

Week 10

day 5

HTML基础补充

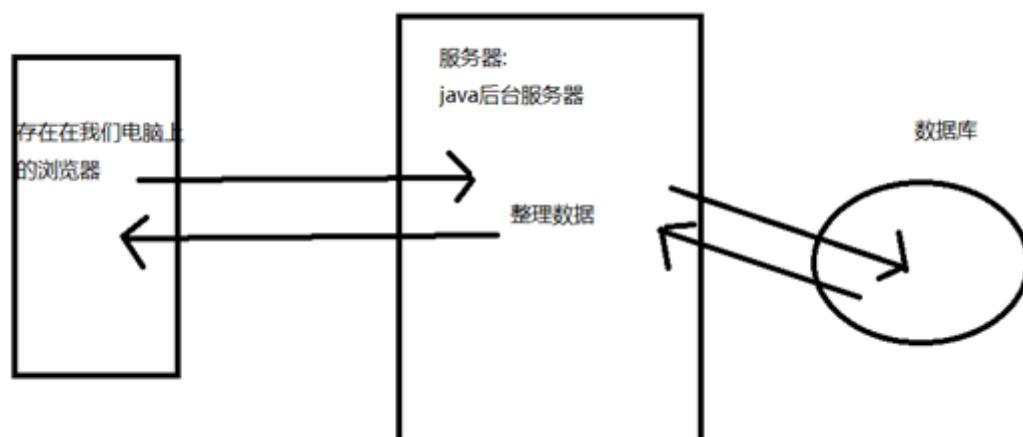
计划

1. 前端基础：html css js dom操作
2. Vue（前端框架）
3. Git HTTP Tomcat

什么是前端

阶段	内容
JavaSE	Java的基础语法
数据结构	数据的组织方式
数据库	存储数据的数据结构
前端	浏览器中的网页
JavaEE	Java的服务器开发
Spring	也是做服务器开发（建立在JavaEE的基础上，封装）
分布式	

什么是服务器



前端：浏览器中的网页

后端：数据 + 数据的处理逻辑

Java服务器拥有整理好的数据，要把数据交给前端
前端拿到数据之后，把数据和前端代码结合到一起，最终在浏览器上显示

HTML

前端语言

弱类型、语法规范不严格

语言	类比
html	人的骨架
css	人的皮肤和血肉
js(JavaScript)	人的思想逻辑
网页	人

www出现的初衷：在不同的主机之间传论文

构建www的要素

- html：承载资源的方式
- http：传输资源的方式
- url：标识目的地址

url:
<https://baike.baidu.com/item/%E8%92%82%E5%A7%86%C2%B7%E4%BC%AF%E7%BA%B3%E6%96%AF%C2%B7%E6%9D%8E/8868412?fr=aladdin>

分三部分： 协议 + 域名/IP + 服务器内部路径（为虚拟路径，是给服务器看的，服务器是由我们实现的Java服务）

IP：唯一标识一台电脑

域名：转化为IP地址

服务器内部路径： 路径 + 参数 （以?分割）

前段语言几个重要的版本

- HTML5 2008
- CSS3 2001
- ES6(JS) 2015

HTML定义

HTML:hyper text markup language 超文本标记语言

- 是个语言
- 是个文本语言
- 是个标记类型的文本语言
- 是一个使用**标签**来描述网页的文本（标签在html文本中有实际意义）
- 用来承载资源的，通过标签标记的形式来承载资源

文本体现在：

- 是一个以html为后缀的文本
- 是一个文本也是一个网页
- 该文本可以用浏览器以网页形式打开（浏览器会根据文本代码解析成网页打开）

超文本：包含文字字体、图片、链接、音乐、程序等元素的文本

一个标准的html

- 在html中标签成对出现，有开始有结束（也存在少量单标签）
- 所有标签都被标签包裹
- 一个html文档分为两个部分： +
暂时只关注body部分的标签，因为页面都是根据body中的标签进行显示
- html是一个用标签描述的文本（具有层级结构）

```
<html>
  <head>
</head>
  <body>
</body>
</html>
```

怎么写html代码

```
<!DOCTYPE html> html文档的声明，说明是html5
<html lang="en"> 语言，没什么用
<head>
  <meta charset="UTF-8"> meta 标签，标识一些页面的元信息：字符集、搜索引擎
  <title>Title</title> 控制页面名
</head>
<body>

</body>
</html>
```

body标签的分类

- 行级标签：在html页面中，放在一行中
- 块级标签：在html页面中独占一行
- 行内块：本质上是一个行级元素（放在一行中），但有一些块级元素的特性（大小边距）

html的标签

1. hr：表示一个条横线，块级标签，单标签 是一条**双线** (不要这样写：

)

2. br：换行符 单标签

注意事项：

- html**代码**中的多个空格或者换行，在页面中只会解释成一个空格
- 自定义标签没有任何实际意义
- html字符：

显示结果	描述	实体名称	实体编号
	空格	 	
<	小于号	<	<
>	大于号	>	>
&	和号	&	&
"	引号	"	"
'	撇号	' (IE不支持)	'
¢	分 (cent)	¢	¢
£	镑 (pound)	£	£
¥	元 (yen)	¥	¥
€	欧元 (euro)	€	€
§	小节	§	§
©	版权 (copyright)	©	©
®	注册商标	®	®
™	商标	™	™
×	乘号	×	×
÷	除号	÷	÷

3. hn：h1-h6是html中的标题标签，只有h1-h6，字体依次缩小

注意：h2-h6都可以在页面代码中重复出现，但h1在1个html文档中只能有1个

问题：为什么h1只能有一个？

因为h1也相当于一个关键词，涉及到搜索引擎优化

SEO：我们在搜索引擎中搜索一个内容，会首先找到搜索引擎的后台服务器的数据库查询

百度会用网络爬虫，获得你的网站数据，存到它的数据库，有关键字标明的优先级会高一点

4. div：对页面进行分区或分节，用来进行页面的大布局（重要）

5. a：超链接标签：从一个页面跳转到另一个页面

```
<a href="http://www.baidu.com">百度</a>
```

```
<--! -->
```

href属性：标识要跳转到的页面的url

target属性：_blank在新页面打开 _self在当前页面打开

注意：target=同组值的问题：当target=自定义的目标名
会首先在浏览器上检测有没有和该目标名相同的页面

如果没有就直接打开一个新页面

如果有就在对应的页面重新打开一个新页面

```
</--!>
```

6. p：段落标签：会在段落前和断落后创建一个边距（与相邻的行）

7. img：图片标签

单位：当什么都不写的时候，是px

写%的时候，是相对于父元素的大小（谁包住它，谁就是它的父元素）

alt：标识图片加载失败的时候

8. input：本质上就是个输入框 type = password, hidden, button, radio

<u>type</u>	button checkbox file hidden image password radio reset submit text	规定 input 元素的类型。
-------------	---	-----------------

9. textarea：多文本 cols rows maxlength placeholder readonly

10. select：下拉选，要和option标签一起使用才具有实际意义

multiple size(框显示的行数) selected

value值：

11. table：标识一个表格，要和tr（行），td（行内的一个单元格）一起使用才具有实际意义

boder 边框 cellpadding 单元格之间的距离 cellspacing 元素到边框的距离

colspan rowspan 合并单元格

th：表头元素（加粗），可以替换td

12. ol：标识有序列表

ul: 标识无序列表

type = 1 a A i l start = 数字 reversed

要和li标签配套使用

form

一个页面怎么产生的(基础版)

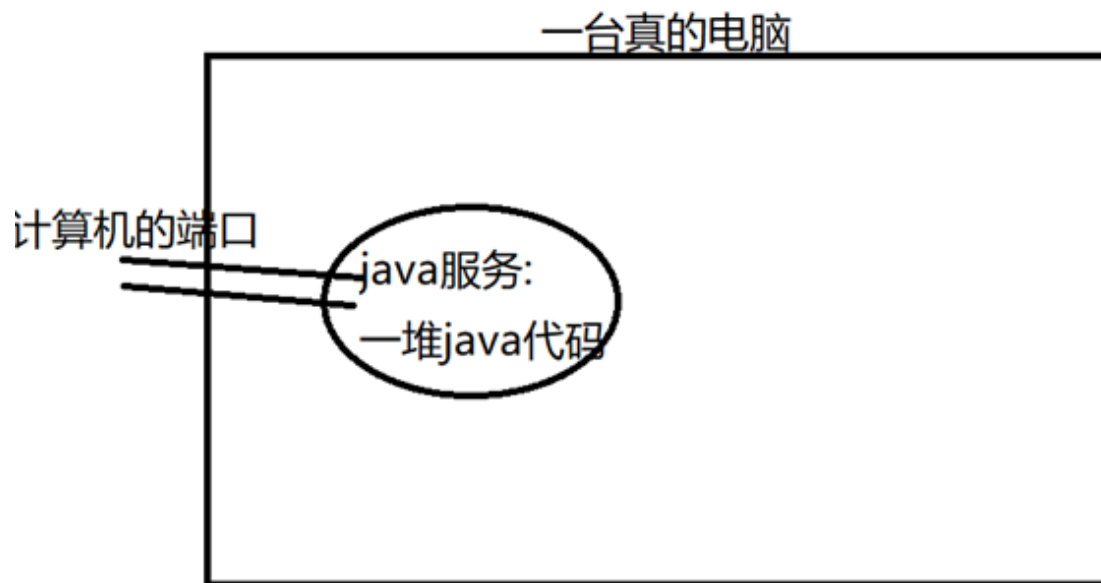


在页面发起一个请求

1. 用户通过某种方式调用浏览器去获取一个页面（代码）（调用浏览器去发起请求）
2. 浏览器拿到用户给的url，把url的第二部分（域名）换成IP地址（通过DNS）
3. 根据IP地址在网络上找到对应的主机
4. 根据端口，进入到该主机的某一个服务中去（某个Java后台会监听这个端口，处理请求）
5. Java后台分析刚刚捕获的请求
6. 根据请求准备数据（先去数据库查询，然后封装数据）
7. 根据分析的请求把数据和对应请求的页面（前端代码）结合起来，获得了一个完整的html页面
8. 把结合好的html页面（代码）返回给浏览器
9. 浏览器获得了一个html代码，解析代码
10. 解析完之后能部分显示（外部资源：如图片、媒体资源还需进一步加载）
11. 单独加载外部资源（图片，单独的css外部文件，单独的js外部文件）
12. 完全正常显示

服务器：

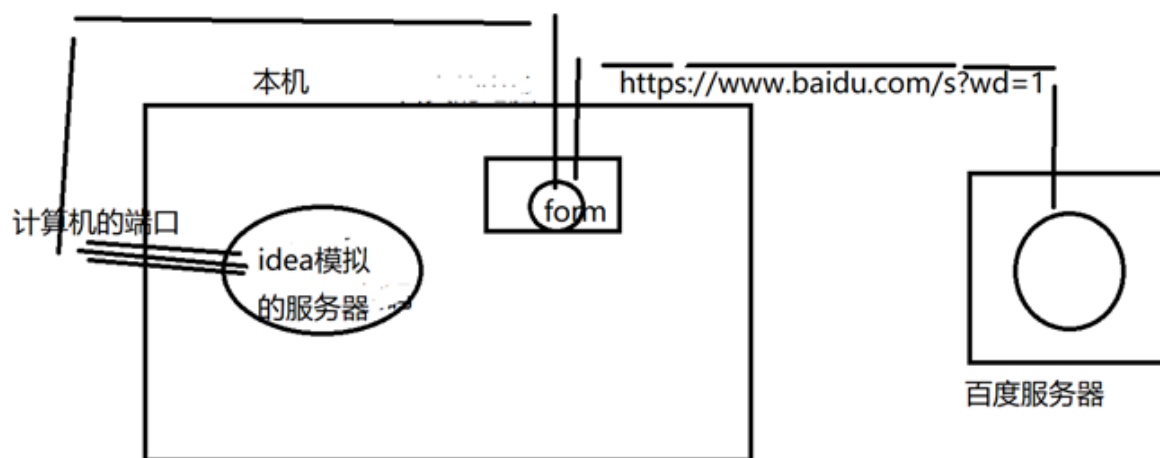
- 物理层面：指的服务所在的主机
- Java后台：我们实现的处理请求的服务（一堆运行在虚拟机之上的字节码文件）



在页面发起一个请求的方式

- 在浏览器中直接输入url
- 超链接
- form表单（在某一个页面内部，发起一个请求，可以携带一些动态参数）

form请求



```
<!--https://www.baidu.com/s?wd=1-->  
<form action="https://www.baidu.com/s" method="get">  
  <input name="wd">  
  <input type="submit">  
</form>
```

action: 是一个不包含参数的url

method: 请求方式 (get post)

get和post的区别:

- 安全性:

- get请求是把请求参数直接拼接在url之后，用户可直接看见
- post一般是把请求参数放在正文中，用户不可直接看见（post也可以把参数拼接在url后）
- **语义化的区别**
- get一般用来获取数据
 - post一般用来提交数据

form中的元素:

- form表单元素（select, textarea, input）（只有表单元素的value才有意义，因为是外界输入）
- 作用：可以构成url?之后的参数，并且这些参数是以key-value的结构拼接在url之后的(get方式)
url前部分+ ? + key=value&key=value
key: 表单元素的name
value: 表单元素的value（就是我们输入的内容）

Week 11

day 1

CSS

css（层叠样式表）要依赖于html存在

css在html中的使用

- **行内**

在html中，直接把css样式写在标签中， style=""

```
<p style="background-color:#FF0000; color:#FFFFFF">
  p标签段落内容。
</p>
```

- **内联**

在html中，把css样式写在标签内部，用