bootstrap

1. 页面栅格可分为12列

.col-lg/md/sm/xs(表示屏幕的大小)-数字 表示所在的列数

2.偏移量.col-md-offset-\*

这些类会把一个列的左外边距（margin）增加 **\*** 列，其中 **\*** 范围是从 **1** 到 **11**。

3. 一个新的 **.row**，可以在已有的 **.col-md-\*** 列内添加一组 **.col-md-\*** 列。被嵌套的行应包含一组列，这组列个数不能超过12

4. 使用 **.col-md-push-\*（提前）** 和 **.col-md-pull-\*** 类来互换这两列的顺序。

5. 移动设备优先

为了让 Bootstrap 开发的网站对移动设备友好，确保适当的绘制和触屏缩放，需要在网页的 head 之中添加 **viewport meta** 标签

<meta name="viewport" content="width=device-width, initial-scale=1.0">

width=device-width 确保能正确显示在不同设备上

initial-scale=1.0 确保1:1比例显示

*maximum-scale=1.0* 与 user-scalable=no使用禁用缩放功能后，用户只能滚动屏幕

6.响应式图片

<img src="..." class="img-responsive" alt="响应式图像">

.img-responsive 可以让图片更好响应式布局

7.容器：container

8. class="lead" 给段落添加强调文本

9.列表

class="list-unstyled"未定义样式列表

class="list-inline"内连列表

class="dl-horizontal"水平定义的列表

10.Bootstrap允许两种方式显示代码：

第一种是 <code> 标签。如果您想要内联显示代码，那么您应该使用 <code> 标签

第二种是 <pre> 标签。如果代码需要被显示为一个独立的块元素或者代码有多行，那么您应该使用 <pre> 标签。

11.使用<var>a</var>设置变量

12.表格的操作：

.active / .success / .waring / .danger / .info

.table-responsive 可以让表格水平滚动适应不同尺寸的窗口

.table-hover 当鼠标悬停在当前行会显示灰色背景

.tabler-border 表格边框

.table-striped 在<tbody>内的行上显示灰色条纹

13.Bootstrap表单

创建垂直或基本表单的步骤：

* 向父 <form> 元素添加 *role="form"*。
* 把标签和控件放在一个带有 class *.form-group* 的 <div> 中。这是获取最佳间距所必需的。
* 向所有的文本元素 <input>、<textarea> 和 <select> 添加 class ="*form-control*" 。

如果要创建一个内连标签需要在form中添加form-inline类

创建一个水平布局的表单：

* 向父 <form> 元素添加 class *.form-horizontal*。
* 把标签和控件放在一个带有 class *.form-group* 的 <div> 中。
* 向标签添加 class *.control-label*。

禁止表单的方法：第一种在标签中添加 disabled；第二种使用<fieldset disabled></fieldset>包裹要禁用标签

13.Bootstrap图片

Bootstrap 提供了三个可对图片应用简单样式的 class：

* *.img-rounded*：添加 *border-radius:6px* 来获得图片圆角。
* *.img-circle*：添加 *border-radius:50%* 来让整个图片变成圆形。
* *.img-thumbnail*：添加一些内边距（padding）和一个灰色的边框。
* .img-response:响应式

14.Bootstrap辅助类

.pull-left ;元素浮动到左边

.puu;-right :元素浮动到右边

.center-block :元素居中并设置display:block

.clearfix :清除浮动

.show : 强制元素显示

. hidden : 强制元素隐藏

.text-hide : 将页面元素所包含的文本替换成背景

.close : 显示关闭按钮

.caret :显示下拉菜单

15. data-toggle指以什么事件触发，常用的如modal,popover,tooltips等

data-target指事件的目标

16.Bootstrap 字体图标

17.html的role属性：增强语义

18.html中的tabIndex属性可以设置键盘中的TAB键在控件中的移动顺序,即焦点的顺序。 当浏览者使用TAB键在网页控件中移动时，将首先移动到具有最小tabIndex属性值

19.指定下拉菜单，下拉菜单都包裹在.dropdown中；dropdown-menu创建下拉菜单

20.Bootstrap输入框，向输入域添加前缀和后缀的内容

向 **.form-control** 添加前缀或后缀元素的步骤如下：

* 把前缀或后缀元素放在一个带有 class **.input-group** 的 <div> 中。
* 接着，在相同的 <div> 内，在 class 为 **.input-group-addon** 的 <span> 内放置额外的内容。
* 把该 <span> 放置在 <input> 元素的前面或者后面。

21.Bootstrap导航元素

创建一个标签式的导航菜单：

* 以一个带有 class **.nav** 的无序列表开始。
* 添加 class **.nav-tabs**。

.nav nav-tabs :导航

.nav nav-pills :胶囊型导航

.nav nav-pills nav-stacked : 胶囊式导航以垂直方向堆叠排列的

.nav nav-justified :两端对齐的导航

.disabled 禁用选项

.tab-content 与 .tab-pane 和 data-toggle="tab" (data-toggle="pill" ) 一同使用, 设置标签页对应的内容随标签的切换而更改

.tab-pane 与 .tab-content 和 data-toggle="tab" (data-toggle="pill")一同使用, 设置标签页对应的内容随标签的切换而更改

22.Bootstrap-----导航栏

创建一个默认的导航栏的步骤如下：

* 向 <nav> 标签添加 class **.navbar、.navbar-default**。
* 向上面的元素添加 **role="navigation"**，有助于增加可访问性。
* 向 <div> 元素添加一个标题 class **.navbar-header**，内部包含了带有 class **navbar-brand** 的 <a> 元素。这会让文本看起来更大一号。
* 为了向导航栏添加链接，只需要简单地添加带有 class **.nav、.navbar-nav** 的无序列表即可。

导航中的表单<form class="navbar-form navbar-left" role="search">

导航中的按钮<button type="button" class="btn btn-default navbar-btn"> 导航栏按钮 </button>

导航中的文本<p class="navbar-text">Runoob 用户登录</p>

**.navbar-left** 或 **.navbar-right** 向左或向右对齐导航栏中的 *导航链接、表单、按钮或文本* 这些组件。

.navbar navbar-fixed-top 固定在顶部

.navbar navbar-fixed-bottom 固定在底部

.navbar-inverse 创建一个带有黑色背景白色文本的反色的导航栏

.breadcrumb 面包屑导航 一般作用在无需列表显示发布日期、类别或标签

23.Bootstrap分页

.pagination 一般作用在无序列表 显示分页

&laquo;左双箭头 &raquo; 右双箭头

Bootstrap 翻页

.pager 一般作用在无序列表 显示翻页

.previous 向左对齐 .next 向右对齐

&larr; 向左单箭头 &rarr;向右单箭头

24.Bootstrap标签

标签可用于计数、提示或页面上其他的标记显示。**.label** 来显示标签

可以使用修饰的 class： **label-default、label-primary、label-success、label-info、label-**

**warning、label-danger** 来改变标签的外观

25.Bootstrap徽章

徽章与标签相似，主要的区别在于徽章的边角更加圆滑。徽章（Badges）主要用于突出显示新的或未读的项。

.badge 徽章

26.Bootstrap超大屏幕

超大屏幕（Jumbotron）的步骤如下：

* 创建一个带有 class **.jumbotron**. 的容器 <div>。
* 除了更大的 <h1>，字体粗细 *font-weight* 被减为 200。

27.Bootstrap缩略图

Bootstrap 创建缩略图的步骤如下：

* 在图像周围添加带有 class **.thumbnail** 的 <a> 标签。
* 这会添加四个像素的内边距（padding）和一个灰色的边框。
* 当鼠标悬停在图像上时，会动画显示出图像的轮廓。

在缩略图中添加自定义内容

具体步骤如下：

* 把带有 class **.thumbnail** 的 <a> 标签改为 <div>。
* 在该 <div> 内，您可以添加任何您想要添加的东西。由于这是一个 <div>，我们可以使用默认的基于 span 的命名规则来调整大小。
* 如果您想要给多个图像进行分组，请把它们放置在一个无序列表中，且每个列表项向左浮动。

28 Bootstrap警告

通过创建一个 <div>，并向其添加一个 **.alert** class 和四个上下文 class（即**.alert-success、.alert-info、.alert-warning、.alert-danger**）之一，来添加一个基本的警告框。

创建一个可取消的警告（Dismissal Alert）步骤如下：

* 通过创建一个 <div>，并向其添加一个 **.alert** class 和四个上下文 class（即**.alert-success、.alert-info、.alert-warning、.alert-danger**）之一，来添加一个基本的警告框。
* 同时向上面的 <div> class 添加可选的 **.alert-dismissable**。
* 添加一个关闭按钮。

29.Bootstrap进度条

创建一个基本的进度条的步骤如下：

* 添加一个带有 class **.progress** 的 <div>。
* 接着，在上面的 <div> 内，添加一个带有 class **.progress-bar** 的空的 <div>。
* 添加一个带有百分比表示的宽度的 style 属性，例如 style="60%"; 表示进度条在 60% 的位置。

创建交替的进度条: .progress-bar-success/info/waring/danger

创建条纹的进度条: 在.progress同等级标签内添加类.progress-striped

创建动画的进度条：在.progress和.progress-striped类同等级标签内添加active

30.Bootstrap多媒体对象

作用：媒体对象可以用更少的代码来实现媒体对象与文字的混排。

两种形式来设置媒体对象：

* **.media**：该 class 允许将媒体对象里的多媒体（图像、视频、音频）浮动到内容区块的左边或者右边。
* **.media-list**：如果你需要一个列表，各项内容是无序列表的一部分，可以使用该 class。可用于评论列表与文章列表。

31.Bootstrap列表组

作用：列表组件用于以列表形式呈现复杂的和自定义的内容

创建一个基本的列表组的步骤如下：

* 向元素 <ul> 添加 class **.list-group**。
* 向 <li> 添加 class **.list-group-item**。

可以向任意的列表项添加徽章组件，它会自动定位到右边。

32.Bootstrap面板

作用：面板组件用于把 DOM 组件插入到一个盒子中。

创建一个基本的面板，只需要向 <div> 元素添加 class **.panel** 和 class **.panel-default** 即可

33.Bootstrap Well

Well 是一种会引起内容凹陷显示或插图效果的容器 <div>。

34.Bootstrao (Modal)模态框

模态框（Modal）是覆盖在父窗体上的子窗体。通常，目的是显示来自一个单独的源的内容，可以在不离开父窗体的情况下有一些互动。

可以切换模态框（Modal）插件的隐藏内容：

* **通过 data 属性**：在控制器元素（比如按钮或者链接）上设置属性 **data-toggle="modal"**，同时设置 **data-target="#identifier"** 或 **href="#identifier"** 来指定要切换的特定的模态框（带有 id="identifier"）。
* **通过 JavaScript**：使用这种技术，您可以通过简单的一行 JavaScript 来调用带有 id="identifier" 的模态框：

$('#identifier').modal(options)

* **aria-labelledby="myModalLabel"**，该属性引用模态框的标题。
* 属性 **aria-hidden="true"** 用于保持模态窗口不可见，直到触发器被触发为止（比如点击在相关的按钮上）。
* <div class="modal-header">，modal-header 是为模态窗口的头部定义样式的类。
* **class="close"**，close 是一个 CSS class，用于为模态窗口的关闭按钮设置样式。
* **data-dismiss="modal"**，是一个自定义的 HTML5 data 属性。在这里它被用于关闭模态窗口。
* **class="modal-body"**，是 Bootstrap CSS 的一个 CSS class，用于为模态窗口的主体设置样式。
* **class="modal-footer"**，是 Bootstrap CSS 的一个 CSS class，用于为模态窗口的底部设置样式。
* **data-toggle="modal"**，HTML5 自定义的 data 属性 data-toggle 用于打开模态窗口。

.data-backdrop : 当用户点击模态框外部时不会关闭模态框。

.data-keyboard: 当按下 escape 键时关闭模态框，设置为 false 时则按键无效

.data-show: 当初始化时显示模态框。

.data-remove: 使用 jQuery *.load* 方法，为模态框的主体注入内容。如果添加了一个带有有效 URL 的 href，则会加载其中的内容。

事件

模态框的事件可在函数中当钩子使用

show.bs.modal 在调用show方法后触发

shown.bs.modal 当模态框对用户可见识触发

hide.bs.modal 当调用hide实例方法时触发

hiden.bs.modal 当模套框对用户完全隐藏时触发

35.Bootstrap 滚动监听（Scrollspy）插件

作用：滚动监听（Scrollspy）插件，即自动更新导航插件，会根据滚动条的位置自动更新对应的导航目标。其基本的实现是随着您的滚动，基于滚动条的位置向导航栏添加 **.active** class。

操作：向您想要监听的元素（通常是 body）添加 **data-spy="scroll"**。然后添加带有 Bootstrap **.nav** 组件的父元素的 ID 或 class 的属性 **data-target**。

**.scrollspy('refresh')**：当通过 JavaScript 调用 scrollspy 方法时，您需要调用 **.refresh** 方法来更新 DOM。

active.bs.scrollspy 每当一个新项目被滚动监听激活时，触发该事件。

'data-offset' 属性的值，是一个数字，决定了当计算滚动条位置时从顶部开始偏移的像素数。

36 Bootstrap标签页（tab）

通过以下两种方式启用标签页：

**通过 data 属性**：您需要添加 **data-toggle="tab"** 或 **data-toggle="pill"** 到锚文本链接中。添加 **nav** 和 **nav-tabs** 类到 **ul** 中，将会应用 Bootstrap [标签样式](http://www.runoob.com/bootstrap/bootstrap-navigation-elements.html)，添加 **nav** 和 **nav-pills** 类到 **ul** 中，将会应用 Bootstrap [胶囊式样式](http://www.runoob.com/bootstrap/bootstrap-navigation-elements.html)。

**通过 JavaScript**：您可以使用 Javscript 来启用标签页

淡入淡出效果：如果需要为标签页设置淡入淡出效果，请添加 **.fade** 到每个 **.tab-pane** 后面。第一个标签页必须添加 **.in** 类，以便淡入显示初始内容

方法：

**$().tab**：该方法可以激活标签页元素和内容容器。标签页需要用一个 **data-target** 或者一个指向 DOM 中容器节点的 **href**。

事件：

show.bs.tab 该事件用在标签页显示时触发，但必须在新标签页显示前分别使用 **event.target** 和 **event.relatedTarget** 来定位到激活的标签页和前一个激活的标签页。

shown.bs.tab 该事件在标签页显示时触发，但是必须在某个标签页已经显示之后。分别使用 **event.target** 和 **event.relatedTarget** 来定位到激活的标签页和前一个激活的标签页

37.Bootstrap 提示(Tooltip)工具插件：不是纯 CSS 插件您必须使用 jquery 激活它

用法：提示工具（Tooltip）插件根据需求生成内容和标记，默认情况下是把提示工具（tooltip）放在它们的触发元素后面。您可以有以下两种方式添加提示工具（tooltip）：

**通过 data 属性**：如需添加一个提示工具（tooltip），只需向一个锚标签添加 **data-toggle="tooltip"** 即可。锚的 title 即为提示工具（tooltip）的文本。默认情况下，插件把提示工具（tooltip）设置在顶部。

**通过 JavaScript**：通过 JavaScript 触发提示工具（tooltip）

激活Tooltip：

$(function () { $("[datatoggle='tooltip']").tooltip(); });

选项：有一些选项是通过 Bootstrap 数据 API（Bootstrap Data API）添加或通过 JavaScript 调用的

data-animation: 默认值true 提示工具使用 CSS 渐变滤镜效果。

data-html: 默认值false 向提示工具插入 HTML。如果为 false，jQuery 的 text 方法将被用于向 dom 插入内容。

data-placement:类型string/function 默认值top 规定如何定位提示工具

data-selector :类型string 默认值false 如果提供了一个选择器，提示工具对象将被委派到指定的目标。

data-title:类型string/function 默认值‘’ 如果未指定 *title* 属性，则 title 选项是默认的 title 值。

data-trigger :类型string 默认值*'hover focus'* 定义如何触发提示工具

data-content: 类型string/function默认值’’

data-delay:类型number/object 默认值0延迟显示和隐藏提示工具的毫秒数

data-container:类型string/false 默认值false 向指定元素追加提示工具。

方法：下面是一些提示工具（Tooltip）插件中有用的方法：

Options:tooltip(options) 向元素集合附加提示工具句柄。

Toggle:tooltip(toggle) 切换显示/隐藏元素的提示工具。

Show:tooltip(show) 显示元素的提示工具。

Hide:tooltip(hide) 隐藏元素的提示工具。

Destroy:tooltip(destroy) 隐藏并销毁元素的提示工具。

事件：

show.bs.tooltop 当调用 show 实例方法时立即触发该事件

shown.bs.tooltip 当提示工具对用户可见时触发该事件（将等待 CSS 过渡效果完成）。

hide.bs.tooltip 当调用 hide 实例方法时立即触发该事件。

hidden.bs.tooltip 当提示工具对用户隐藏时触发该事件（将等待 CSS 过渡效果完成）。

38 Bootstrap弹出框(Popover)：不是纯 CSS 插件您必须使用 jquery 激活它

需激活弹出框，用户只需把鼠标悬停在元素上即可。

用法：

弹出框（Popover）插件根据需求生成内容和标记，默认情况下是把弹出框（popover）放在它们的触发元素后面。您可以有以下两种方式添加弹出框（popover）：

**通过 data 属性**：如需添加一个弹出框（popover），只需向一个锚/按钮标签添加 **data-toggle="popover"** 即可。锚的 title 即为弹出框（popover）的文本。默认情况下，插件把弹出框（popover）设置在顶部。

**通过 JavaScript：**通过 JavaScript 启用弹出框（popover）

选项：类似Tooltip，见上

39 Bootstrap警告框(Alert)

有以下两种方式启用警告框的可取消（dismissal）功能：

**通过 data 属性**：通过数据 API（Data API）添加可取消功能，只需要向关闭按钮添加 **data-dismiss="alert"**，就会自动为警告框添加关闭功能。

**通过 JavaScript**：通过 JavaScript 添加可取消功能

方法：

alert() 该方法让所有的警告框都带有关闭功能。

Close Method:alert(‘close’) 关闭所有的警告框

事件：

close.bs.alert()当调用 *close* 实例方法时立即触发该事件。

closed.bs.alert()当警告框被关闭时触发该事件（将等待 CSS 过渡效果完成）。

40 Bootstrap 按钮插件

通过按钮（Button）插件，您可以添加进一些交互，比如控制按钮状态，或者为其他组件（如工具栏）创建按钮组。

加载状体：只需要简单地向 button 元素添加 **data-loading-text="Loading..."** 作为其属性即可

如需激活单个按钮的切换（即改变按钮的正常状态为按压状态，反之亦然），只需向 button 元素添加 **data-toggle="button"**作为其属性即可

可以创建复选框组，并通过向 **btn-group** 添加 data 属性 **data-toggle="buttons"** 来添加复选框组的切换。

可以创建单选按钮组，并通过向 **btn-group** 添加 data 属性 **data-toggle="buttons"** 来添加单选按钮组的切换。

用法：

**通过 JavaScript** 启用按钮（Button）插件 $(‘.btn’).button()

方法：

button(toggle“) 切换按压状态。赋予按钮被激活的外观。您可以使用 **data-toggle** 属性启用按钮的自动切换。

button(‘loading’) 当加载时，按钮是禁用的，且文本变为 button 元素的 **data-loading-text** 属性的值

button(‘reset’) 重置按钮状态，文本内容恢复为最初的内容。

button(‘string’) 该方法中的字符串是指由用户声明的任何字符串使用该方法，重置按钮状态，并添加新的内容。

41.Bootstrap折叠(Collapse)插件

创建可折叠的面板：

* 1. **data-toggle="collapse"** 添加到您想要展开或折叠的组件的链接上。
  2. **href** 或 **data-target** 属性添加到父组件，它的值是子组件的 *id*。
  3. **data-parent** 属性把折叠面板（accordion）的 id 添加到要展开或折叠的组件的链接上。

方法：

类： .collapse隐藏内容

.collapse in 显示内容。

.collapsing 当过渡效果开始时被添加，当过渡效果完成时被移除。

通过以下两种方式使用折叠（Collapse）插件：

* **通过 data 属性**：向元素添加 **data-toggle="collapse"** 和 **data-target**，自动分配可折叠元素的控制。**data-target** 属性接受一个 CSS 选择器，并会对其应用折叠效果。请确保向可折叠元素添加 class **.collapse**。如果您希望它默认情况下是打开的，请添加额外的 class **.in**。
* **通过 JavaScript**：可通过 JavaScript 激活 collapse 方法

$().collapse();

选项：

data-parent 类型selector 默认值false 如果提供了一个选择器，当可折叠项目显示时，指定父元素下的所有可折叠的元素将被关闭

data-toggle 类型Boolean 默认值 true 切换调用可折叠元素。

方法：

Options:.collapse(options) 激活内容为可折叠元素。接受一个可选的 options 对象。

Toggle:.collapse(‘toggle’) 切换显示/隐藏可折叠元素

Show:.collapse(‘show’) 显示可折叠元素。

Hide:collapse(‘hide’) 隐藏可折叠元素。

事件：

show/shown/hide/hidden .bs.collapse

41 Bootstrap 轮播插件(Carousel)

步骤：

1. 轮播（Carousel）指标

div.carousel slide #myCarousel

ul.carousel-indicators

li data-target=’#myCarousel’ data-slide-to=’0’

</div>

2. 轮播（Carousel）项目

<div class=’carousel-inner’>

div.item

img

</div>

3. 轮播（Carousel）导航

<a href="#myCarousel" class="carousel-control left" data-slide="prev">&l</a>

.carousel-caption 可添加标题

**用法**

* **通过 data 属性**：使用 data 属性可以很容易控制轮播（Carousel）的位置。
  + 属性 **data-slide** 接受关键字 *prev* 或 *next*，用来改变幻灯片相对于当前位置的位置。
  + 使用 **data-slide-to** 来向轮播传递一个原始滑动索引，**data-slide-to="2"** 将把滑块移动到一个特定的索引，索引从 0 开始计数。
  + **data-ride="carousel"** 属性用于标记轮播在页面加载时就开始动画播放。
* **通过 JavaScript**：轮播（Carousel）可通过 JavaScript 手动调用 $().carousel();

选项：

data-interval 类型number 默认值5000 自动循环每个项目之间延迟的时间量

data-pause 类型 string 默认值 hover 鼠标进入时暂停轮播循环，鼠标离开时恢复轮播循环

data-wrap 类型 Boolean 默认值 true 轮播是否连续循环

方法：

carousel(options) 初始化轮播为可选的 options 对象，并开始循环项目

carousel(‘cycle’) 从左到右循环轮播项目。

carousel(‘pause’) 停止轮播循环项目。

carousel(number) 循环轮播到某个特定的帧（从 0 开始计数，与数组类似）。

carousel(‘prev/next’) 循环轮播到上/下一个项目。

事件：

slide.bs.carousel 当调用 slide 实例方法时立即触发该事件。

slid.bs.carousel 当轮播完成幻灯片过渡效果时触发该事件。

42 Bootstrap 附加导航插件(Affix)

附加导航（Affix）插件允许某个 <div> 固定在页面的某个位置。您也可以在打开或关闭使用该插件之间进行切换。将在某个位置开始，但当页面点击某个标记，该 <div> 会锁定在某个位置，不会随着页面其他部分一起滚动。

用法：

您可以通过 data 属性或者通过 JavaScript 来使用附加导航（Affix）插件。

* **通过 data 属性**：如需向元素添加附加导航（Affix）行为，只需要向需要监听的元素添加 **data-spy="affix"** 即可。请使用偏移来定义何时切换元素的锁定和移动。
* **通过 JavaScript**：您可以通过 JavaScript 手动为某个元素添加附加导航（Affix）