. 变量

* 使用关键字var声明一个变量
* 变量的名称是区分大小写的
* 第一个字符必须是字母、下划线（\_）或美圆符号（$）
* 余下的字符可以是下划线、美圆符号或任何字母或数字字符。

1. 值，类型

基本数据类型：number\string\boolean\undefined\numm

3数学运算与比较

加法 +   
减法 -  
乘法 \*  
除法 /   
自增 ++  
自减 --

要注意的是，在JavaScritp中，连接字符串时也使用“＋”号。当字符串与数字相遇时，数字都会当成字符串连接

小于 <  
大于 >  
小于或等于 <=  
大于或等于 >=  
相等 ==  
不相等 !=

1. 分支判断与循环

分支

* 单一选择结构（if）
* 二路选择结构（if/else）
* 内联三元运算符 ?:
* 多路选择结构（switch…case）

循环

* 由计数器控制的循环（for）
* 在循环的开头测试表达式（while）
* 在循环的末尾测试表达式（do/while）
* 对对象的每个属性都进行操作（for/in）

1. 函数与内置对象

函数

* 函数定义
* 函数参数
* 函数调用
* 函数返回值

作用域

* 全局作用域
* 局部作用域

内置对象

* Math
* Date
* String
* Array

1. 第十七天：js练习题

通过一套js习题来考察之前的学习成果，通过习题也学到了很多新的知识点。

比如斐波那契数列的两种写法：

/\*  
\*  
\*7. 斐波纳契（Fibonacci）数列的第一项是1，第二项是1，  
\* 以后各项都是前两项的和。试用递归函数和非递归函数各编写一个程序，求斐波纳契数列第N项的值。  
 1 1 2 3 5  
 \* \*/  
  
**function** fibonacci1(num){  
 **if**(num == 1 || num == 2){  
 **return** 1;  
 }  
 **return** fibonacci1(num - 1) + fibonacci1(num - 2);  
}  
  
**function** fibonacci2(num){  
 **var** arr = [1, 1];  
 **for**(**var** i=2; i<num; i++){  
 arr[i] = arr[i-1] + arr[i-2];  
 }  
 **return** arr[num-1];  
}

随机产生一定范围内的整数的函数：

**function** getRangeRandom(low, high){  
 **return** Math.floor(Math.random() \* (high - low + 1)) + low;  
}

js内置对象中常用的函数的使用：

* [Array](itss://chm/www.w3school.com.cn/jsref/jsref_obj_array.asp)
* [Date](itss://chm/www.w3school.com.cn/jsref/jsref_obj_date.asp)
* [Math](itss://chm/www.w3school.com.cn/jsref/jsref_obj_math.asp)

[String](itss://chm/www.w3school.com.cn/jsref/jsref_obj_string.asp)

1. 第十八天:前端开发流程：

老板或甲方是一个需求的真正发起者，也是一个基础idea的梦想师，产品是需求专业化梳理或进行有效评估细化需求负责的，  
而设计是前端的上游，前端是设计的下游。设计的工作目的是把产品宏观的思维结果进行专业的处理，因为按一般的习惯，产品最终的结果是原型图，而原型图可以理解为设计的草图，  
对真正的用户来说，这个草图过于简单或不符合使用的操作习惯，所以需要设计师进行专业的处理，比如颜色搭配，布局分隔，有时候还兼交互的一部分工作，设置用户与页面发生交互的预订流程，

这里边有一个成本风险控制的一个理念，因为在前期，尤其是设计，主观感受大于理性的思考，所以每天的结果都不一样，所以需要设计师去消化掉这部分主观感受带来的误区，  
而且从成本上来讲，有些场景设计师改图比改代码要容易控制一些。  
设计师的结果是psd文件，他是很多个图层叠加在一起的结果，而前端的工作结果html页面，是把很多图层上的效果，有机的用html组织起来的过程。  
前端是把转化后html交给下游服务端开发工程师，或叫后台开发，这个html里边包括一些交互的js文件等。总的来说前端是一个承前启后的岗位。

基本的流程是，领导或甲方提出需求，然后产品分析需求，并且根据需求画出原型图，然后根据原型图出设计稿。  
出完设计稿团队评审，过后交与前端制作静态页面，然后静态页面，交与设计审核，过后交给开发人员，进行动态数据的添加。  
添加完之后，发布测试环境，产品测试领导审核，成功后，直接发布产品环境。或进行版本迭代。  
这是整个的一个设计，开发，部署的流程。

1. 第十九天：整站开发

css精灵：

css sprites直译过来就是**CSS精灵**。通常被解释为“**CSS图像拼合**”或“**CSS贴图定位**”。其实就是通过将多个图片融合到一张图里面，然后通过[CSS background](http://www.divcss5.com/rumen/r125.shtml)背景定位技术技巧布局网页背景。这样做的好处也是显而易见的，因为图片多的话，会增加http的请求，无疑促使了网站性能的减低，特别是图片特别多的网站，如果能用css sprites降低图片数量，带来的将是速度的提升。

1、适合：一般小图标素材  
小的图标ico类素材，一般图标很小十多像素几十像素的宽度高度，这种适合拼合成一张图实现sprites background背景定位布局。多小ico太多自然加载网页时瞬间会消耗些http iis链接数，但很快加载完又会释放。

2、不适合：大图大背景  
大背景一般用于网页背景，拼合时，设置为网页背景时所有背景都会显示出来。大图拼接拼合会增大图片大小，网络带宽不好的访问者访问时由于背景图大文件大会加载稍慢些，所以大图不推荐拼接拼合来使用css sprites背景定位布局。

3、sprites适合推荐小结  
一般此sprites拼合布局用于局部小盒子布局不适合大背景大布局背景使用。比如小局部布局小图标背景、小导航背景等DIVCSS布局。

1. 第二十天：浏览器兼容性问题

所谓的浏览器兼容性问题，是指因为不同的浏览器对同一段代码有不同的解析，造成页面显示效果不统一的情况。在大多数情况下，我们的需求是，无论用户用什么浏览器来查看我们的网站或者登陆我们的系统，都应该是统一的显示效果。所以浏览器的兼容性问题是前端开发人员经常会碰到和必须要解决的问题。

* 1. png图片在IE6下不透明

解决办法：引入DD\_belatedPNG\_0.0.8a.js文件，DD\_belatedPNG.fix('\*');

<!—[if IE 6]>

<script src=”DD\_belatedPNG\_0.0.8a.js”></script>

<script>

DD\_belatedPNG.fix(‘\*’);

</script>

<![endif]-->条件级hack，解决IE6下面不支持png透明问题

* 1. css透明度的写法

标准浏览器(0-1)opacity:0.2，

IE (0-100);filter:  alpha(opacity=20);

* 1. position:fixed在IE6下不支持

解决办法：

**window**.onscroll=**function**(){

**var** iScrollTop=**document**.**documentElement**.**scrollTop**||**document**.**body**.**scrollTop**;

***oContainer***.**style**.**top**=iScrollTop+200+**'px'**;

};

* 1. js获取滚动条高度：**document**.**documentElement**.**scrollTop**||**document**.**body**.**scrollTop**
  2. IE6中浮动式产生一个双边距的问题：把元素的display:inline
  3. IE6只支持 a标签的：hover，不支持其他标签的：hover
  4. 获取事件源：

var target = e.target || window.event.srcElement;

* 1. css3兼容性：

Firefox: -moz-

Safari: -webkit-

Opera: -o-

IE: -ms-

* 1. 条件注释

<link rel="stylesheet" type="text/css" href="css.css" />

<!--[if IE 7]>

<link rel="stylesheet" type="text/css" href="ie7.css" />

<![endif]-->

<!--[if lte IE 6]>

<link rel="stylesheet" type="text/css" href="ie.css" />

<![endif]-->

* 1. **<!DOCTYPE HTML>文档类型的声明。**

**产生条件：IE6浏览器，当我们没有书写这个文档声明的时候，会触发IE6浏览器的怪异解析现象；**

**解决办法：书写文档声明。**

* 1. **img外部的border**

**产生条件：img外部有a标签，即img标签有链接时**

**解决办法：设置img边框border:0;**

* 1. **子代选择器在IE6中不能使用**

**产生条件：IE6浏览器，使用E>F子代选择器；**

**解决办法：采用其他选择器或者采用后代选择器进行控制，如：div p{margin:10px;} div p p{margin:0;}替代掉 div>p{margin:10px;}。**