

移动 Web 应用设计

复习课问题汇集（扩展）

版本 3（2017.1.6）

期末考试题目类型：单选题、填空题、判断题、简答题

* 下面很多题目都是做过实验的内容，可以通过做题目检验一下对课程内容的掌握程度

HTML

- 1、常用的块级元素有哪些？常用的行内元素（内联元素）有哪些？块级元素和行内元素的区别是什么？

常用的块级元素：`h1`、`div`、`p`、`ul`、`li`、`table`、`form`、`fieldset`、`menu`

常用的行内元素：`span`、`a`、`em`、`img`、`input`、`br`、`i`、`label`、`select`

区别：

- 1) 块级元素默认不与其它元素在同一行显示，即会自动垂直摆放；多个行内元素可以在同一行显示。
- 2) 块级元素可以定义高度和宽度；行内元素的高度和宽度由其内容确定。
- 3) 块级元素中可以包含行内元素和其它块级元素。行内元素只能行内元素，不能包括块级元素。

- 2、HTML4 布局元素常用的有哪些？HTML5 布局元素常用的有哪些？

HTML4 布局元素常用的有哪些？：`h1~h6`、`p`、`div`

HTML5 布局元素常用的有哪些？：`h1~h6`、`p`、`div`、`header`、`nav`、`section`、`article`、`aside`、`footer`

- 3、元素 `a` 有哪些主要属性？它们的作用是什么？

1) `href` 属性：用于指定超链接目标的 URL

2) `name` 属性：用于指定锚（`anchor`）的名称；可以创建（大型）文档内的书签

- 4、元素 `img` 的 URL 用什么属性定义？

`src` 属性

- 5、请解释 tomcat 虚拟主机和虚拟目录的含义。

虚拟主机：就是把一台运行在互联网上的物理服务器划分成多个“虚拟”服务器。本质上是将不同的域名绑定到同一个 IP 地址上面。

虚拟目录：用于将服务器上面的一个 web 应用目录、映射为可供外部用户访问的 url 路径。

6、URL 的地址主要有哪几部分组成？

模式(scheme)+主机名(host)+路径(path)+文件名(filename)+查询(query)

主机名部分可以是 IP 地址、域名或计算机名；

下面的路径/2000/cn 也称为虚拟路径。

URL的基本格式:

模式(scheme) 主机名(host) 路径(path) 文件名(filename) 查询(query) 参数

http: // www.group.com /2000/cn/main.html ? type=5&gp=isl

7、说明 URL 的绝对路径和相对路径的区别。

绝对路径从网站根目录开始定位

相对路径从当前网页所在目录开始定位

如果当前网页的 URL 为 <http://tools.ietf.org/html/rfc6694.html>，下面两个链接分别使用了绝对路径和相对路径：

[URI Scheme Syntax](/html/div/spec.html)

[URI Scheme Syntax](div/spec.html)

8、Web 服务器如何利用 HTTP 请求中的 URL 找到要执行或要下载的文件？

浏览器显示一个静态网页（例如，一个 html 文件）的过程：

（1）浏览器确定 Web 服务器所在主机的 IP 地址和 Web 服务器使用的端口号(默认为 80)，然后把该文件的 URL 发送 Web 服务器

（2）Web 服务器通过 URL 确定虚拟主机，然后确定虚拟目录，再确定虚拟目录下的子目录，最后找到该文件(.html)并把它发送给浏览器。

（3）浏览器收到该文件后显示网页。如果 html 文件中包含其他关联文件（img 等）的链接，则用获得 html 文件的类似步骤获得它们并显示出来。

9、列表元素常用的有哪两个？

ul 无序列表

li 子元素(ol、ul、div、menu 都采用 li) // 它们属于块级元素

10、表格元素常用的有哪三个？

<table>

<td>: table data cell 单元格

<tr>: table row

11、表单元素的属性主要有几个？它们常用的取值是什么？

四个: action method enctype target

action 属性用 URL 地址指出后台程序名。默认为当前程序。

method 属性指出使用哪种提交方式，提交方式主要有两种 get(默认)和 post。用 get 方式提交主要用于从 Web 服务器上取回网页，并且可以在 URL 参数加入少量输入数据(长度有限制)，这和用链接获取网页的工作过程完全一样。用 get 提交的数据会记录在历史数据中，不具有保密性。如果要提交大量数据，最好采用 post 方法。

采用 form 除了可以用于上传表单的输入数据之外，还可以用于上传文件给服务器。那服务器到底如何区分它们呢？答案是采用 form 的属性 enctype。元素 form 的属性 enctype 主要有三个取值：

（1）application/x-www-form-urlencoded：用于上传表单数据，这是默认方式

（2）multipart/form-data：用于上传文件，还可以同时上传表单数据。

（3）text/plain：上传纯文本。

如果对于提交表单的响应需要在新窗口显示，可以利用 form 的 target 属性：_blank 每次都在新窗口打开，_self 在当前窗口或框架打开(默认)，_top 在浏览器窗口打开，_parent 在父框架打开，frameName 在指定窗口打开(没有就新建该窗口)

12、说明 iframe 的使用方法。

iframe 可以放在网页的任何地方，而不用放在 frameset 中。

iframe 所取回的网页会在 iframe 元素位置显示出来。

iframe 可以用于实现 ajax 技术和用作编辑器。

```
<iframe width="1000" height="600"
```

```
frameborder="0" scrolling="yes" (不显示边框，显示滚动条)
```

```
src="http://www.sysu.edu.cn">
```

```
Your browser does not handle frames!
```

```
</iframe>
```

// frameset 不可以放在 body 里，iframe 可以

13、回车和空格元素是什么？

注释：<!--这是 html 的注释 --> 内容可以占据多行。

回车：

空格：

小于号：< 或者小于号"<"后加一个空格

水平线：<hr>

CSS

1、基本选择器有哪三个？

元素选择器 p {font-size: 14px;} 选择所有 p 元素 (或称为类型选择器)

类选择器 .even {font-style: italic;} 选择所有类为 even 的元素

id 选择器 #start {font-weight: bold;} 选择 id 为 start 的元素

// id 是唯一的，class 是类名，可以不唯一

2、复合选择器常用的有哪些？它们的含义是什么？

后代选择器：p span {font-size: 14px;} 选择 p 元素的所有子孙元素中的 span 元素

子女选择器：p>span {font-size: 14px;} 选择 p 元素的所有子女元素中的 span 元素

组合选择器：p#start {font-size: 14px;} 选择 id 为 start 的 p 元素(交集)

群组选择器：em, .even {font-size: 14px;} 选择 em 元素或者类名为 even 的元素

相邻兄弟选择器：h1 + p {color:red;} 选择 h1 之后的相邻兄弟元素(必须为 p 元素)

后续兄弟选择器：h1 ~ p {color:red;} 选择在 h1 之后所有兄弟元素中的 p 元素

否定选择器：p:not(#start){ color:red;} 选择所有 id 不是 start 的 p 元素

3、属性选择器[alt]和[alt='fig2-1']分别表示什么意思？

[alt] 选择所有具有属性 alt 的元素

[alt="fig 2-1"] { width:20px; } 选择所有属性 alt 的值等于"figure 2-1"的元素。

[alt] { width:200px; } 选择所有具有属性 alt 的元素

img[alt] { width:200px; } 选择所有具有属性 alt 的 img 元素

[alt="fig 2-1"] { width:20px; } 选择所有属性 alt 的值等于"figure 2-1"的元素。

img[alt^="fig"] { width:20px; } 选择 alt 值以“figure”开头的所有 img 元素。

img[alt\$="2-1"] { width:20px; } 选择 alt 值以“2-1”结尾的所有 img 元素。

img[alt*="5-"] { width:20px; } 选择 alt 值包含字符串“5-”的所有 img 元素。

`img[alt~="fig"] { width:20px; }` 选择 alt 值包含单词 “fig” 的所有 img 元素。

- 4、如何选出所有没有属性 alt 的 img 元素？如何选择网页中的所有元素？

否定选择器 `img:not([alt])` 没有属性 alt 的 p 元素

`p:not(#start){ color:red;}` 选择所有 id 不是 start 的 p 元素

通配符* 选择所有元素

- 5、伪元素选择器常用的有哪些？它们的含义是什么？

`p:first-letter {font-size:2em;}` p 元素内容的第一个字母

`p:first-line {color:red;}` p 元素内容的第一行

`p:after {content:"...";}` 在 p 元素的内容之后插入内容...

`p:before {content:"***";}` 在 p 元素的内容之前插入内容***

`::selection` 为选中的文本设置样式

- 6、伪类选择器常用的有哪些？它们的含义是什么？

`a:link {color:blue;}` 没有访问过的链接

`a:visited {color:red;}` 访问过的链接

`a:hover {font-size:2em;}` 鼠标悬停下的元素(其他元素也可以)

`a:active {color:green;}` 正在访问(按下鼠标)的链接

`input:focus {background-color:red;}` 获得焦点的元素(一般为输入元素)

遵循 LoVe/Hate 原则进行设置

- 7、`p:nth-child(3n+2)` 的含义是什么？`p:nth-of-type(3n+1):nth-of-type(-2n+9)` 会选到哪些元素？

`p:nth-child(3n+2)`: 选择 p 的双亲的第 2,5,8,11..... 个子女，相应位置为元素 p 才会被选中。

`p:nth-of-type(3n+1):nth-of-type(-2n+9)`: 选择 p 元素的双亲的第 1,7 位子女，只有 p 元素参与计数。

- 8、样式可以放在哪些地方定义？

(1) 内联样式：直接放在元素的 style 属性中。

例如：`<p style="font-size:2em"> This is paragraph. </p>`

(2) 内部样式表：用 style 元素存放。style 元素可以放在 html 文档的任何地方，也可以多次出现。一般放在元素 head 中。

例如：`<style type="text/css">`

`p {font-size:2em}`

`</style>`

(3) 外部样式表：存放在一个或多个外部文件(.css)中。

引入样式的方法：

①在 head 元素中定义 link 元素把样式引入网页， 例如`<link type="text/css" href="/css/main.css">`

②css 文件还可以在 style 元素中或者 css 文件中用 @import 语句 加入

`@import url(/css/main.css);`

文件 main.css 包含：`p {font-size:2em}`

- 9、如何确定一个选择器的特殊性？请说明如何确定一个元素所使用的样式。

确定特殊性：

(1) 对于 ID 选择器，每个特殊性加 0, 1, 0, 0。

- (2) 对于类选择器、属性选择器、伪类选择器，每个特殊性加 0, 0, 1, 0。
- (3) 对于元素选择器和伪元素选择器，每个特殊性加 0, 0, 0, 1。
- (4) 每个元素只能定义一个行内样式，行内样式的特殊性为 1, 0, 0, 0。
- (5) 结合符和通配选择器，对特殊性没有任何贡献，即其特殊性为 0, 0, 0, 0。
- (6) 继承得来的选择器没有任何特殊性，即其特殊性为 0, 0, 0, 0。
- (7) 加上 !important 的样式具有最高优先权。例如，p {color:blue!important; }

特殊性排序：

Important > 行内样式 > ID 选择器 > 类选择器、属性选择器、伪类选择器 > 元素选择器、伪元素选择器 > 结合符、通配选择器、继承的选择器

确定一个元素所使用的样式：

选择器越特殊优先权越高，选择器的优先权最高的那个样式被选中。如果最高优先权的样式有多个，则要看这些样式中哪个样式离当前元素最近，离得最近的被选中。

10、结合符和通配符对特殊性没有贡献，这句话是否正确？

正确。

11、说明下面两个选择器的特殊性：

(1) tr:nth-child(2) td:nth-child(n+1):nth-child(-n+3) {color:grey;}

0,0,1,1 + 0,0,2,1 =0,0,3,2

(2) tr:nth-child(2) td:nth-child(n+4):nth-child(-n+7):after,tr:nth-child(3) {color:grey;}

0,0,1,1 + 0,0,2,2 =0,0,3,3 (after 是伪元素)

0,0,1,1

* 拆分成两个，得到两个特殊性

12、拆分下面的选择器为最小属性可以得到多少个选择器？它们的特殊性有哪些取值？

p#msg, ul:hover li:nth-child(3) {margin: 20px;border: dashed 1px gray}

4 个

p#msg {margin: 20px;} /*0,1,0,1*/

p#msg {border: dashed 1px gray} /*0,1,0,1*/

ul:hover li:nth-child(3) {margin: 20px;} /*0,0,2,2*/

ul:hover li:nth-child(3) {border: dashed 1px gray} /*0,0,2,2*/

参考：



13、你是否可以讲出下面文本格式化所用的样式属性的作用和取值方法吗？

文本格式化

- 字体: font-family, @font-face
- 文本颜色和字号: color, font-size
- 文本修饰: font-style, font-weight, text-transform
font-variant, text-decoration, text-shadow
- 格式化段落: line-height, text-indent, text-align
white-space, word-wrap, word-break

出题方法的例子:

- (1) color 取值有哪些常用的表示方法 (三种, 用例表示)?

```
p { color: red; }  
p { color: #3E8988; }  
p { color: rgb(0,0,255); }  
p { color: rgba(0,0,255,0.5); }  
p { color: hsla(300,100%,50%,0.3); }
```

- (2) font-size 有哪些常用单位? (写三个)

px、em、rem

- (3) 下划线使用哪个属性定义的?

text-decoration

Keywords: underline(下划线) overline(上划线) line-through(删除) none

- (4) font-style 用来定义什么?

文本字体样式

font-style: italic; 斜体字

font-style: normal; 正常字体

- (5) 文本的水平对中采用什么样式属性?

text-align:center;

- (6) white-space 有哪些取值? 作用是什么?

white-space:pre; : 保存源码中的空白和回车

white-space:pre-wrap; : 保持源码中的空白和回车, 自动换行

white-space:nowrap; 不保留源码中的空白和回车也不自动换行

14、说明盒模型的下列属性的作用和取值, 其中标注了下划线的是一些重要属性。

盒模型(box model)

- 概述: margin(-TopRightBottomLeft), padding(-TRBL), border(-TRBL+style|color|width), box-sizing
width, height, display, visibility
- 设置背景色或背景图: background-color, background-image, background-repeat, background-clip
background-origin, background-attachment, background-size, background-position
linear-gradient, repeating-linear-gradient, radial-gradient, repeating-radial-gradient
- 修饰边框: border-radius, box-shadow, border-image
- 设置尺寸与溢出: height, width, box-sizing, max-width, min-width, max-height, overflow, overflow-x, overflow-y
- 设置浮动元素: float, clear
- 使用定位技术: position, z-index
- 垂直对齐: vertical-align
- 列表样式: list-style-type, list-style-position, list-style-image
- 表格样式: border-collapse, border-spacing, empty-cells, table-layout
- 变换、过渡和动画: transform(skew,scale,rotate,translate,rotate3d), transform-origin
backface-visibility, perspective, perspective-origin
transition(property,duration,delay,timing-function)
animation(name,duration,timing-function,direction,fill-mode,play-state)

出题方法的例子:

(1) 说明 margin 和 padding 有何区别?

margin: 是一个元素与另一个元素之间的空白(外边距)

padding: 是内容和其边框线之间的空间 (内边距)

(2) 说明含义: (a) margin: 10px auto (b) margin: 1em 2em 3em 4em

(a) 元素之间上下留空白 10px, 左右自动居中

(b) 元素与上边元素的之间留空白为 1em, 元素与右边元素的之间留空白 2em, 元素与下边元素的之间留空白 3em, 元素与左边元素的之间留空白 4em

(3) display:none 和 visibility:hidden 都是让元素看不见, 有何差别?

display:none: 元素不显示, 完全不占用空间

visibility:hidden: 元素不显示, 在文本流依然占据原来的空间

(4) overflow、float 和 position 的主要取值有哪些? 什么含义? clear 有什么用?

Overflow: scroll: 显示滚动条

Overflow: hidden: 不显示溢出部分

Overflow: auto: 溢出时才显示滚动条

Overflow: visible: 全部显示出来

Float: left: 元素浮动到左边

Float: right: 元素浮动到右边

Float: none: 默认值。元素不浮动, 并会显示在其在文本中出现的位置。

Position: absolute: 绝对布局: 相对于非 static 的最靠近的祖先元素进行定位

Position: relative: 相对布局: 相对于自己在正常文档流的位置进行定位

Position: fixed: 相对于浏览器窗口进行定位

Position: static: 设置为正常文档流 (默认)

clear:规定元素的指定一侧不允许出现浮动元素。

(5) 说明值的含义: `box-shadow: -5px 6px 5px #DDD;`

添加阴影: 水平偏移量为-5px, 垂直偏移量为 6px, 模糊半径(模糊度)为 5px, 阴影颜色为#DDD

(6) 说明值的含义: `div {border-radius: 2em/1em;}`

设置元素边框的圆角半径, x 方向半径为 2em, y 方向半径为 1em

(7) 显示单表格线的样式属性是什么?

`border-collapse: collapse`

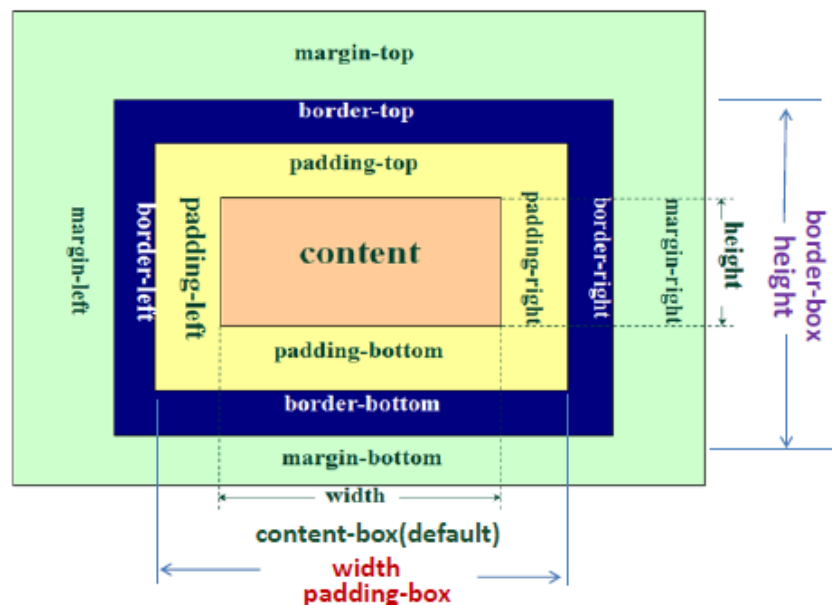
(8) 让无序列表不显示点的样式属性是什么?

`ol { list-style-type: none; }`

(9) 默认的 width 是指什么宽度? box-sizing 如何改变这个定义?

height 和 width 默认只是内容的高度和宽度, 并非盒子的整个高度和宽度。

盒模型包含内容 (content)、内边距 (padding)、边框 (border)、外边距 (margin)



- padding是内容和其边框线之间的空间(内边距)。border是边框。
- background-color 用颜色填充边框内部的空间, 包括padding部分和虚线边框。
- margin是一个元素与另一个元素之间的空白(外边距)。
- 改变box的height和width的计算方法: `box-sizing: border-box; // padding-box content-box`
(Firefox: `-moz-box-sizing` Safari : `-webkit-box-sizing`)

box-sizing {
border-box: width包含内容宽度、左右padding和左右border-width
content-box: width只包含内容宽度 (默认)

(10) z-index 在 position 取什么值的时候起作用?

`absolute relative`

属性值为 absolute 对象的 z-index 属性可以设置层叠显示的次序, 它是直接有

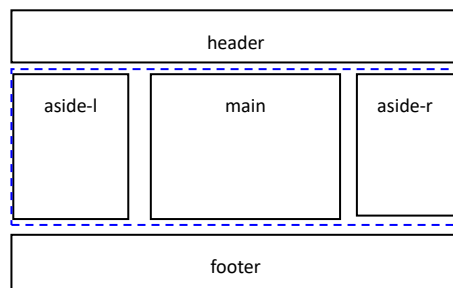
效的;而属性值为 relative 对象的 z-index 属性在设置时要小心,把当前对象的 z-index 设置为-1 是不行的,在 firefox 中它会无法显示,必须设置为 0 以上,我们如果想让别的对象挡住它,只有将其他对象也设置 position 为 relative,并将 z-index 属性取一个比它大的值即可. 而 IE 不自动识别 relative 的深度的问题,就是默认值就是他们的出现顺序。

http://blog.sina.com.cn/s/blog_59a3e0a90100c38l.html

(11) background-repeat 有哪些取值?

Repeat, repeat-x, repeat-y, no-repeat

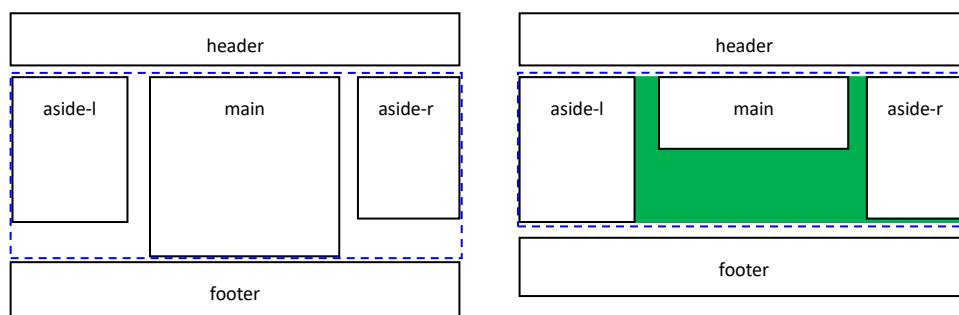
- 15、设计一个如下用 6 个 div 实现的三列布局。如果 header、footer、container(虚线框)都定义了相同的百分比宽度且用 margin 对中, aside-l 和 aside-r 采用固定宽度, main 随 container 宽度变化而变化,如何使用 CSS 来布局这些元素?



```
<head>
<meta http-equiv="Content-Type" content="text/html; charset=utf-8">
<title>复习测试</title>
<style>
#header, #container, #footer{
    width: 50%;
    margin: 0 auto;
    margin-top: 10px;
}
#aside-l {
    float:left;
    width: 200px;
}
#aside-r {
    float:right;
    width: 200px;
}
#main {
    /* width: 60%; */
    width: calc(95% - 400px); /* 16、使得main的内容保持在中间 */
    margin: 0 auto;
}
#header, #footer, #aside-l, #aside-r, #main{
    border: solid 1px black;
}
#container {
    border: dashed 1px black;
    background-color: #DDD; /* 16 */
}

```

- 16、在上题中,如果要让 main 在高度超出 aside 时内容保持在中间而不会铺到 aside 下面需要用什么样式属性? 如果要让 main 在高度低于 aside 时 container 的背景颜色填满整个中间区域(与 aside 平齐)需要加上什么样式属性?



```

/* 16、文本框内仍然为原来的颜色，即白色 */
#aside-l, #aside-r, #main{
    background-color: white;
}
</style>
</script>
</script>
</head>
<body>
    <div id="header"> header </div>
    <div id="container">
        <div id="aside-l">aside-l aside-l aside-l
        aside-l aside-l aside-l
        aside-l aside-l aside-l
        aside-l aside-l aside-l
        aside-l aside-l aside-l</div>
        <div id="aside-r">aside-r aside-r aside-r
        aside-r aside-r aside-r
        aside-r aside-r aside-r
        aside-r aside-r aside-r
        aside-r aside-r aside-r</div>
        <div id="main">mainmainmain</div>
        <div style="clear:both"></div> <!-- 空的div用于使container的高度与浮动元素相同 -->
        </div>
        <div id="footer"> footer </div>
    </body>

```

- 17、什么是媒体查询？如果 15 题要求还可以用于手机浏览器，布局元素从上到下排列方式（无浮动）为：header、main、aside-l、aside-r、footer，15 题的设计需要做什么修改？

创建弹性布局

- （1）桌面电脑和平板电脑采用浮动的多列，手机不浮动，即利用块级元素的特性上下布局。
- （2）桌面电脑要设置外部容器的max-width（例如，960px）。
- （3）采用百分数设置width。例如，并排的两个元素(main,sidebar)的width可以设置为61%和39%。为了使宽度包含padding和border，可以设置box-sizing:border-box。如果要加上margin，则要通过calc(61%-20px)计算加上margin（10px）的宽度。
- （4）缩紧空白区：手机和平板使用较小的margin和padding。
- （5）调整字号：手机上的字号应设置大一点，标题字号不要过大。为了容易更改字号大小，最好采用em和rem定义字号，还可以使用关键字(xx-small, x-small, ..., xx-large)定义字号。
- （6）修改导航菜单：手机上导航菜单不要采用多行，最好下拉。
- （7）隐藏内容：在手机上隐藏不重要的内容，点击后展开。
- （8）使用背景图像。

媒体查询

利用媒体查询可以根据不同设备的特点来分别设计样式。例如：下面样式可以分别为手机、平板电脑、桌面电脑定义样式。

```
<head>
...
<link href="small.css" rel="stylesheet" media="only screen and (max-width:480px)" />
<link href="medium.css" rel="stylesheet" media="only screen and (min-width:481px)
and(max-width:768px)" />
<link href="large.css" rel="stylesheet" media="only screen and (min-width:769px)" />
...
</head>
```

也可以直接在head元素或一个样式文件中直接定义样式：

```
@media only screen and (min-width:481px) and (max-width:767px) {
  p {
    color:red;
    font-weight:bold;
  }
}
```

一般先定义移动设备的样式，然后利用@media指令对在平板电脑和桌面电脑中不同的样式进行修改。

媒体查询的一般语法：

@media logic type and (feature:value) and (feature:value)... {...}

- ① type取值screen、print等。
- ② logic取值only或not。not表示出了这个类型之外所有类型。
- ③ feature的表示如下：
 - min-width、max-width
 - width、height
 - device-width、device-height
 - orientation（方向）
 - aspect-ratio（高宽比）
 - color（彩色）、monochrome（单色）
 - color-index（颜色数）
 - resolution（分辨率）
 - scan（扫描）
 - grid（栅格）
 - webkit-device-pixel-ratio
 - moz-device-pixel-ratio

```

<head>
<meta http-equiv="Content-Type" content="text/html; charset=utf-8">
<title>复习测试</title>
<style>
#header, #container, #footer{
    width: 50%;
    margin: 0 auto;
    margin-top: 10px;
}
#header, #footer, #aside-l, #aside-r, #main{
    border: solid 1px black;
}
#left {
    border: solid 1px red;
}
#container {
    border: dashed 1px black;
    background-color: #DDD; /* 16 */
}
/* 16、文本框内仍然为原来的颜色，即白色 */
#aside-l, #aside-r, #main{
    background-color: white;
}
@media only screen and (min-width:769px) {
    #left {
        width: calc(100% - 220px);
        float:left;
    }
    #aside-l {
        width: 200px;
    }
    #aside-l {
        width: 200px;
    }
    #main{
        width: calc(100% - 220px); /* 16、使得main的内容保持在中间 */
        float: right;
    }
    #aside-r {
        width: 200px;
        float: right;
    }
}
</style>

<script>
</script>
</head>

<body>
<div id="header"> header </div>
<div id="container">
    <div id="left">
        <div id="main">main main main</div>
        <div id="aside-l">left left left</div>
    </div>
    <div id="aside-r">right right right</div>
    <!-- 空的div用于使container的高度与浮动元素相同 -->
    <div style="clear:both"></div>
</div>
<div id="footer"> footer </div>
</body>

```

Java

1、什么是 Java 的包？为什么要引入包？

/*什么是 Java 包*/

包（package）是 Java 提供的一种区别类的名字空间的机制，是类的组织方式，是一组相关类和接口的集合，它提供了访问权限和命名的管理机制。Java 通过子目录来有效地管理它们。同一个子目录下的所有.class 文件被认为在同一个包(package)中。可以把一个或者多个这样的子目录打包成一个 Java 的类库（.jar 文件）。

*/*为什么要引入 Java 包*/ //PPT+百度一答案*

Java 中的一切都是对象(类)。类放在 java 文件中，每个 java 文件可以包含多个类，但只能包含一个 **public** 类，编译该文件后每个类会产生一个.class 文件(字节码文件)。一个大型程序由很多这样的.class 文件组成。

为了更好地组织类，Java 提供了包机制。通俗来讲，就是把 Java 类分门别类的整理起来，需要的时候在引入，更具条理性和直观性。功能上，包的引入主要是要符合 Java 面向对象的编程思想。

导入包是为了能够引用里面的类和方法。包的作用就像书的目录，当你要用到某个方法的时候，然后根据目录找到你要用的方法，按不同的功能把它分为好几个种类、索引，每个种类下面有很多方法(当然还有别的)，就要去找目录。包是对类的封装，导入包后就可以使用其中的类，也可以调用类的方法。

此外，目前很多现有的类基本上都是以 **package** 包的形式存在的，类似于一个 Java 的基础库，包的引入可以使用其中的类，也相当于使用了现成的方法。

*/*如何定义 Java 包*/*

除了注释语句，Java 的所有文件可以在第一行用 **package** 语句指明所属包的名称，即所在子目录的名称。在子目录 **com\group\food** 下的文件的包语句为 "**package com.group.food**", 其中，**com.group.food** 就是包的名称，这里用网站(group.com)及子目录(food)唯一确定一个包。没有定义包的文件属于默认包。

*/*如何引用 Java 包*/*

- 如果要引入包 **com.group.food** 中的所有类，就要把 "**import com.group.food.***" 加到 Java 文件的前面。也可以指定引入一个类，用语句 "**import com.group.food.Cookie**" 引入类 **Cookie**。

- 如果引用的多个包包含相同名字的类，则需要加入包名使用类：**java.util.Vector v = new java.util.Vector()**。

2、解释一下 Java 类方法的访问权限（按照相同包和不同包、子类和非子类进行分析）。

- 类的访问权限

没有修饰词 只能被同一个包的类所访问

public 能被任何类在任何地点所访问

*类没有 **private** 和 **protected** 权限。

- 成员变量和方法的访问权限

public 能被任何类在任何地点所访问(通过类的方法)

protected 只能被导出类(导出类也称子类)和同一个包的类所访问

没有修饰词 只能被同一个包的类所访问(类似 **friendly**)

private 只能被同一个类的方法所访问。

- 对一个类的数据域和方法的访问：

- (1) 被同一个类的方法所访问；

- (2) 被子类的方法所访问；

- (3) 被该类对象的方法所访问。

- 访问权限的另一种描述：

- (1) 一个方法可以访问同一个类的所有数据域和方法。

- (2) 一个方法访问同一个包的所有非 **private** 的数据域和方法。

- (3) 一个方法只能访问非同一个包的 **public** 和 **protected** 的数据域和方法。其中的 **protected** 的数据域和方法只能被子类所访问。

3、为什么 Java 有了 String 还要引入 StringBuilder？

`String` 是不可变的对象，每次对 `String` 类型的对象内容进行改变的时候其实都等同于生成了一个新的 `String` 对象，然后指向新的 `String` 对象，原来的字符串将不再使用，并由垃圾回收器自动清理它们。所以经常改变内容的字符串最好不要用 `String`，而是使用 `StringBuffer` 和 `StringBuilder`。

如果需要对字符串频繁操作，可以采用 `StringBuilder` 和 `StringBuffer` 提高效率。`StringBuffer` 和 `StringBuilder(java.lang.*)` 都是通过缓冲区操作字符串的对象。它们都不需要重新生成字符串。

/*主要方法*/

主要操作有 `append` 和 `insert` 方法：

```
StringBuffers20=new StringBuffer("uv");
```

```
s20.append("xyz"); // 并入末尾。uvxyz
```

```
s20.insert(3,"w"); //插入到中间。uvwxyz
```

```
String s21=s20.toString();//取出 s20 的内容
```

`StringBulider` 用于单线程环境。`StringBuffer` 对方法有同步机制，可以用于多线程环境。

对于单线程编程，`StringBulider` 比 `StringBuffer` 更有效率。

4、说明 `ArrayList` 和 `HashMap` 的使用方法和特点。

`ArrayList` 是一个用顺序存储结构实现线性表的类，随机访问速度快，插入删除操作比较慢。

/*而 `LinkedList` 用链表实现，插入删除快，随机访问慢。*/

`ArrayList`和`LinkedList`的主要方法：

```
boolean add(E e)           // 将指定的元素添加到列表尾部。E为泛型
void add(int index, E e)    // 将指定的元素插入列表指定位置。
void clear()                // 移除列表的所有元素。
boolean contains(Object o)  // 如果列表包含指定的元素，则返回 true。
E get(int index)            // 返回列表指定位置上的元素。
int indexOf(Object o)       // 返回列表首次出现的指定元素的索引或 -1。
boolean isEmpty()           // 如果列表为空，则返回 true
int lastIndexOf(Object o)   // 返回列表最后一次出现指定元素的索引或 -1。
E remove(int index)         // 移除此列表中指定位置上的元素。
boolean remove(Object o)    // 移除列表中首次出现的指定元素（如果存在）。

E set(int index, E e)       // 用指定的元素替代此列表中指定位置上的元素。
int size()                  // 返回此列表中的元素数。
void trimToSize()           // 将此 ArrayList 实例的容量调整为列表的当前大小
protected void removeRange(int fromIndex, int toIndex)
                           // 移除 fromIndex和 toIndex(不包括)之间的所有元素。
```

`HashMap` 和 `TreeMap` 是分别用哈希和树结构的方法实现键值映射的类，而 `TreeMap` 采用红黑树结构实现，它们的随机访问和插入删除操作速度都很快，`HashMap` 要更快一些，但是内存占用量更大，主要差别是 `TreeMap` 的键值可以有序遍历，而 `HashMap` 则不行。

HashMap的主要方法:

```
void clear()                // 从此映射中移除所有映射关系。
Object clone()              // 返回此 HashMap 实例的副本(不复制键和值)
boolean containsKey(Object key) // 是否包含指定键
boolean containsValue(Object value) // 是否包含指定值
V get(Object key)           // 返回指定键所映射的值或null。
boolean isEmpty()           // 映射集是否为空。
Set<K> keySet()              // 返回此映射中所包含的键的 Set 视图。
V put(K key, V value)       // 在此映射中关联指定值与指定键。
V remove(Object key)        // 从此映射中移除指定键的映射关系(如果存在)
int size()                  // 返回此映射中的键-值映射关系数。
Collection<V> values()      // 返回此映射所包含的值的 Collection 视图。
```

* 泛型(generic type)E可以使用用户自定义类或标准类(Integer, String等)

JSP

1、什么是静态网页？什么是动态网页？

在网站设计中，纯粹 HTML（标准通用标记语言下的一个应用）格式的网页通常被称为“静态网页”，静态网页是标准的 HTML 文件，它的文件扩展名是.htm、.html，可以包含文本、图像、声音、FLASH 动画、客户端脚本和 ActiveX 控件及 JAVA 小程序等。静态网页是网站建设的基础，早期的网站一般都是由静态网页制作的。静态网页是相对于动态网页而言，是指没有后台数据库、不含程序和不可交互的网页。

所谓的动态网页，是指与静态网页相对的一种网页编程技术。静态网页，随着 html 代码的生成，页面的内容和显示效果就基本上不会发生变化了——除非你修改页面代码。而动态网页则不然，页面代码虽然没有变，但是显示的内容却是可以随着时间、环境或者数据库操作的结果而发生改变的。

* 静态与动态的区别

1、网页制作使用的制作语言：

静态网页使用语言：超文本标记语言（标准通用标记语言的一个应用）

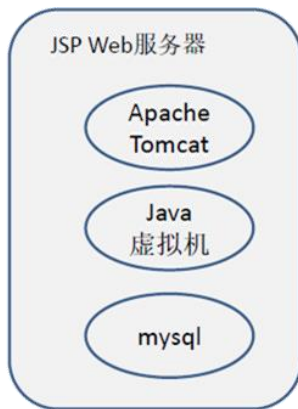
动态网页使用语言：超文本标记语言+ASP 或超文本标记语言+PHP 或超文本标记语言+JSP 等。

2、程序是否在服务器端运行，是重要标志。

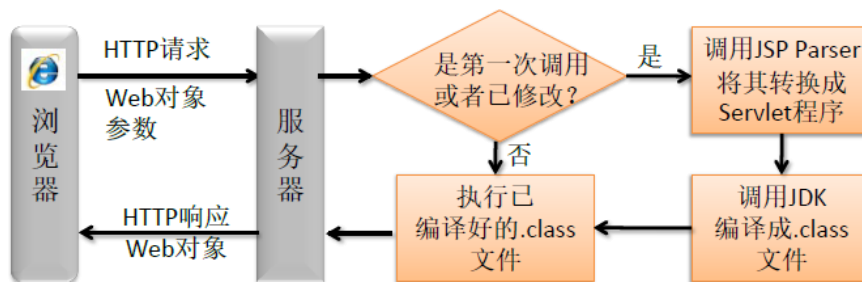
在服务器端运行的程序、网页、组件，属于动态网页，它们会随不同客户、不同时间，返回不同的网页，ASP、PHP、JSP、ASPnet、CGI 等。运行于客户端的程序、网页、插件、组件，属于静态网页，例如 html 页、Flash、JavaScript、VBScript 等等，它们是永远不变的。

2、简述 tomcat 服务器的工作原理。

- JSP 服务器：Web 服务器是接收 HTTP 请求并返回 HTTP 响应的软件。下面是一种 JSP web 服务器结构，Apache Tomcat 为 Web 服务器，它是基于 Java 虚拟机的。这个结构采用了 MySQL 数据库。如果编程使用 Eclipse，注意其内部与外部的 Tomcat 可能会冲突（只能启动一个）。



- 工作原理



(1) 当 Web 服务器收到对 JSP 文件(JSP 页面)的 HTTP 请求，如果所请求的 JSP 文件是修改后的第一次访问，则调用 JSP Parser 将其转换成 Servlet 程序 (.java)，然后调用 JDK 编译成 Servlet 字节码 (.class)。

(2) Java 虚拟机执行 Servlet 字节码发回 HTTP 响应。

(3) 如果有多个 HTTP 请求发给同一个 JSP 文件，则 Tomcat 服务器会为每一个请求启动一个线程执行该文件对应的 Servlet 字节码。

3、简述 tomcat 服务器如何把一个 JSP 程序转化为 Java 程序的。* 只说明<% %>内外的转换。

经服务器处理转换为 servlet 的 .java 程序，<% %>内部的内容会转换为 `out.write("xx\r\n")`，<% %>外部的会被转换为 `out.print("xx")`，`out.print()`只能输出字符，而 `out.write` 还可以输出字节流。e.g.如下：

```

<h1> Test </h1>
<p>                                out.write("<h1>Test</h1>\r\n");
<%                                out.write("<p>\r\n");
    out.print("Hello world!"); out.print("Hello world!");
%>                                out.write("</p>\r\n");
</p>
  
```

4、JSP 程序常用的内置对象有哪些？

JSP 包含可直接使用的 9 种基本内置对象：

`request` 用来获取客户端请求的信息

`response` 网页传回用户端的回应

`pageContext` 网页的属性是在这里管理

`session` 与请求有关的会话期

`applicationServlet` 正在执行的内容

`out` 用来传送响应的输出

`configServlet` 的构架部件

`page` JSP 网页本身

exception 针对错误网页，未捕捉的例外

/*内置对象（又叫隐含对象，有 9 个内置对象）：不需要预先声明就可以在脚本代码和表达式中随意使用*/ || 参考 1: <http://blog.csdn.net/yenange/article/details/5949518>

|| 参考 2: <http://www.233.com/Java/jichu/20100329/091644792.html>

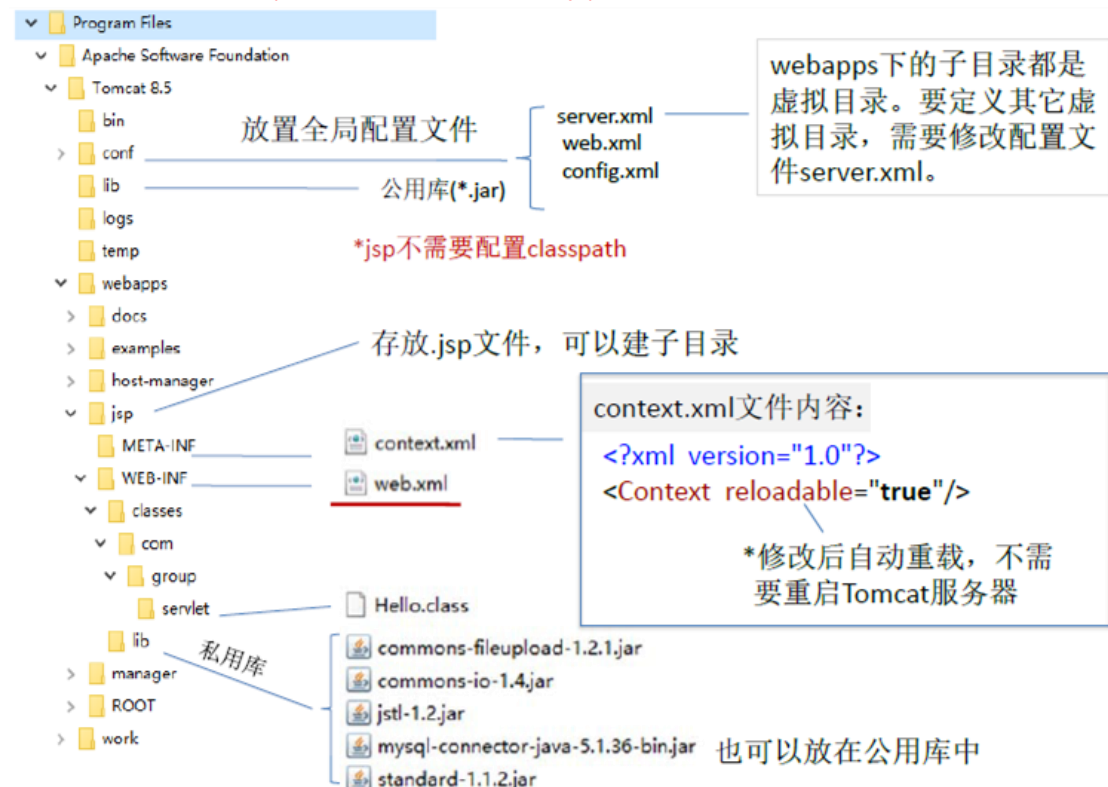
5、函数和类在 JSP 程序中哪里定义？

JSP 程序的函数和类只能在声明`<%!... %>`中定义，它们会被分别转换为 **servlet** 类的方法和内部类。声明中定义的变量会被所有用户线程所共享。

// 在声明中不能使用 **jsp** 内置对象，例如：**out**、**session**、**request**、**response** 等。这些对象可以作为参数传入到声明中。

6、tomcat 服务器要求用户 jar 包和.class 文件放在虚拟目录的什么子目录下？JSP 程序如何使用包中的类？

目录结构如下：以 <http://172.18.187.230:8080/jsp/servlet/Hello> 为例



使用包中的类需要把 **jar** 放入相应目录位置后导入项目中，接着在 JSP 程序中 **import** 这个包，就可以使用包中的类和方法了；添加方法以及规则如下：

- JSP 的类一定要指定包名，不能用默认包

- 一个网页可以出现多个 **import**：

```
<%@page import="a.b.*"%>    <%@page import="x.y.*"%>
```

- 可以合并，用逗号隔开: `<%@page import="a.b.*,x.y.*"%>`

e.g. `<%@page import="com.group.jsp.Sum" %>` 或: `import="java.util.*"`

7、如何编程可以在提交给自己后保持原有输入？用 **input-text** 或 **input-radio** 举例说明。

用 **string** 来存从提交给自己的数据里面提取的参数，**input-text** 比较简单直接返回取回的参数，**radio** 的想法就是有几个 **radio** 就用一个相应大小的字符串数组，然后哪一个选中了，对应的字符串赋值为 **“checked”**

```

<%@ page language="java" import="java.util.*"
contentType="text/html; charset=utf-8"%>
<% request.setCharacterEncoding("utf-8");%>
<%
    String submit1 = request.getParameter("submit1");
    String submit2 = request.getParameter("submit2");
    String method = request.getMethod();
    boolean post = method.equalsIgnoreCase("post"); //method:GET 或POST
    String user = request.getParameter("user");
    if(user==null) user="";
    String name = request.getParameter("name");
    if(name==null) name="";
    String stuid= request.getParameter("stuid");
    if(stuid==null) stuid="";
    String grade = request.getParameter("grade");
    if(grade==null) grade="";
    String grades[] = {"", "", "", ""};
    if(grade.equals("freshman"))
        grades[0] = "checked";
    else if(grade.equals("sophomore"))
        grades[1] = "checked";
    else if(grade.equals("junior"))
        grades[2] = "checked";
    else if(grade.equals("senior"))
        grades[3] = "checked";

<!DOCTYPE HTML>
<html>
<head>
<title>postAndKeep</title>
</head>
<body>
    <form action="postToSelf.jsp?user=<%=user%>" method="post">
        <input type="hidden" name="stuid" value="<%=stuid%>" />
        name: <input type="text" name="name" value="<%=name%>" />
        grade:
        freshman<input type="radio" name="grade" value="freshman" <%=grades[0]%>/>
        sophomore<input type="radio" name="grade" value="sophomore" <%=grades[1]%>/>
        junior<input type="radio" name="grade" value="junior" <%=grades[2]%>/>
        senior<input type="radio" name="grade" value="senior" <%=grades[3]%>/>
        hobby:
        sports<input type="checkbox" name="ah1" value="sports" <%=hobby1%>/>
        travel<input type="checkbox" name="ah2" value="travel" <%=hobby2%>/>
        music<input type="checkbox" name="ah3" value="music" <%=hobby3%>/>
        other: <textarea rows="5" cols="30" name="other"><%=other%> </textarea>
        <input type="submit" name="submit1" value="save"/>
        <input type="submit" name="submit2" value="exit"/>
    </form>
</body>
</html>

```

8、在网页之间传递参数有哪些方法？

1、URL 参数：利用 HTTP 响应形成网页时放在链接里。

`做加法`

用 HTTP 请求传递到下个网页，下一个网页通过 `request.getParameter("x")` 取到值。

2、利用 session

利用服务器端保存<key value>对，服务器利用 HTTP 响应向客户端发回 SessionID(cookie)，根据 SessionID 该服务器可以唯一确定客户端。以后客户端每次向该服务器发出 HTTP 请

求时会发出该 SessionID，服务器端利用该 SessionID 去访问临时存放<key， value>对。当浏览器窗口完全关闭或很长时间没有访问该服务器后该 sessionID 失效。

3、利用 cookie

利用客户端保存<key， value>对，每次向同一个网站发出 HTTP 请求时会一起发出，可以设置保存时间。key=value

4、利用 post

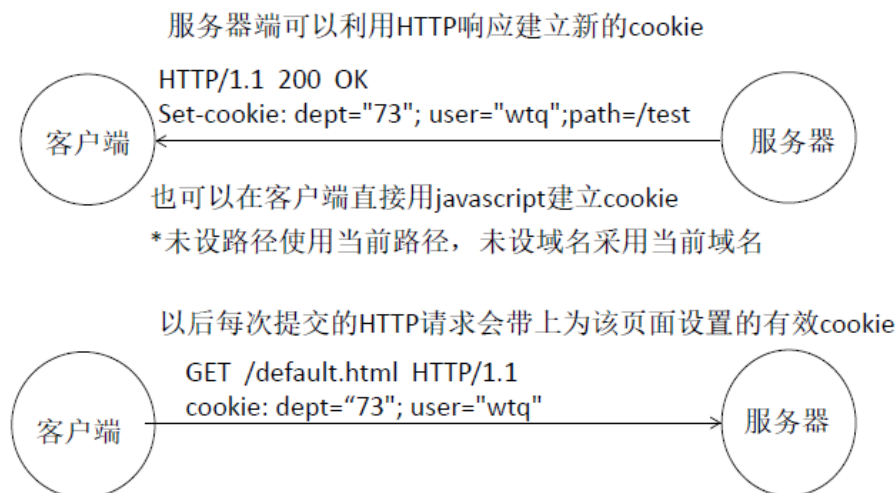
9、说明 Cookie 和 session 的工作原理。

1. 用 Cookie 把数据保存在客户端

- 概述：

- Web 服务器不保留状态。
- Cookie 采用 key-value 方式在客户端保留一些短信息，例如：用户名和密码。当再次访问相同的网站，浏览器会自动把这些数据提交给服务器。
- 每个 cookie 有很多项：名(name)、值(value)、域(domain)、路径(path)、过期时间(expires)。
- 查看 IE 上的 cookie：工具/选项/常规/浏览历史记录/设置/查看文件。

• http中的cookie



* 客户端每次提交网页都会附上在当前页面及其祖先路径上设置的Cookie。

```
Set-Cookie: <name>=<value>; <name>=<value>...
           [< expires=<date> <domain=<domain_name> <path=<some_path> <secure> <httponly>
```

```
<%
Cookie cookie = new Cookie("cookieName","cookievalue");
cookie.setMaxAge(3600); // 设置保留时间 3600秒。
                        // 设置为负值表示只保存在内存，关闭浏览器则消失。
                        // 设置为0表示要删除该cookie。
cookie.setPath("/"); //设置路径为根目录
response.addCookie(cookie);
%>
```

[setCookie.jsp](#)

```
<%
Cookie[] cookies = request.getCookies();
for(Cookie cookie : cookies){
    String name= cookie.getName();    // get the cookie name
    String value=cookie.getValue();    // get the cookie value
}
%>
```

[getCookie.jsp](#)

2. 用 Session 对象把数据保存在服务器端

- 概述:

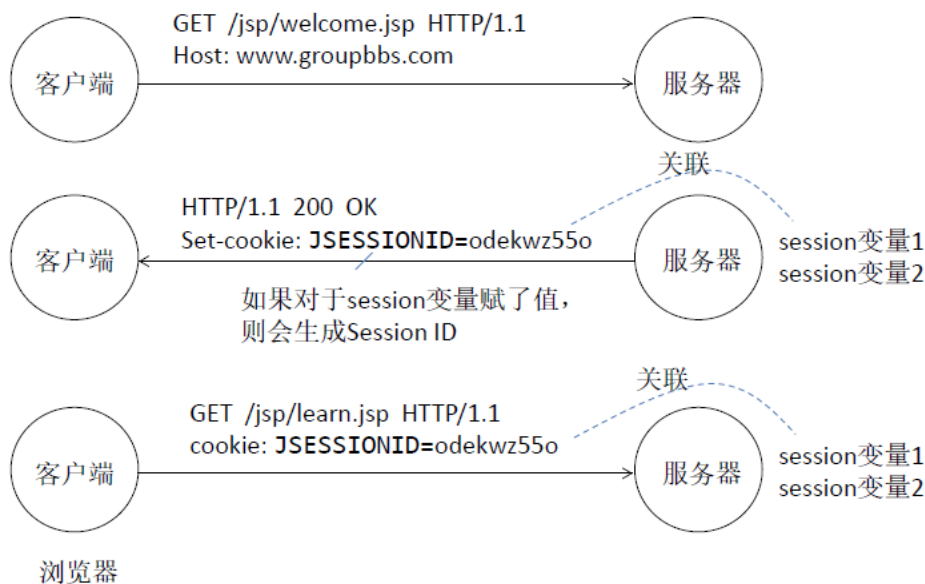
http 协议是无状态的，但是通过 session 可以在 Web 服务器上临时保存少量变量的状态。session 保存的状态只与当前浏览器相关。在用户第一次访问页面时 Web 服务器就会送回 sessionId(作为 cookie)。服务器根据 sessionId 访问 session 变量。

通过服务器端语言(JSP, ASP.net, PHP)设置 session 对象。一个网站的后台程序可以利用 session 变量在后台（服务器）保留状态。session 与 Web 服务器有关，一个 Web 服务器中设置的 session，另一个 Web 服务器访问不了。

如果一个网站的后台程序设置了 session 变量，当通过浏览器第一次访问该网站时，服务器脚本语言就会生成一个唯一的 sessionId，并作为 cookie 传送到客户端保存。以后每次访问该网站，该 session ID 都会和其它 cookie 一起送到服务器。通过 session ID 服务器端可以找到为该 session 保存的变量，使得这些变量可以被不同网页所共享。

session 的生命周期和浏览器有关，浏览器关闭 session 将被删除。但是如果你长时间不关闭浏览器，又不访问该网站，session 的生命周期也会结束，一般是三十分钟。

• http协议中的session



```
// 取出当前浏览器的 session Id<%String id= session.getId(); %>
```

```
//在一个网页中给 session 变量 user 赋值<%session.setAttribute("user", "LiJN"); %>
```

```
//在另一个网页中取出 session 变量 user 的值
```

```
<%String user= (String)session.getAttribute("user"); %>
```

```
//删除 session 变量
```

```
session.removeAttribute("user");
```

10、使用 include 指令可以带来什么好处？

功能：用于在 JSP 转换为 Servlet 程序前将源文件包含进来。include 为指令标签。

语法：<%@ include file="文件的绝对路径【注意：不是 url】或相对路径" %>

说明：该属性指定被包含的文件，不支持表达式，也不允许传递参数。被包含的文件的内容原封不动地插入到主文件中，只要被包含的文件发生改变，整个主页面文件就会重新被编译，编译后只有一个.class 文件。主文件和子文件的 page 指令不应重复。

好处：JSP 可以通过 include 指令来包含其他文件。被包含的文件可以是 JSP 文件、HTML 文件或文本文件。包含的文件就好像是该 JSP 文件的一部分，会被同时编译执行。相当于可以直接插入了一个 JSP 文件，而这个文件可能被多个 JSP 这样包含，使得代码更清晰简洁，同时拓展性较强。

JavaScript

1、js 的函数有何特点？

函数是 Function 类的一个实例，可以带参数并返回执行结果，也就是函数名为引用类型，指向该函数对象。

JavaScript 的函数都是对象的方法，也可以看做对象。

使用 JS 的函数要求先定义再引用。

2、js 的对象有哪些定义方法？

方法	举例	如何引用
用 Object 定义	<pre>var person = new Object(); 等同于 var person = {};</pre>	Person.name = "XXX"
用字面量定义	<pre>var person = {name : "XXX", age : 26, print : function(){alert(person.name)}}; }</pre>	<pre>alert(person.age); alert(person["age"]);</pre>
用构造器的方法创建对象	<pre>function Person(name,age,job){ this.name = name; this.age = age; this.job = job; this.sayName = function(){alert(this.name);};</pre>	<pre>var person1 = new Person("XXX",29,"teacher"); (PS:作为函数，Person 是 对象，用于创建对象， Person 可以看成是类名)</pre>
工厂模式创建对象	<pre>function createPerson(name,age,job) { var o = new Object(); o.name = name; o.age = age; o.job = job; o.sayName=function(){alert(this.name);}; return o;}</pre>	<pre>var person1=createPerson("Nicholas",29,"Software Engineer"); (引用方法同上)</pre>
用 this 代表当前对象	全局函数的 this 为 window 的对象； 作为构造函数为当前对象，在对象方法中也是当前对象。	
直接执行构造函数，即作为	<pre>function Person(name,age,job){ this.name = name;</pre>	

window 对象的一个方法	<pre> this.age = age; this.job = job; this.sayName = function(){ alert(this.name+" "+this.age+ " "+this.job); } }; </pre>	
-----------------------	---	--

3、什么是 js 的闭包？

创建对象时将自动保存外层函数的上下文环境（参数和局部变量）供以后调用其方法时使用。这里的上下文环境就是闭包(closure)。

如果外层为 **window**，其闭包由全局变量组成。

4、什么是原型？其作用是什么？

每个函数（以及每个对象）都会有一个原型（**prototype**）属性。原型对象上可以定义属性和方法。通过函数创建的对象实例会继承其原型对象上定义的属性和方法。如果在对象实例中又定义了同名属性和方法，它们会覆盖原型中的属性和方法。

原型定义举例：

```

function Person() {}
Person.prototype.name="Nicholas"; // 在函数原型上定义新属性
Person.prototype.age=29;
Person.prototype.job="Software Engineer";
Person.prototype.sayName= function(){alert(this.name);};

```

原型的作用：使得函数或者对象可继承，不必重复定义。

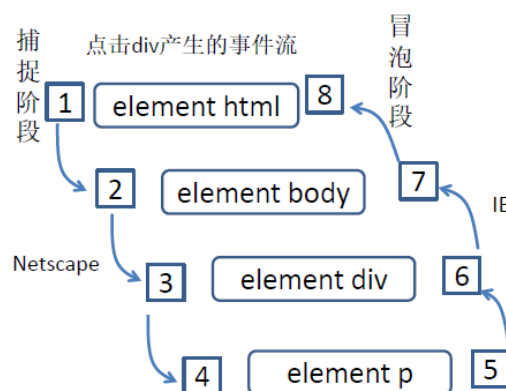
5、js 的事件绑定有哪些方法？

方法 1：采用元素属性 比如 **onclick** 属性

方法 2：采用 **addEventListener** 使用：通过 **getElementById** 获得一个元素，然后对其监听，**p1.addEventListener(“click”,f1,false)**;第一个参数表示要捕捉的事件，第二个参数为捕捉后执行的函数，第三个参数表示在冒泡阶段触发（**false** 为默认，**true** 即为在捕捉阶段触发）。

6、什么是事件的捕捉阶段和冒泡阶段？

浏览器将事件的处理分为两个阶段，一个为捕捉阶段，一个为冒泡阶段，其中冒泡阶段是指浏览器从内层向外层依次处理事件的过程，范围由小到大，捕捉阶段是指浏览器从外层向内层依次处理事件，范围由大到小。



7、js 有哪些鼠标事件？鼠标事件中的 event 有哪些常用属性？event.target 和 event.currentTarget 有何差别？

js 的鼠标事件主要有单击 click，双击 dbclick，按下鼠标 mousedown，放开鼠标 mouseup，进入元素 mouseenter，离开元素 mouseout，在元素上方移动 mousemove，悬浮在元素上方 mouseover。

鼠标事件中 event 的常用属性有：

event.button：取值可以用来判断哪些鼠标键被按下：0 没按键,1 按左键,2 按右键,3 按左右键,4 按中间键,5 按左键和中间键,6 按右键和中间键,7 按所有的键。

event.offsetX,event.offsetY：鼠标相对于触发元素进行定位

event.clientX,event.clientY：鼠标相对于浏览器客户区进行定位

event.pageX,event.pageY：鼠标相对于当前页面进行定位

event.screenX,event.screenY：鼠标相对于屏幕进行定位

event.x,event.y：鼠标相对于父文档进行定位(父文档默认为 BODY，可以是 relative 本体)；

可以取到鼠标事件产生的元素：

鼠标移动离开的对象 event.fromElement、

鼠标移动进入的对象 event.toElement、

触发事件(鼠标点击)的元素 event.srcElement、

冒泡事件中的最小范围元素 event.target

设置当前事件的元素(同 this) event.currentTarget。

二者区别如上，如果有多个鼠标事件，那么用 currentTarget 拿到的是当前正在引发事件的元素，而 target 是事件处于冒泡阶段时，最内层的那个元素。

8、什么是 Ajax 技术？使用它有什么好处？

9、Ajax (Asynchronous Javascript And XML) 是一种创建快速动态网页的技术。通过在后台与服务器进行少量的数据交换，实现网页的异步更新。传统的网页如果需要更新内容，必需重载整个网页。使用 AJAX 技术除了可以利用从 Web 服务器上获取的信息局部更新网页，还可以实现在保持当前网页的情况下提交数据给 Web 服务器。

好处：无刷新更新数据。减少用户等待时间。

异步与服务器通信，不阻塞用户，响应更迅速。

前端和后端负载平衡：将部分服务器负担的功能交给客户端，利用客户端的闲置能力处理，减轻服务器和带宽的负担。

HTTP 协议

1、点击一个网页的链接后取回另一个包含 5 个外部文件(例如，图片)的网页并显示出来需要发出几次 http 请求？

6 次。取回网页需要一次，另外 5 个外部文件各需要一次。

2、利用浏览器地址栏输入 URL 地址，取回一个不包含外部文件的网页并进行提交，需要发出几次 http 请求？

2 次。第一次用于取回网页，第二次提交网页。

3、说明 http 请求中 referer、host、if-modified-since、user-agent 的含义和用途。

referer：产生该请求的网页的 URL。

host：发出请求的主机地址，包括 IP 和端口号。

if-modified-since: 记录了缓存中的文件上次修改的时间。

user-agent: 用户代理。发出 HTTP 请求的客户端版本。

```
GET /jsp/file/img01.jpg HTTP/1.1
Accept: image/png, image/svg+xml, image/jxr, image/*;q=0.8, */*;q=0.5
Referer: http://202.116.76.22:8080/jsp/welcome.jsp
Accept-Language: zh-CN
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) ...Edge/13.10586
Accept-Encoding: gzip, deflate
Host: 202.116.76.22:8080
If-Modified-Since: Sun, 24 Apr 2016 11:14:50 GMT
If-None-Match: W/"5295-1461496490242" ETags
Connection: Keep-Alive
Cookie: JSESSIONID=9FB94A041ECF73217D8A3D147BE1CBA3; username=admin;
password=3214
(空行)
```

q为偏好系数（默认为1）

- 4、提交普通表单(application/x-www-form-urlencoded)与提交多部分表单(multipart/form-data)在 http 请求的正文部分有何差别?

普通表单 HTTP 请求的正文部分以&连接所有提交的参数，多部分表单的 HTTP 请求正文部分有多个 boundary 和空行界定的区域，每一个区域含一个提交参数，用 Content-Disposition 说明该区域提交的是哪个参数。

```
Referer: http://localhost:1272/FormWebSite1/HTML Page2.htm
Content-Type: multipart/form-data; boundary=-----94673242526867
Content-Length: 474

-----94673242526867
Content-Disposition: form-data; name="str"

一个字符串，别管它

-----94673242526867
Content-Disposition: form-data; name="file1"; filename="a.txt"
Content-Type: text/plain

aaaaaaaaaaaaa

-----94673242526867
Content-Disposition: form-data; name="file2"; filename="b.txt"
Content-Type: text/plain

bbbbbbbbbbbbbbbbbb

-----94673242526867--
```

- 5、为什么提交数据在字符编码后还要进行 URL 编码?

因为字符编码都是本地文件所用的编码。而 URL 编码主要用于在网页之间传递参数。并 URL 不支持任何非 ASCII 的编码。如果不对 URL 预先设置编码，页面在取出提交值前系统会自动进行 URL 解码，并且采用的字符是默认的 ISO-8859-1，当与我们的本地字符不符时，很可能造成页面出现乱码问题。

Eclipse 软件

- 1、为什么在本地浏览静态网页可以直接采用操作系统的文件路径而远程浏览静态网页必须使用 URL?

因为本地操作的服务器系统是确定的，所以我们可以使用操作系统的文件路径到对应的目录下寻找对应的文件，远程浏览静态网页时由于我们不清楚服务器的操作系统是什么，比如说我们用的是 windows 的目录系统，而服务器那边是 linux 的系统，那么就无法找到，但 URL 作为一个通用的方法，对操作系统是没有区分的。

2、使用 Eclipse 时是否也可以利用外部浏览器运行其中 JSP 程序？

可以。