# Bootstrap 教程

## 什么是Bootstrap

1. Bootstrap简介

(1)简洁、直观、强悍的前端开发框架，让web开发更迅速、简单。

(2)来自Twitter，是目前很受欢迎的前端框架之一。

(3)Bootstrap是基于HTML、CSS、JavaScript的，让书写代码更容易。

(4)移动优先，响应式布局优先。

1. Bootstrap资源

<http://getbootstrap.com/>

<https://github.com/twbs>

[http://www.bootcss.com/](https://www.bootcss.com/) (中文官网)

下载-基于JQ-引入

注释: bootstrap.js依赖于jquery.js

1. 使用Bootstrap准备工作

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="utf-8">

**<meta http-equiv="X-UA-Compatible" content="IE=edge">**

**<meta name="viewport" content="width=device-width, initial-scale=1">**

**<link rel="stylesheet" href="./css/bootstrap.css">**

</head>

<body>

**<script src="./js/jquery-1.11.1.js"></script>**

**<script src="./js/bootstrap.js"></script>**

</body>

</html>

注释：准备工作①设置IE8浏览器的渲染方式。②设置视口。③引入bootstrap.css。④引入jquery.js。⑤引入bootstrap.js。

## bootstrap栅格系统

1. 容器

布局放在容器里，布局分为12列。

表示行: row 表示列: col

1. 流体布局/**自适应容器**(container-fluid)
2. 固定布局/**固定容器**(container)

固定:1170 970 750 auto

屏幕宽度**阈值**:1200 992 768 768

注释: (1170)-1200-(970)-992-(750)-768-(auto)

实例:

<div class="**container-fluid**">

宽度**自适应**，两边padding为**15px**

</div>

<div class="**container**">

根据屏幕分辨率的阈值区间，**固定容器宽度**(1170/970/750/auto)

两边padding为**15px**

</div>

注释:①自适应容器与固定容器的区别:自适应容器的宽度永远自适应，固定容器的宽度只在屏幕分辨率小于768时自适应，在屏幕宽度大于768时根据阈值区间固定宽度。②由于容器都有默认的padding值，故应避免容器发生嵌套，影响布局的合理性。③可个性化修改Bootstrap默认CSS样式，例如: .container{width: 1000px **!important**;}

1. col-lg-\* 大屏幕 col-md-\*中等屏幕

col-sm-\*小屏幕 col-xs-\* 移动端

大屏幕: lg-1200-md-992-sm-768-xs

实例: <div class="container">

<div class="row">

<div class="**col-md-1**">col-lg-1</div>

<div class="**col-md-11**">col-lg-11</div>

</div>

<div class="row">

<div class="**col-xs-3**">col-xs-1</div>

<div class="**col-xs-9**">col-xs-11</div>

</div>

</div>

注释: ①在栅格系统中，大于当前设置的屏幕尺寸范围时，水平排列；小于当前设置的屏幕尺寸范围时，竖向排列。即: lg只在屏幕分辨率大于1200时有效，md只在屏幕分辨率大于992时有效，sm只在屏幕分辨率大于768时有效，xs在任何屏幕分辨率时都有效。②栅格系统无效时，竖向排列。

注意:①对于容器而言，分两类，一类是自适应容器，一类是固定容器，自适应容器永远自适应，固定容器在移动端自适应，在PC端根据屏幕分辨率阈值区间固定宽度。②栅格系统分四类，xs的适应范围最广，lg的适应范围最窄。

1. 栅格组合模式

<div class="container">

<div class="row">

<div class="**col-lg-3 col-md-4 col-sm-6**">1</div>

<div class="col-lg-3 col-md-4 col-sm-6">2</div>

<div class="col-lg-3 col-md-4 col-sm-6">3</div>

<div class="col-lg-3 col-md-4 col-sm-6">4</div>

</div>

</div>

注释：①随着屏幕分辨率的减少，一行依次显示4,3,2,1个子元素。②class="row"的容器可以抵消class="container"容器的左右15px的内边距。

1. 列偏移

<div class="container">

<div class="row">

<div class="col-lg-4 col-lg-offset-4 col-md-4 col-md-offset-4">col</div>

</div>

</div>

注释:①偏移量为1到12。②可以个性化的设置当屏幕分辨率在不同阈值区间时，网格的宽度和偏移量。

1. 列排序

col-[\*]-push-\*

col-[\*]-pull-\*

<div class="container">

<div class="row">

<div class="col-lg-2 **col-lg-push-10**">col-lg-2</div>

<div class="col-lg-10 **col-lg-pull-2**">col-lg-10</div>

</div>

<div class="row">

<div class="col-lg-4 **col-lg-offset-5**">col-lg-4</div>

<div class="col-lg-4 ">col-lg-4</div>

</div>

<div class="row">

<div class="col-lg-4">col-lg-4</div>

<div class="col-lg-4 **col-lg-push-6**">col-lg-4</div>

</div>

</div>

注释:排序(push/pull)与偏移(offset)的对比—①排序既可以向右移动，又可以向左移动；偏移只能够向右移动。②偏移块与块之间可能发生排挤，不会覆盖，右侧块可能换行；排序块与块之间可能发生覆盖，不会发生排挤，可能出现横向滚动条。

1. 嵌套

<div class="container">

<div class="row">

<div class="col-lg-6">col-lg-6

<div class="row">

<div class="col-lg-4">col-lg-4</div>

<div class="col-lg-4">col-lg-4</div>

<div class="col-lg-4">col-lg-4</div>

</div>

</div>

</div>

</div>

注释:嵌套是以父级元素作为容器。

1. 清除浮动

<div class="container">

<div class="row">

<div class="col-lg-4" style="height:200px">col-lg-4</div>

<div class="col-lg-4">col-lg-4</div>

<div class="col-lg-4">col-lg-4</div>

**<div class="clearfix"></div>**

<div class="col-lg-4">col-lg-4</div>

</div>

</div>

注释:①栅格系统各部分高度不一定统一。②栅格系统排列规则遵循浮动原理。③可通过给空的div添加clearfix类名的方式清除浮动。

## bootstrap响应式工具

1. 概念

针对不同设备展示或隐藏页面内容。

1. 可见类

visible-\*-\*

lg md sm xs

block inline inline-block

含义:屏幕宽度在某一个阈值区间时显示，且显示样式为块级/行内/行内块。

hidden-\*

lg md sm xs

含义:屏幕宽度在某一个阈值空间内隐藏。

实例: <div class="container">

<div class="row">

<div class="**visible-lg-block**">

屏幕宽度在**lg阈值空间**内时以**块级**元素样式显示

</div>

<div class="hidden-sm hidden-xs">

屏幕宽度在**sm和xs阈值空间**内时元素**隐藏**

</div>

</div>

</div>

1. 打印类

visible-print-\* hidden-print

<div class="container">

<div class="row">

<div class="**visible-print-block**">

在打印时可见，样式为块级

</div>

<div class="**hidden-print**">在打印时隐藏</div>

</div>

</div>

1. 实例：天猫侧边栏

<div class="container-fluid">

<div class="row">

<div class="tips1 **affix visible-lg-block**"></div>

<div class="tips2 **affix hidden-lg**"></div>

</div>

</div>

注释:affix类名用来设置固定定位，pull-right类名用来设置右浮动。

## Glyphicons 字体图标

1. 好处: 减少请求、容易控制样式、例子: 淘宝
2. 用法: font-face 字体路径

<span class="glyphicon glyphicon-film"

style="font-size:40px;color:red"></span>

注释: ①将fonts文件夹放在当前html文件的同级目录下。②通过添加类名的方式显示字体图标。③通过设置字体大小和字体颜色的方式修改图标大小和图标颜色。

注意:①字体图标的元素节点的innerHTML应为空。②fonts文件夹不可更改名字。

1. 自制图标:

http://jingyan.baidu.com/article/f79b7cb346cf499145023e78.html

## 预定义样式风格

1. primary(首选项) success(成功) info(一般信息)

warning(警告) danger(危险)

语法:btn/bg/text/alert/panel-

primary/success/info/warning/danger

含义:设置背景颜色、字体颜色、按钮风格、提示框风格、面板风格。

<input type="button" value="默认样式">

<input type="button" value="btn样式" class="btn">

<input type="button" value="首选项" class="btn btn-primary">

<input type="button" value="成功" class="btn btn-success">

<input type="button" value="一般信息" class="btn btn-info">

<input type="button" value="警告" class="btn btn-warning">

<input type="button" value="危险" class="btn btn-danger">

<p>默认段落</p>

<p class="bg-primary/success/info/warning/danger">背景</p>

<span class="text-primary/success/info/warning/danger">

字体颜色

</span>

<div class="alert alert-primary/success/info/warning/danger">

提示

</div>

<div class="panel panel-primary/success/info/warning/danger">

<div class="panel-heading">面板标题</div>

<div class="panel-body">面板主体内容</div>

</div>

注意: 预定义样式风格只是对CSS样式的一个封装，并不针对于某个标签。

<a href="#" class="btn btn-primary">链接</a>

<div href="#" class="btn btn-primary">链接</div>

1. 实例: 登录框

<div class="container">

<div class="panel **panel-primary**">

<div class="panel-heading">

Login

</div>

<div class="panel-body">

<div class="form-group">

<label>用户名：</label>

<input type="text" class="form-control">

<div class="alert **alert-warning**">用户名不能为空</div>

<label>密码:</label>

<input type="password" class="form-control">

</div>

<a href="#" class="**text-info**">忘记密码？</a>

<input type="button" value="登录" class="**btn btn-primary** pull-right">

</div>

</div>

</div>

## 按钮

1. 按钮基础类:btn

<input type="button" value="默认样式">

<input type="button" value="btn样式" class="**btn**">

<input type="button" value="首选项" class="btn **btn-primary**">

<input type="button" value="成功" class="btn **btn-success**">

<input type="button" value="一般信息" class="btn **btn-info**">

<input type="button" value="警告" class="**btn btn-warning**">

<input type="button" value="危险" class="**btn btn-danger**">

注释:一般情况下，按钮应添加基础类btn。

1. 样式: btn-default(默认) btn-link(链接)

<input type="button" value="按钮" class="**btn btn-default**">

<input type="button" value="按钮" class="**btn btn-link**">

1. 大小: btn-\*[lg,sm,xs]

<input type="button" value="btn样式" class="btn">

<input type="button" value="btn样式" class="btn **btn-lg**">

<input type="button" value="btn样式" class="btn **btn-sm**">

<input type="button" value="btn样式" class="btn **btn-xs**">

注释:btn-\*[lg,sm,xs]是对css样式的封装，不仅仅可以作用在按钮上。

1. 状态: active disabled

<input type="button" value="btn样式" class="btn btn-primary">

<input type="button" value="btn样式" class="btn btn-primary **active**">

<input type="button" value="btn样式" class="btn btn-primary **disabled**">

注释:active类设置按钮的默认颜色为一般按钮鼠标悬浮时的颜色，disabled类设置按钮禁用。

1. 种类: a input button

<**input** type="button" value="btn样式" class="btn btn-primary">

<**a** href="#" class="btn btn-primary">按钮</a>

<**button** class="btn btn-primary">按钮</button>

1. 块级: btn-block

<input type="button" value="btn样式" class="btn">

<input type="button" value="btn样式" class="btn btn-block">

注释: **块级按钮的样式为块级，默认占一整行**。

1. 按钮组

**btn-group btn-group-justified btn-group-vertical**

<div class="btn-group">

<input type="button" value="按钮1" class="btn">

<input type="button" value="按钮2" class="btn">

<input type="button" value="按钮3" class="btn">

</div>

注释: btn-group按钮组中的按钮之间连接无间隔。

<div class="btn-group btn-group-justified">

<a href="#" class="btn">按钮1</a>

<a href="#" class="btn">按钮2</a>

<a href="#" class="btn">按钮3</a>

</div>

注释:btn-group btn-group-justified两端横向对齐。

<div class="btn-group btn-group-justified">

<div class="btn-group">

<input type="button" value="按钮1" class="btn">

</div>

<div class="btn-group">

<input type="button" value="按钮2" class="btn">

</div>

<div class="btn-group">

<input type="button" value="按钮3" class="btn">

</div>

</div>

注释:使用a标签做按钮时，直接给父级div添加btn-group 和btn-group-justified类名即可实现两端横向对齐；使用input和button做按钮时，还需要给每个按钮外包裹类名为btn-group的div才可实现两端横向对齐。

<div class="btn-group-vertical">

<a href="#" class="btn btn-info">按钮1</a>

<a href="#" class="btn btn-info">按钮2</a>

<a href="#" class="btn btn-info">按钮3</a>

</div>

注释:设置两端横向对齐时，需要给外层div添加btn-group 和btn-group-justified两个类名，设置纵向两端对齐时，只需要给外层div添加btn-group-vertical一个类名即可。

<div class="btn-group-vertical btn-group-lg">

<a href="#" class="btn btn-info">按钮1</a>

<a href="#" class="btn btn-info">按钮2</a>

<a href="#" class="btn btn-info">按钮3</a>

</div>

注释: 按钮组同样可以设置尺寸大小。

1. 带箭头的按钮

<a href="#" class="btn btn-info">按钮

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

</a>

<button class="btn btn-info">按钮

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

</button>

注释:a和button可以给按钮加箭头，input不可以给按钮加箭头。

箭头可以单独放在一个按钮里:

<div class="btn-group">

<a href="#" class="btn btn-danger">按钮</a>

<a href="#" class="btn btn-danger">

<span class='caret'></span>

</a>

</div>

箭头向上的按钮：

<div class="btn-group dropup">

<button class="btn btn-warning">按钮</button>

<button class="btn btn-warning">

<span class='caret'></span>

</button>

</div>

1. 实例：github按钮组

<div style="font-weight:500">

<div class="btn-group btn-group-sm">

<button class="btn btn-default active">

<span class="glyphicon glyphicon-eye-open"></span> Watch

<span class='caret'></span>

</button>

<button class="btn btn-default ">3,064</button>

</div>

<div class="btn-group btn-group-sm">

<button class="btn btn-default active">

<span class="glyphicon glyphicon-star"></span>

Unstar

</button>

<button class="btn btn-default ">36,296</button>

</div>

<div class="btn-group btn-group-sm">

<button class="btn btn-default active">

<span class="glyphicon glyphicon-fork"></span>

fork

</button>

<button class="btn btn-default ">9,258</button>

</div>

</div>

## 下拉菜单

1. 属性

data-\* 操纵JavaScript交互

aria-\* 针对特殊人群服务(通过屏幕阅读器了解下拉列表具体信息)

role 描述当前状态

open 下拉菜单默认展开

dropdown-menu-riht 下拉菜单默认靠右排列

dropdown-header 下拉菜单标题

divider 一条横线

text-center 文本居中

<div class="dropdown">

<button class="btn btn-default dropdown-toggle" type="button" data-toggle="dropdown">

Dropdown

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

</button>

<ul class="dropdown-menu">

<li><a href="#">Action</a></li>

<li><a href="#">Another action</a></li>

<li><a href="#">Something else here</a></li>

<li><a href="#">Separated link</a></li>

</ul>

</div>

注释: aria-\*与role可以省略。

1. 例子: 运动且带图标的菜单。

.dropdown-menu{ animation:.3s linear fadeIn;}

@keyframes fadeIn{

0%{ opacity:0; transform:translateY(-20px);}

100%{ opacity:1; transform:translateY(0px);}

}

## 标签页/选项卡

1. 头部

nav 选项卡

nav-tabs 选项卡

nav-justified 格子两端对齐

nav-tabs-justified 线两端对齐

nav-pills 药丸样式选项卡

nav-stacked 竖向排列