# 20180428

## web服务器

node.js linux/Unix/Windows可以运行，使用JavaScript开发语言

IIS: internet information services 微软Windows平台运行 asp/asp.net

apache 可以在几乎所有的平台linux/Unix/Windows可以运行，可以解析Java，Tomcat，Python

## BC/CS结构

B/S结构（Browser/Server，浏览器/服务器模式），Client/Server架构，即客户端/服务器架构。

B/S最大的优点就是可以在任何地方进行操作而不用安装任何专门的软件。只要有一台能上网的电脑就能使用，客户端零维护。系统的扩展非常容易，只要能上网，再由系统管理员分配一个用户名和密码，就可以使用了。

（在线用密码登陆服务器）

C/S的优点是能充分发挥客户端PC的处理能力，很多工作可以在客户端处理后再提交给服务器。对应的优点就是客户端响应速度快。

（用客户端登陆服务器）

## HTML 5

广义上的HTML5，指的是HTML5+css3+JavaScript6

HTML5的新特性：1. 多媒体video audio 2.地理定位geolocation 3.动画，游戏 4. canvas 5.本地储存。

## 响应式

responsive web design响应式网站设计是一种网络页面设计布局。集中创建页面的图片排版大小，可以智能地根据用户行为以及使用的设备环境进行相对应的布局。（一个页面适用多终端）

*Content is like water*, a saying that illustrates the principles of RWD

## 常见的布局

### 静态布局(static layout)

宽高固定，当窗口小于固定宽高时，横向纵向都通过滚动条来查看。

### 自适应布局（adaptive layout）

为不同的屏幕分辨率设置，使用媒体查询，@media

设备的分类：

超小屏幕--手机（<768px）

小屏幕--平板（>=768px）

中等屏幕--桌面显示器（>=992px）

大屏幕--桌面显示器（>=1200px）

媒体类型：

all 所有

print 打印机

screen 用于电脑屏幕，平板，

speech 发声设备，盲人

媒体功能：

device-height 设备可见屏幕高

device-width 设备可见屏幕宽

max-width

max-height

min-width

min-height

媒体查询的使用语法：

方式1：(常用)

@media 媒体类型 and （媒体功能）{

css代码

}

方式2：

link media=”mediatype媒体类型 and|not|only (媒体功能)”

方式3：

@import url(“\*.css”)

### 流式布局(liquid layout)

流式布局也叫百分比布局，特点是页面的元素的宽度按照屏幕进行适配调整，元素位置不变，元素大小会变。

### 伸缩布局（flex layout）

flexbox伸缩布局盒

### 响应式布局 responsive layout

为不同的屏幕分辨率定义布局，在每个布局中，应用流式布局，页面元素随窗口调整而自动适配。将三种已有的开发技巧（弹性网格布局，弹性图片，媒体，媒体查询）整合起来，并命名为响应式网页设计。

弹性图片、媒体：max-width:100%;

本身图片的分辨率并没有变化，而是通过设置宽高来限定图片的大小（比较浪费流量，现在无限流量时代，不是问题）

为不同的设备做媒体查询（媒体断点）

@media 设备类型 and (媒体功能){

覆盖原来布局css样式

}

## 视口view-port

视口的作用是：让移动设备在页面可以正常显示，width=device-width（页面宽度等于设备宽度）

meta name=”keywords” content=”搜索的关键词”

meta name=”description” content=”网页的描述信息”

meta name=”author” content=”作者”

如何使用viewport

<meta name=”viewport” content=”width=device-width, initial-scale=1.0, minimum-scale=1.0, maximum-scale=1.0, user-scalable=no,minimal-u”>

height高

width宽

device-height 设备屏幕的输出高度

device-width 设备屏幕的输出宽度

initial-scale初始化比例一般设置为1.0

user-scalable 用户是否可以对页面进行缩放 yes/no 一般为No

minimum-scale 允许的最小缩放比例，一般设置为1.0

maximum-scale允许的最大缩放比例，一般设置为1.0

minimal-ui最小用户界面

## 像素

物理尺寸相同，分辨率越高，图片和文字看起来越小；密度越高，分辨率越高，图像和文字看起来越清晰。

分辨率和密度相同的情况下，物理尺寸越小，图像和文字看起来越清晰

设备的像素比：  
 window.devicePixelRatio （比值越大，屏幕的清晰度越高）

屏幕实际尺寸：  
 window.screen.width

屏幕模拟尺寸：

document.documentElement.clientWidth

## 媒体查询

Media Query中，媒体特性min-width和max-width

超小屏幕--手机（<768px）

小屏幕--平板（>=768px）

中等屏幕--桌面显示器（>=992px）

大屏幕--桌面显示器（>=1200px）

@media(min-width: 768px){css 样式} 大于768px时的样式

@media(min-width: 992px){css 样式} 大于992px时的样式

@media(min-width: 1200px){css 样式} 大于1200px时的样式

查看20180428文件夹的index.html

实际应用：@media all and (max-width: 1200px)

# 20180502

Bootstrap 是一个用于快速开发 Web 应用程序和网站的开源前端UI框架。Bootstrap 是基于 HTML、CSS、JAVASCRIPT 的。前端两大核心“布局”“交互”。

## 优点

### 移动设备优先

### 所有的“主流浏览器”都支持 Bootstrap

### 响应式设计

### 容易上手

### 其他优点：开源，web定制，强大的内置组件，易于定制

## bootstrap入门

* + - 1. 导入css
      2. 导入js
      3. 导入依赖的jquery
      4. 导入bootstrap.js

注意：jquery导入应该在导入bootstrap.js之前；

把bootstrap引入项目中时，不要加版本号。方便以后升级。

开发引入源码版本（没有压缩），不是引入压缩过的。

## bootstrap基本使用

### 布局容器：

1） 响应式布局容器 container 设置一个版心

2) 流式布局容器 container-fluid 宽度100%

### 全局样式：

1） p 标签是经过修饰的

2） lead 突出 mark 标记 del删除线 ins下划线 strong加粗 small 小号字体 em斜体

3）文本五种对齐方式

左对齐 text-left

居中对齐 text-center

右对齐 text-right

两端对齐 text-justify

不换行 text-nowrap

1. 文本转化

全转成小写 text-lowercase

全转成大写 text-uppercase

首字母大写 text-capitalize

### 表格：

1） .table 让普通表格变成响应式的表格

2） .table-striped 条纹状表格

3） .table-bordered 带边框的表格

4） .table-hover 鼠标悬停效果

5） .table-condensed 让表格紧凑一点

6） 状态类

.active 鼠标悬停在行或单元格上时所设置的颜色

.success 绿色

.info 蓝色

.warning 黄色

.danger 红色

### 表单：

1） 每一项放在一个 form-group 里面

2） form-control 控制表单输入框的样式的类

### 按钮：

* + - 1. 注意： 要写按钮 先给一个通用的类 .btn

1. 控制按钮的颜色
   * + 1. btn-default 默认
       2. btn-primary 深蓝色
       3. btn-success 绿色
       4. btn-info 浅蓝色
       5. btn-warning 黄色
       6. btn-danger 红色
       7. 2）控制按钮的大小
       8. btn-lg 大
       9. btn-sm 小
       10. btn-xs 超小
       11. 3) 变成块级元素
       12. btn-block

### 图片

<!-- img-rounded圆角 -->

<img src="dn.jpg" alt="..." class="img-rounded">

<!-- img-circle变圆 -->

<img src="dn.jpg" width="400" height="400" alt="..." class="img-circle">

<!--img-thumbnail 圆角边框（缩略图） -->

<img src="dn.jpg" alt="..." class="img-thumbnail">

### 辅助类

#### 文本颜色：

text-muted 暗色

text-primary 亮蓝色

text-success 绿色

text-info 暗蓝色

text-warning 土黄色

text-danger 暗红色

#### 背景色

bg-primary 深蓝色

bg-success 浅绿色

bg-info 浅蓝色

bg-warning 粉红色

bg-danger 淡紫色

#### 关闭按钮x

<button type="button" class="close" aria-label="Close">

<span aria-hidden="true">&times;</span>

</button>

#### 向下实心三角符号

<span class="caret"></span>

#### 浮动

.pull-left: 左浮动

.pull-right: 右浮动

.clearfix： 清除浮动

<div class="clearfix">

<div class="box1 pull-right">div1</div>

<div class="box2 pull-right">div2</div>

</div>

注意: 浮动不用于导航条，导航条有自己的.navbar-left 或 .navbar-right

#### 显示和隐藏

<div class="show">111111111111111111111</div>

<div class="hidden">222222222222222222222</div>

.show 和 .hidden类可以强制任意元素显示或隐藏(****对于屏幕阅读器也能起效****)

### 栅格系统

栅格系统会把一行 分成12列（份），可以通过控制占多少份 来实现多列布局，栅格的外层 必须使用一个 容器 .row 类 包裹起来

<div class="row">

<div class="col-sm-6 bordercolor">.col-sm-6</div>

<div class="col-sm-6 bordercolor">.col-sm-6</div>

</div>

<div class="row">

<div class="col-sm-1 bordercolor">.col-sm-1</div>

<div class="col-sm-1 bordercolor">.col-sm-1</div>

<div class="col-sm-1 bordercolor">.col-sm-1</div>

<div class="col-sm-1 bordercolor">.col-sm-1</div>

<div class="col-sm-1 bordercolor">.col-sm-1</div>

<div class="col-sm-1 bordercolor">.col-sm-1</div>

<div class="col-sm-1 bordercolor">.col-sm-1</div>

<div class="col-sm-1 bordercolor">.col-sm-1</div>

<div class="col-sm-1 bordercolor">.col-sm-1</div>

<div class="col-sm-1 bordercolor">.col-sm-1</div>

<div class="col-sm-1 bordercolor">.col-sm-1</div>

<div class="col-sm-1 bordercolor">.col-sm-1</div>

</div>

### 组件

导航条，下拉框，轮播，模态框。。。。参见手册