# Task1

## 新标签

在学习html5的过程中，知道了一些新的标签，能够更清晰的表达文档的结构（原来都是用div，可能加上id或者class来划分）。举个简单的例子

* html
  + head
    - body
  + header (包含网站的标题，或者logo，SLOGAN之类的)
  + nav （导航栏）
  + main (文档的主要内容，不包含侧边栏、导航栏、版权信息，网站logo等附属信息)
    - article （文档中可以脱离其他部分，独立出来而又完整，甚至可以复用的一部分，通常有自己的标题，当article内嵌article时，里外层的内容应该是相关的，比如一篇微博和它的评论）
    - section (文档中一段主题性内容，通常也有自己的标题，跟article的区别在于他是整体的一部分或者说文章的一节)
  + aside （侧边栏或者嵌入内容，通常认为是独立拆分出来而不受整体影响的一部分，作为主要内容的附属信息，如索引，词条列表，或者页面及站点的附属信息，如广告，作者资料介绍等）
  + footer （页脚，通常包含作者、版权信息或者相关链接等）

以下标签语义化的表示章节结构，在HTML5之前都是用div标签进行分隔  
header 页面头部／标题／logo／搜索表单 等等  
nav 任何有导航性质的标签都可以用到，比如页面中的导航，可以有多个  
aside 和主要内容不相关的内容，工具内容／广告什么的  
article 表示独立的可重复的结构：论坛的帖子／博客的评论  
section 表示文档中的一个区域：一般会带一个标题　相临的section之间是有相关性的，article则不相关，是独立的  
footer 章节的尾部：版权信息／相关文档／一些独立内容的尾部也可以用到

## 打开新窗口：

<a href="[http://www.baidu.com](http://www.baidu.com/)" target="\_Blank">百度</a>  
\_Blank是新窗口  
\_Self是自身  
\_Parent是父窗口  
\_Top是顶层窗口

.空格：在html中输入空格是没有作用的，需要输入&nbsp这个代码

## 表格：

加边框<table border="1"> </table>

表格的跨行和跨列：colspan rowspan

colspan 属性规定单元格可横跨的列数。

## 表单：

<input type="radio" checked="checked"/>

<input type="submit" value="确认提交">

label标签，为input元素定义标注，改进了表单控件的可用性，当你点击到label标签时，会自动聚焦到对应控件上。

<**label**>请输入用户名<**input** **type**="text" id="username" name="username"></**label**>

<**input** **type**="radio" id="nan" name="gender">

<**label** **for**="nan">男</**label**>

可以为radio, checkbox添加checked属性以及为option添加selected属性

## Ife.baidu.com页面源码结构：

<!DOCTYPE html>

<html>

<head>

…

<meta name=”viewport” content=””>

<meta name=”keywords” content=””>

<meta name=”description” content=””>

</head>

<body>

<div>

<header><nav></nav></header>

<main><div></div></main>

<footer></footer>

</div>

</body>

</html>

# T**ask2**

## 关于外边距的合并的问题

外边距合并指的是，当两个垂直外边距相遇时，它们将形成一个外边距。合并后的外边距的高度等于两个发生合并的外边距的高度中的较大者。

解决办法：

1. 都用float来定位（有条件要求，适用范围较广）
2. 为父元素添加overflow不为visiable 的属性 （适用范围极广，推荐使用）
3. 为元素添加border（一般不用）
4. 使用绝对定位（适用范围较窄）
5. 父元素增加padding-top属性（改变尺寸，不建议使用）

## 对于便签选择器的灵活运用

. SideBar>h1;子选择器

.Gallery img包含选择器

.pic01,.pic02,.pic03分组选择器

：**>**作用于元素的第一代后代，**空格**作用于元素的所有后代。

注意：后代选择器，后面不能跟分组选择器，只会作用第一个。

:nth-child(*n*) 选择器匹配属于其父元素的第 N 个子元素，不论元素的类型。

Odd 和 even 是可用于匹配下标是奇数或偶数的子元素的关键词（第一个子元素的下标是 1）

使用公式 (*an* + *b*)。描述：表示周期的长度，n 是计数器（从 0 开始），b 是偏移值。

:last-child 选择器匹配属于其父元素的最后一个子元素的每个元素。

## 使用超链接a标签的伪类

a:link {color: #FF0000} /\* 未访问的链接 \*/  
a:visited {color: #00FF00} /\* 已访问的链接 \*/  
a[:hover](https://www.baidu.com/s?wd=%3Ahover&tn=44039180_cpr&fenlei=mv6quAkxTZn0IZRqIHckPjm4nH00T1YznHb3mhDzrHfYrHmYPvPB0ZwV5Hcvrjm3rH6sPfKWUMw85HfYnjn4nH6sgvPsT6KdThsqpZwYTjCEQLGCpyw9Uz4Bmy-bIi4WUvYETgN-TLwGUv3EnHm3rHbvrjDYnWRLnHcsPHnsr0) {color: #FF00FF} /\* 鼠标移动到链接上 \*/  
a:active {color: #0000FF} /\* 选定的链接 \*/

## CSS3 box-shadow

box-shadow: h-shadow v-shadow blur spread color inset;

h-shadow 必需的。水平阴影的位置。允许负值

v-shadow 必需的。垂直阴影的位置。允许负值

blur 可选。模糊距离

spread 可选。阴影的大小

color 可选。阴影的颜色。在CSS颜色值寻找颜色值的完整列表

inset 可选。从外层的阴影（开始时）改变阴影内侧阴影

## 首行文本的缩进

text-indent: 2em; 首行空2个字

text-indent 属性规定文本块中首行文本的缩进。

## Table

border-collapse 属性设置表格的边框是否被合并为一个单一的边框

separate 默认值。边框会被分开。

collapse 如果可能，边框会合并为一个单一的边框

## 表单的布局

左侧对齐：

左侧文字用<span>,定位absolute，left：40%

右侧对齐；

右侧input用<label>包裹，定位relative，left：40%

# T**ask3**

## 三栏式布局

方法一：inline-block方法

left,center,right{ display：inline-block} /\* 包裹作用，不会塌陷 \*/

left{width:200px}

right{width:120px}

center{width:calc(100%-200px-120px-40px)} /\* 动态计算中间盒子的宽度 \*/

方法二：float布局

<div class="left">

<div class="right"> /\* 这里将right提到center前面 \*/

<div class="center">

.left{float:left}

.center{ margin-left: 220px; margin-right: 140px; }

.right{float: right;}

方法三：Flex布局（新）

.middle{display: flex;}

.left,.right{width: 100px;}

.right{order: 2;}

.main{

order: 1;

flex: 1; /\*让所有弹性盒模型对象的子元素都有相同的长度，忽略它们内部的内容\*/

margin: 0 20px;

}

方法四：圣杯和双飞翼布局（旧）

基本思路及区别：  
1.两种布局的基本构思为：首先让中间层100%宽度占满同一高度的空间，在左右两层被挤出中间层所在区域时， 使用margin-left的负值将左右两个层拉回与中间层同一高度的空间，接下来调整左右两层到指定位置， 最后使用中间层的margin或padding属性使中间层的内容躲出左右两层占住的显示区。

2.两种布局的主要区别在于：圣杯布局采用一个父层包含中间、左右三个子层，设置父层的padding值腾出左右两层的显示区， 并对左右两层使用relative和left、right值调整位置；双飞翼采用中间、左右三层并列，再在中间层里设置一个子层， 设置中间层子层的margin值腾出左右两层的显示区，对左右两层使用margin值即可调整位置；

圣杯html结构：

<**div** id="container">

<**div** id="main"></**div**>

<**div** id="left"></**div**>

<**div** id="right"></**div**>

</**div**>

　　圣杯css基本代码：

**#container**{

padding: 0px 120px 0 140px;*/\*腾出宽度\*/*

}

**#main**{

width: 100%;

position: relative;

}

**#left**{

width: 140px;

margin-left: -100%;*/\*将left层拉回main层所在高度区域\*/*

left: -140px;*/\*调整位置\*/*

position: relative;

}

**#right**{

width: 120px;

margin-left: -120px;*/\*将right层拉回main层所在高度区域\*/*

right: -120px;*/\*调整位置\*/*

position: relative;

}

　　双飞翼html结构：

<**div** id="main">

<**div** id="main-inner"></**div**>

</**div**>

<**div** id="left"></**div**>

<**div** id="right"></**div**>

　　双飞翼css基本代码：

**#main**{

width: 100%;

position: relative;

}

**#main-inner**{

margin：0 120px 0 140px；

｝

**#left**{

width: 140px;

margin-left: -100%;*/\*将left层拉回main层所在高度区域\*/*

}

**#right**{

width: 120px;

margin-left: -120px;*/\*将right层拉回main层所在高度区域\*/*

}

## box-sizing 属性

取值说明

1、content-box:此值为其默认值，其让元素维持W3C的标准Box Model，也就是说元素的宽度/高度（width/height）等于元素边框宽度（border）加上元素内边距（padding）加上元素内容宽度/高度（content width/height）即：Element Width/Height = border+padding+content width/height。

2、border-box:此值让元素维持IE传统的Box Model（IE6以下版本），也就是说元素的宽度/高度等于元素内容的宽度/高度。（从上面Box Model介绍可知，我们这里的content width/height包含了元素的border,padding,内容的width/height【此处的内容宽度/高度=width/height-border-padding】）。

## inline-block之间间距

元素间留白间距出现的原因就是标签段之间的空格，因此，去掉HTML中的空格，自然间距就木有了。考虑到代码可读性，显然连成一行的写法是不可取的，我们可以：

<div class="space">

<a href="##">

惆怅</a><a href="##">

淡定</a><a href="##">

热血</a>

</div>

或者是：

<div class="space">

<a href="##">惆怅</a

><a href="##">淡定</a

><a href="##">热血</a>

</div>

或者是借助HTML注释：

<div class="space">

<a href="##">惆怅</a><!--

--><a href="##">淡定</a><!--

--><a href="##">热血</a>

</div>

# T**ask4**

## 水平及上下居中

方法一：绝对定位的居中实现

position: absolute;

top: 50%;

left: 50%;

transform: translate(-50%,-50%); /\*x轴y轴各位移div盒子的一半\*/

方法二：margin:auto实现绝对定位元素的居中

position: absolute; left: 0; top: 0; right: 0; bottom: 0;

margin: auto; /\* 有了这个就自动居中了 \*/

方法三：Flex布局方法：

width:100%;

height:100%;

display: -webkit-flex;

display: flex;

justify-content:center;

align-items:center;

## 清除浮动

方法一：overflow:hidden;（或者auto，hidden更好）

方法二：清除浮动更好的方法是**clearfix技巧**

“clearfix”技巧是基于在父元素上使用“:before”和“:after”两个伪类。使用这些伪类，我们可以在浮动元素的父容器前面和后面创建隐藏元素。“:before”伪类是用来防止子元素顶部的外边距塌陷，使用“display: table”创建一个匿名的“table-cell”元素。这也确保在IE6和IE7下具有一致性。“:after”伪类是用来防止子元素的底部的外边距塌陷，以及用来清除元素的浮动。

一个常见的方法是将定义一个类名，把这个类名加到需要清除浮动的容器上。例如使用“clearfix”清除浮动，Dan Cederholm为容器设置了一个类名“group”。在需要清除浮动的容器上添加这个类名“group”。

.group:before,

.group:after {

content: "";

display: table;

}

.group:after {

clear: both;

}

.group {

\*zoom: 1;

}

## absolute定位初始位置

在文档流中，定义为position:absolute的元素已经被删除了。

那它的定位到底是相对于body，还是父级元素呢？

结论如下：

position:absolute是相对于他的包含块中第一个有position:absolute或relative及 fixed属性的父级元素，如果都没有，就是相对于body。

## border-radius属性

border-radius属性其实可以分为四个其他的属性：

border-radius-top-left /\*左上角\*/

border-radius-top-right /\*右上角\*/

border-radius-bottom-right /\*右下角\*/

border-radius-bottom-left /\*左下角\*/

//提示：按顺时针方式

制作半圆：

width:100px;

height:50px;//高度是宽度的一半

background-color:#000;

border-radius:50px 50px 0 0;//左上和右上至少为height值

画实心圆

width:100px;

     height:100px;

    background-color:#cb18f8;

     border-radius:50px;

（或者border-radius:50%）

# T**ask5**

## 水平及上下居中