1. Internet和WWW的区别是什么
2. URN的好处以及为什么没有应用。
   1. 资源可以改变位置
   2. 需要一个支持系统来解析资源位置
3. Web1.0 2.0 3.0
   1. 1.0
      1. 门户时代
      2. 信息发布网站、门户网站等
      3. 信息主要由网站运行维护者发布，用户只能查看或评论
      4. 如新浪网易搜狐
   2. 2.0
      1. 搜索社交时代
      2. 交互性
      3. 用户创造内容，网络的主角不是网站而是用户
      4. 如博客中国
   3. 3.0
      1. 大互联网时代
      2. 主动性
4. XHTML比HTML多的要求
   1. 标签必须小写
   2. 标签必须关闭
   3. 标签必须正确嵌套
   4. 属性值必须加引号
   5. 属性不能简写
   6. DOCTYPE不同
5. CSS语法
   1. .类，#id
   2. /\* 注释\*/
6. CSS 内联>内部样式表（定义在head）>外部样式表（link）
   1. <link rel=”style sheet” type=”text/css” href=”xxx.css”>
   2. <head><style type=”text/css”>p{color:red}</style></head>
   3. <p style=”{color:red;}”>
7. CSS P26例 有无style，ID数，属性数，标签数
8. CSS盒模型：margin border padding content
9. HTML5新在何处
   1. 造好轮子，如email格式验证，不需要程序员自己写
   2. 简单性。如<!DOCTYPE html>、标记简化
   3. 通用性。
   4. 为音频、视频、canvas、SVG、语音等提供原生支持，减少使用插件
   5. 安全性好
   6. Article,figure,details,header,footer等新元素，增强语义化
   7. 新表单功能：不需要js，客户端表单验证
   8. 离线应用
10. 语义化的好处
    1. 从SEO的角度看，更容易被搜索引擎准确的分析
    2. 代码干净，可读性好
    3. 还有呢？
11. 响应式设计:RWD responsive web design
    1. 用meta tag重设viewport width
    2. 还有呢？
12. Js和php的对比
    1. 相似
       1. 解释型而非编译型
       2. 语法宽松，弱类型
       3. 大小写敏感
       4. 都内置正则表达式进行文本处理
    2. 不同
       1. Js更oo，php更过程
       2. Js注重用户接口，和html文档的互动；php注重html输出和文件、表单处理
       3. Js运行在客户端浏览器，php运行在服务器
13. <script src=”xxx.js” type=”text/javascript”>
14. 语法
    1. 注释：和java一样
    2. 对象定义：objBob={name:”Bob”, level:3}
    3. 对象增加属性：objBob.age=37
    4. typeof()
    5. int和double都一样
    6. 没有char，char就是长度为1的string。”1” + 1 = “11”
    7. var boolValue = Boolean(otherValue)等价于var boolValue = !!(otherValue)
    8. 5.0==5 true 5.0===5 false 5<”7” true
15. DOM
    1. P49诡异的例子
16. js cookie P59
    1. 创建或修改：document.cookie=”username=Tom”
    2. 删除：document.cookie = "username=; expires=Thu, 01 Jan 1970 00:00:00 UTC";
17. 事件驱动编程
    1. 语法
       1. <a href=”xx.html” onclick=myFunction()>
       2. element.onclick=”myFunction”
       3. (DOM 2) element.addEventListener(“click”, myFunction),第三个参数为true表示capture，false表示bubble
       4. element.removeEventListener(“click”, myFunction)
    2. 阶段
       1. 捕获：从window到target的parent，这一阶段所注册的listener要在event到达target之前处理
       2. 目标：event到达target，本阶段所注册的listener处理之。如果此种event类型是不冒泡的，在此处停止。
       3. 从target的parent到window
    3. 要阻止冒泡，调用event.stopPropagation()
    4. 要阻止调用event的默认行为，调用event.preventDefault()
18. 闭包
    1. var add=(function(){var count=0;return function(){return count+=1;}})()
    2. 实现私有成员
    3. 保护命名空间
    4. 避免污染全局变量
    5. 变量需要长期驻留内存
19. CGI
    1. Common Gateway Interface
    2. 允许浏览器请求服务器运行某一程序（常用perl），而不是请求一份文档
    3. 服务器如何辨别这类请求：
       1. 请求文件的位置（例如cgi-bin）
       2. 请求文件的扩展名
    4. 实现步骤：
       1. 服务器通过请求中包含的信息，设置环境变量
       2. 服务器fork()出子进程，子进程执行程序
       3. cgi程序从环境变量中获得信息
20. PHP lifecycle
    1. 浏览器请求\*.php
    2. 服务器读取文件，运行其代码，产生输出
    3. 将输出返回给浏览器
21. PHP语法
    1. 注释：跟java一样 还多一个#开头
    2. 关键字、类、函数大小写不敏感，变量敏感
    3. 无变量声明，第一次使用即赋值
    4. 变量名$开头
    5. 在函数内使用全局变量：global $x 或 GLOBALS[‘x’]
    6. is\_string(..) gettype(..) $age=(int)”21”
    7. 5 + “7” = 12
    8. isset(NULL) = false
    9. 字符串用”.”连接
    10. “”里面的变量会替换为值，’’里面的不会
    11. foreach($数组名 as $变量名)
    12. function xxx($x,$y=0)
    13. require(x.php)和include的区别：当找不到文件时，require报错并停止，include产生警告并继续
22. POST通过http request；GET通过URL，长度限制2000字符
23. 表单验证的不同方式
    1. 客户端：用户体验好，但是不安全
    2. 服务器端：安全性好，但是速度慢
    3. 两种都用：方便，安全性好，但是开发代价高
24. PHP的构造函数和析构函数是\_\_construct和\_\_destruct
25. MVC优点
    1. 好的架构设计。代码有组织，结构化，容易理解。
    2. 代码维护容易
    3. 容易扩展
26. 框架的优缺点
    1. 优点：
       1. 使用模板
       2. 为常见问题提供了解决方案
       3. 功能抽象
       4. 使快速开发变得更加容易
    2. 缺点
27. 优化思路
    1. HTML CSS JS要使得浏览器渲染顺畅
    2. 减少http请求数
    3. 优化单个http请求
       1. 缩短请求时间
       2. 收发数据量
          1. 有效利用缓存
          2. 收发数据大小
28. 优化技术
    1. CSS优化
       1. 将CSS放在HTML文档顶部
       2. 避免使用某些CSS表达式
       3. 最小化CSS文件
    2. 图片优化
       1. 图片文件直接就应该是想要的大小，不要在HTML里用宽高去resize
       2. 如果可能的话用sprites
    3. JS优化
       1. 将JS放在HTML文档顶部
       2. 使之最小
       3. 放在外部文件中
    4. HTML优化
       1. 标准化
       2. 去掉空白字符
       3. 简化结构
       4. 注意浏览器兼容性、移动兼容性
    5. 服务器优化
       1. utilize Gzip/bzip2 compression
       2. reduce DNS lookups
       3. implement ETags
    6. 用文本代替图片
       1. header里避免使用图片
       2. 文本渲染更快，SEO更好
    7. 缓存：
       1. 服务器cache
       2. 浏览器cache静态资源
       3. Ajax请求也可以cache
    8. 避免Bad Request
    9. PHP优化
       1. 尽量静态
       2. 尽量常量，多使用局部变量
       3. 在循环之前计算循环次数（如数组长度）
       4. 循环中不要使用方法
       5. 用foreach替代for和while
       6. 小文件用fread，大文件用file\_get\_contents
       7. 少用PCRE
       8. 使用cache
29. XML
    1. eXtensible Markup Language
    2. 用途：
       1. 用来交换数据、
       2. 存储数据
       3. 创造新的语言
       4. web服务器用xml存储数据
       5. db有时用xml返回查询结果
       6. web service之间用xml交流
    3. 优点：
       1. 独立于硬件和软件
       2. 降低了数据交换的复杂性
       3. 容易扩展升级
    4. Schema：定义XML的结构
30. DTD？
31. RIA: Rich Internet Applications
    1. 具备传统桌面运用的特性和功能（如高度交互性的GUI）
    2. 在web浏览器或sandbox中运行
    3. 不需要安装软件
32. Ajax
    1. Asynchronous javascript and xml
    2. 优点：
       1. 交互性和响应性变好
       2. 用户体验变好
       3. 减少和服务器的连接
       4. 节省带宽
       5. 缓解网络拥堵
    3. 缺点：
       1. 后退和刷新按钮无效
       2. 不能加入收藏夹
       3. 需要浏览器开启JS
       4. 网络延迟破坏用户体验
       5. 通过Ajax加载的数据不是SEO friendly的
    4. 最重要的组件是JS的XMLHttpRequest
    5. XSS: Cross Site Script
    6. SOP: Same Origin Policy
    7. 2-request limit:一个单用户客户端不能和server或proxy维持超过两个连接
33. web service
    1. RPC: remote procedure call
    2. SOA: service-oriented architecture
    3. REST: representational state transfer
    4. SOAP: simple object access protocol
    5. Mashup:
       1. 一种web应用，把两个或多个第三方source的data、presentation、functionality、集成在一起
       2. 例如：在地图上找到一家宾馆，立马显示出点评