
Table of Contents

介绍	1.1
KickOff Party	1.2
1.1 教学目的	1.2.1
1.2 软件和工具	1.2.2
1.3 教学方法	1.2.3
1.4 WEB基本知识和历史	1.2.4
1.5 课前自学	1.2.5
Intro to HTML	1.3
2.1 初识HTML	1.3.1
2.2 HTML常用术语	1.3.2
2.3 解剖网站	1.3.3
2.4 动手 - 修改更多HTML	1.3.4
Intro to CSS	1.4
3.1 初识CSS	1.4.1
3.2 CSS常用术语	1.4.2
3.3 CSS盒子模型	1.4.3
3.4 使用Chrome开发者工具调试样式	1.4.4
3.5 动手 - 修改更多CSS样式	1.4.5
Intro to Bootstrap	1.5
4.1 布局	1.5.1
4.2 基础Bootstrap CSS	1.5.2
4.3 组件	1.5.3
成果展示	1.6

Introduction

此教程是专门为 Girls Coding Day 女性编程素养教育而设计。

工作坊结束后，学员将具备能力在Bootstrap模板的基础上，做出 [Girls Coding Day](#) 这样的静态网站。

学员将获得：

1. 修改模板的基本HTML和CSS的能力。
2. 修改bootstrap 的能力。
3. 部署在Github Page的能力。
4. 自学编程的基本能力。
5. 一群热爱编程的闺蜜以及友好的程序员社群。

Kickoff Party

本章节目标

1. 下载和安装工作坊所需要的软件
2. 了解工作坊提倡的学习方法
3. 了解基本的Web基础知识
4. 自学HTML和CSS基本概念

1.1 教学目的

课程结束后具备能力在Bootstrap模板的基础上，做出[Girls Coding Day](#)这样的个人网站。

1. 修改模板的基本HTML和CSS的能力。
2. 修改bootstrap 的能力
3. 部署在*Github Page*的能力

1.2 软件和工具

Cooking time: 30min active / 30-45min passive

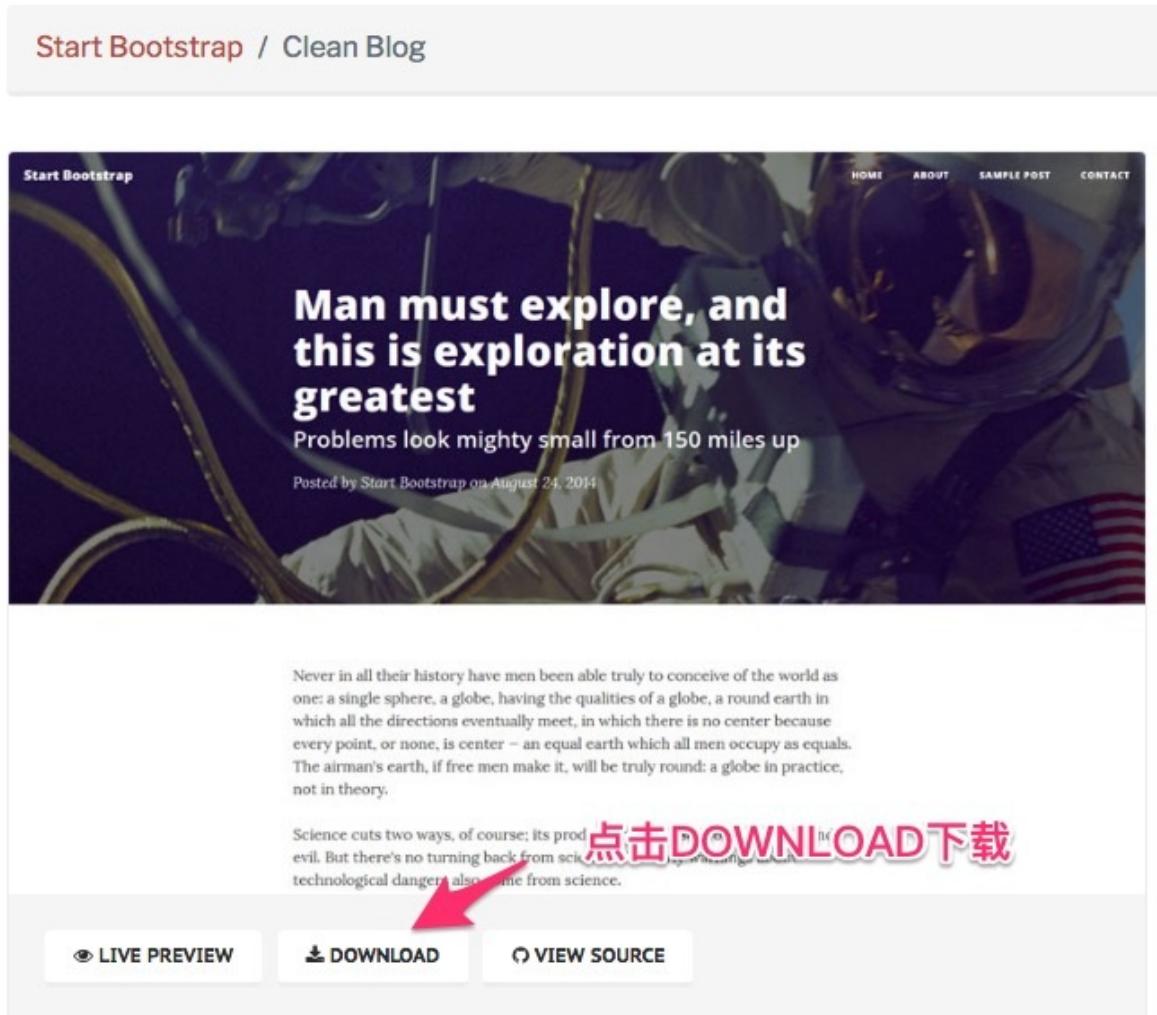
- 下载并安装 [浏览器 Chrome](#)
- 下载并安装 [文本编辑器 Atom](#)
 - 你是否好奇「文本编辑器」是什么？那么赶紧用搜索引擎搜索一下关键字「文本编辑器」吧。
 - 略读 [文本编辑器 - 维基百科](#)
- 注册[GitHub](#)
 - Github能用来做什么？请略读 [怎样使用 GitHub？ - 知乎](#)
- 安装[GitHub Desktop](#)(请下载最新的Desktop Beta版本)
 - 阅读并操作 [Authenticating to GitHub](#)
 - 按照阅读内容，配置好[github.com](#) 的帐号信息

Tasks For Tutors: 请教练介绍一下这些软件。tips：尽量使用类比方法，让学员快速理解这些软件的用途。

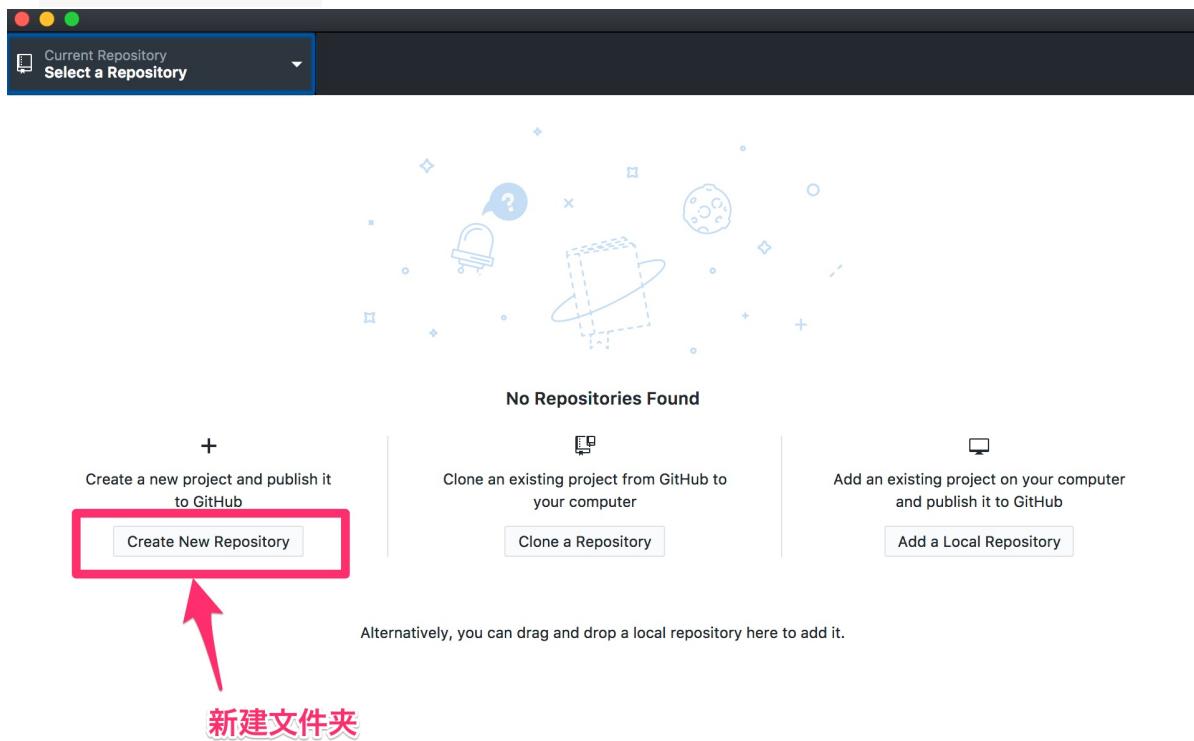
Github Desktop 配置

1. 去[Clean Blog - Bootstrap Blog Theme - Start Bootstrap](#)， 下载免费Bootstrap模板，保存在桌面（或者是你想要放代码的文件夹）。

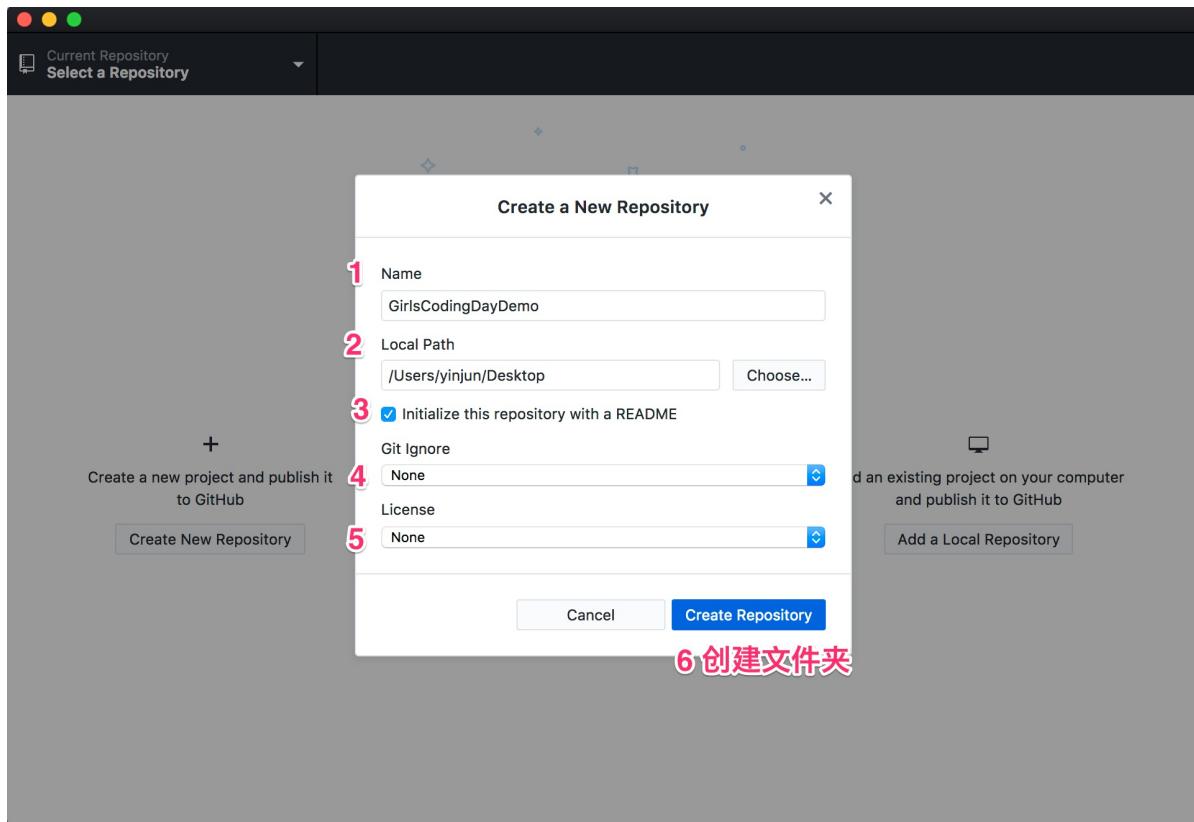
1.2 软件和工具



2. 打开Github Desktop，在桌面上（或者是你想要放代码的文件夹中）创建一个名为 GirlsCodingDayDemo 的文件夹。

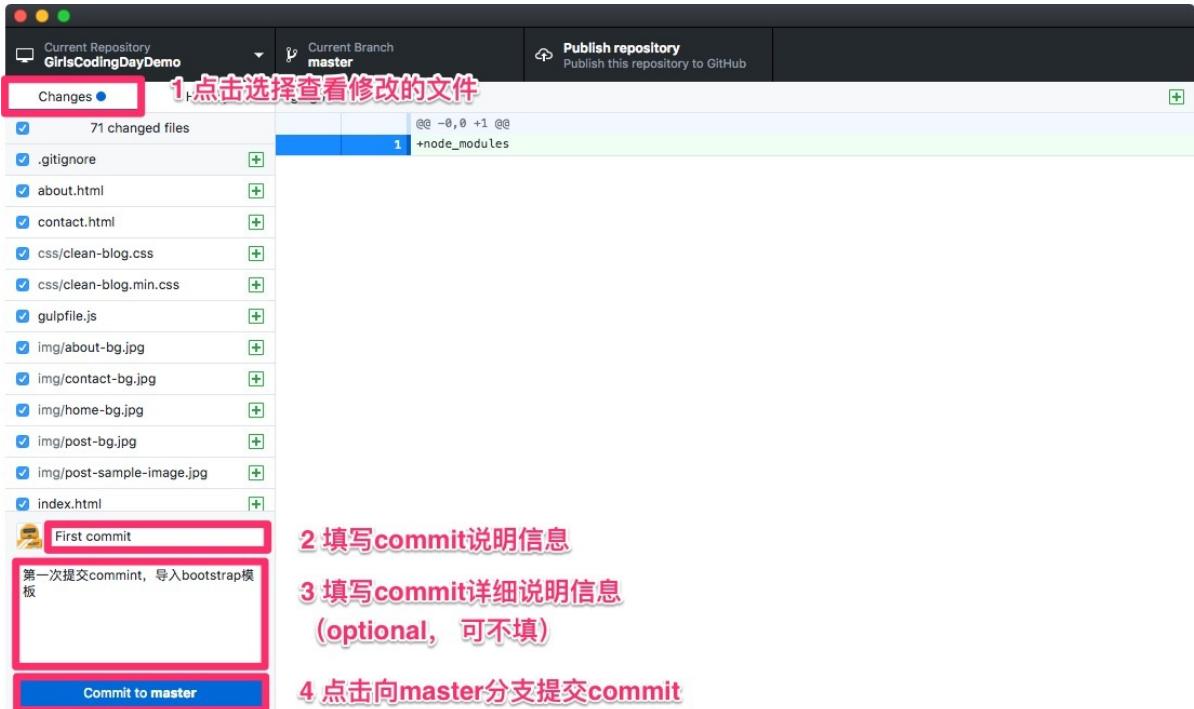


1.2 软件和工具

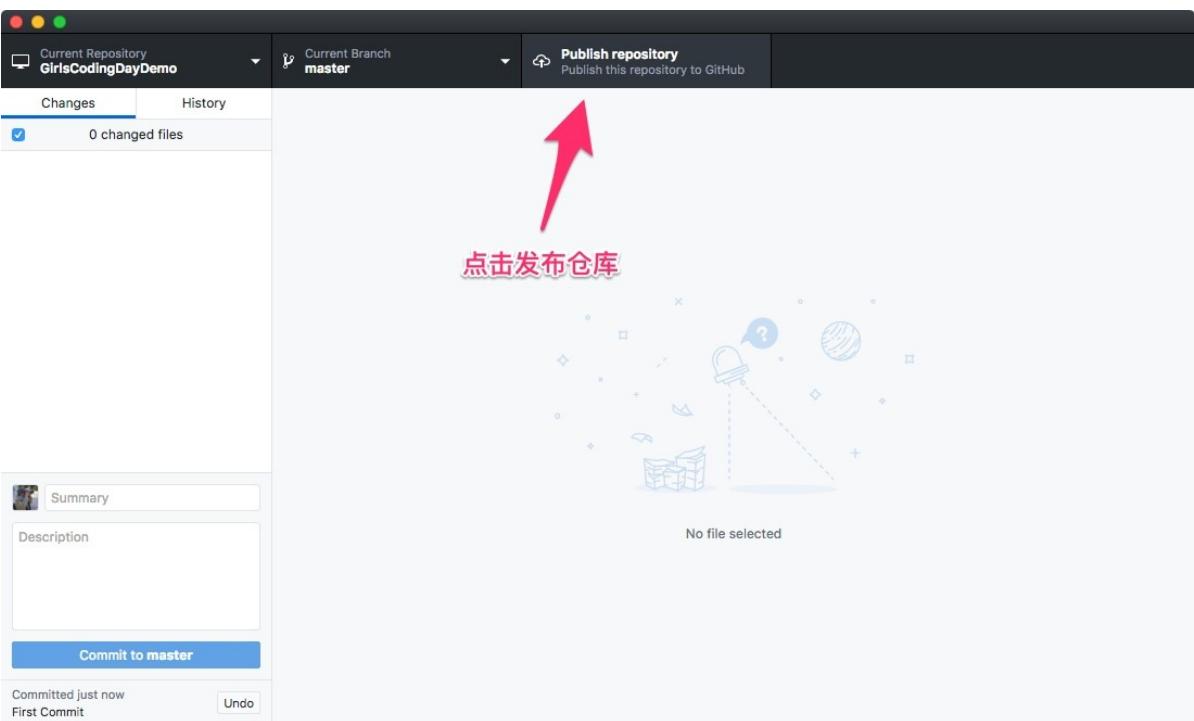


- a. 填写 Name 信息为 GirlsCodingDayDemo
 - b. local Path 选择你的桌面 (或者是你以后想要放代码的文件夹)
 - c. Initialize this repo with a Readme 的选项打钩
 - d. git ignore 保持none
 - e. license 保持none
 - f. 点击 create repository
3. 现在你的桌面上(或者是你想要放代码的文件夹中)会出现一个叫做 GirlsCodingDayDemo 的文件夹。解压下载的免费Bootstrap模板，把模板内的文件夹拷贝到 GirlsCodingDayDemo 这个文件夹中。
 4. 提交commit到master分支， (注意changes)

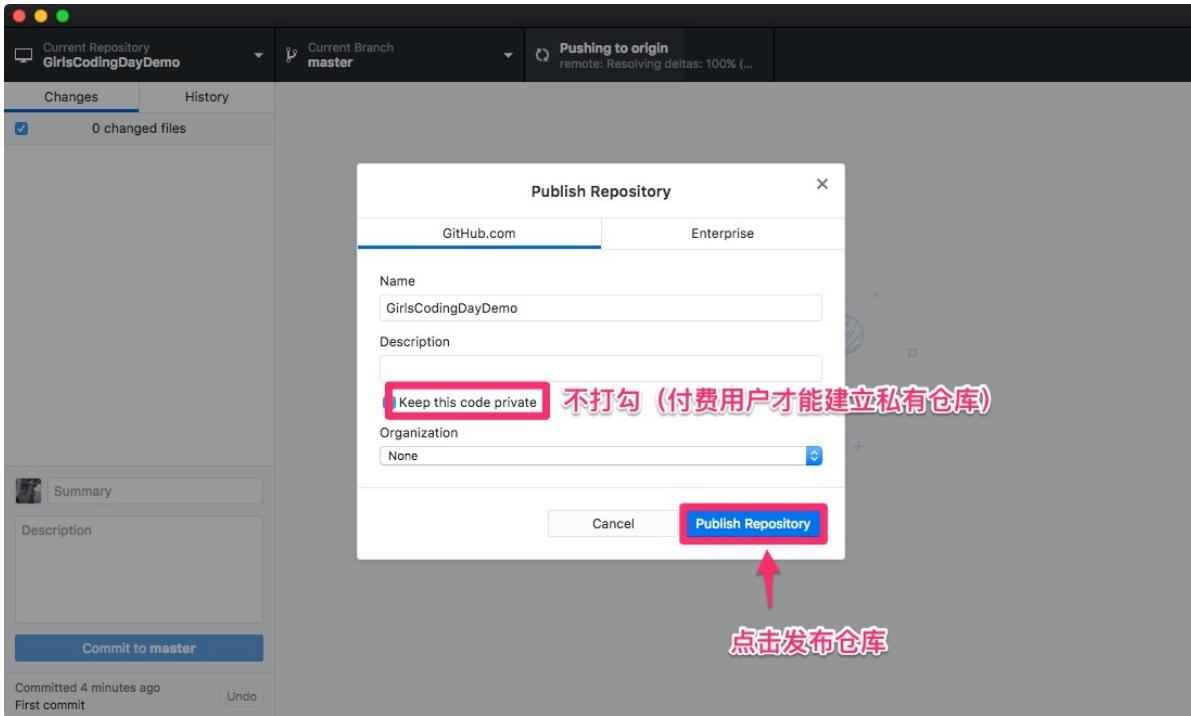
1.2 软件和工具



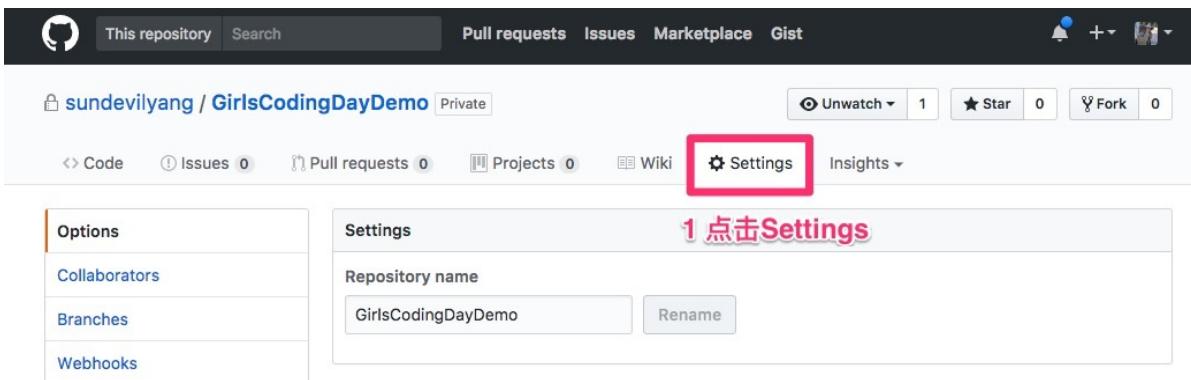
5. 发布到Github



1.2 软件和工具



6. 访问github.com 网站，你可以看到你刚刚通过Github Desktop软件在github.com 网站新建了一个仓库。点击Settings



设置这个仓库为Github Pages

The screenshot shows the GitHub Pages settings interface. At the top, it says "Your site is published at <http://sundevilyang.com/GirlsCodingDayDemo/>". Below that, it says "GitHub Pages is designed to host your personal, organization, or project pages from a GitHub repository." A red arrow points to this link with the text "4 点击此链接即可访问". In the "Source" section, there is a dropdown menu set to "master branch" and a "Save" button. A red arrow points to the "Save" button with the text "3 点击Save保存". Below this, a section titled "2 选择master分支" (2 Choose master branch) has a "Choose a theme" button. Further down, there is a "Custom domain" section with a "Save" button. At the bottom, there is a checkbox for "Enforce HTTPS" which is unchecked.

4 点击此链接即可访问

3 点击Save保存

2 选择master分支

Choose a theme

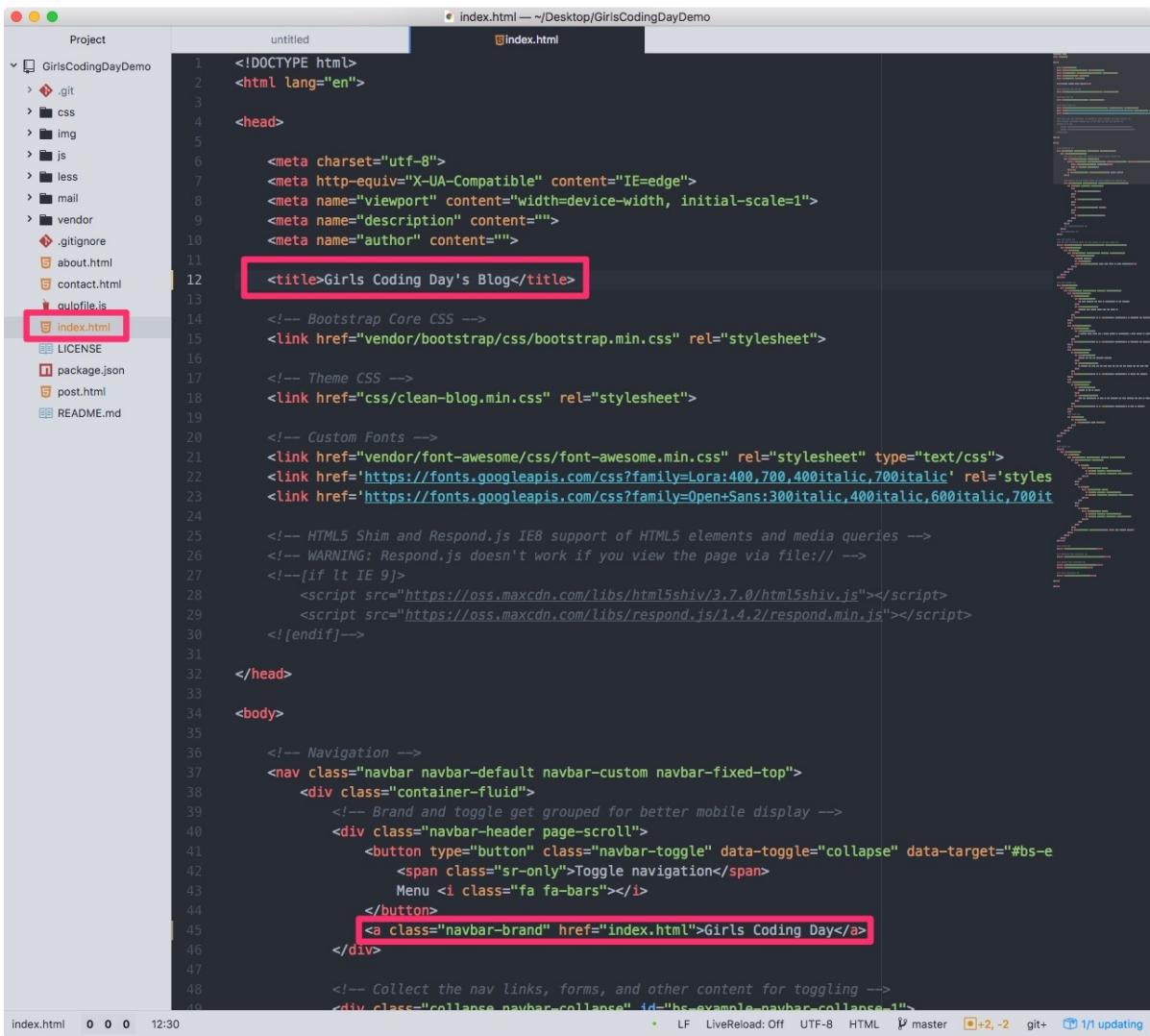
Custom domain

Save

Enforce HTTPS — Unavailable for your site because you have a custom domain configured ([sundevilyang.com](#))
HTTPS provides a layer of encryption that prevents others from snooping on or tampering with traffic to your site.
When HTTPS is enforced, your site will only be served over HTTPS. [Learn more](#).

7. 使用Atom编辑器打开桌面的文件夹 GirlsCodingDayDemo 。找到 index.html 文件，修改其中两处HTML代码, 然后保存（一定要保存!）。

1.2 软件和工具



```
<!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">
    <meta name="description" content="">
    <meta name="author" content="">
    <title>Girls Coding Day's Blog</title>
    <!-- Bootstrap Core CSS -->
    <link href="vendor/bootstrap/css/bootstrap.min.css" rel="stylesheet">
    <!-- Theme CSS -->
    <link href="css/clean-blog.min.css" rel="stylesheet">
    <!-- Custom Fonts -->
    <link href="vendor/font-awesome/css/font-awesome.min.css" rel="stylesheet" type="text/css">
    <link href="https://fonts.googleapis.com/css?family=Lora:400,700,400italic,700italic" rel="stylesheet">
    <link href="https://fonts.googleapis.com/css?family=Open+Sans:300italic,400italic,600italic,700it
    <!-- HTML5 Shim and Respond.js IE8 support of HTML5 elements and media queries -->
    <!-- WARNING: Respond.js doesn't work if you view the page via file:// -->
    <!--[if lt IE 9]>
        <script src="https://oss.maxcdn.com/libs/html5shiv/3.7.0/html5shiv.js"></script>
        <script src="https://oss.maxcdn.com/libs/respond.js/1.4.2/respond.min.js"></script>
    <![endif]-->
</head>
<body>
    <!-- Navigation -->
    <nav class="navbar navbar-default navbar-custom navbar-fixed-top">
        <div class="container-fluid">
            <!-- Brand and toggle get grouped for better mobile display -->
            <div class="navbar-header page-scroll">
                <button type="button" class="navbar-toggle" data-toggle="collapse" data-target="#bs-example-navbar-collapse-1">
                    <span class="sr-only">Toggle navigation</span>
                    Menu <i class="fa fa-bars"></i>
                </button>
                <a class="navbar-brand" href="index.html">Girls Coding Day</a>
            </div>
            <!-- Collect the nav links, forms, and other content for toggling -->
            <div class="collapse navbar-collapse" id="bs-example-navbar-collapse-1">

```

8. 使用chrome浏览器打开 index.html 预览修改效果



**Man must explore, and this is
exploration at its greatest**

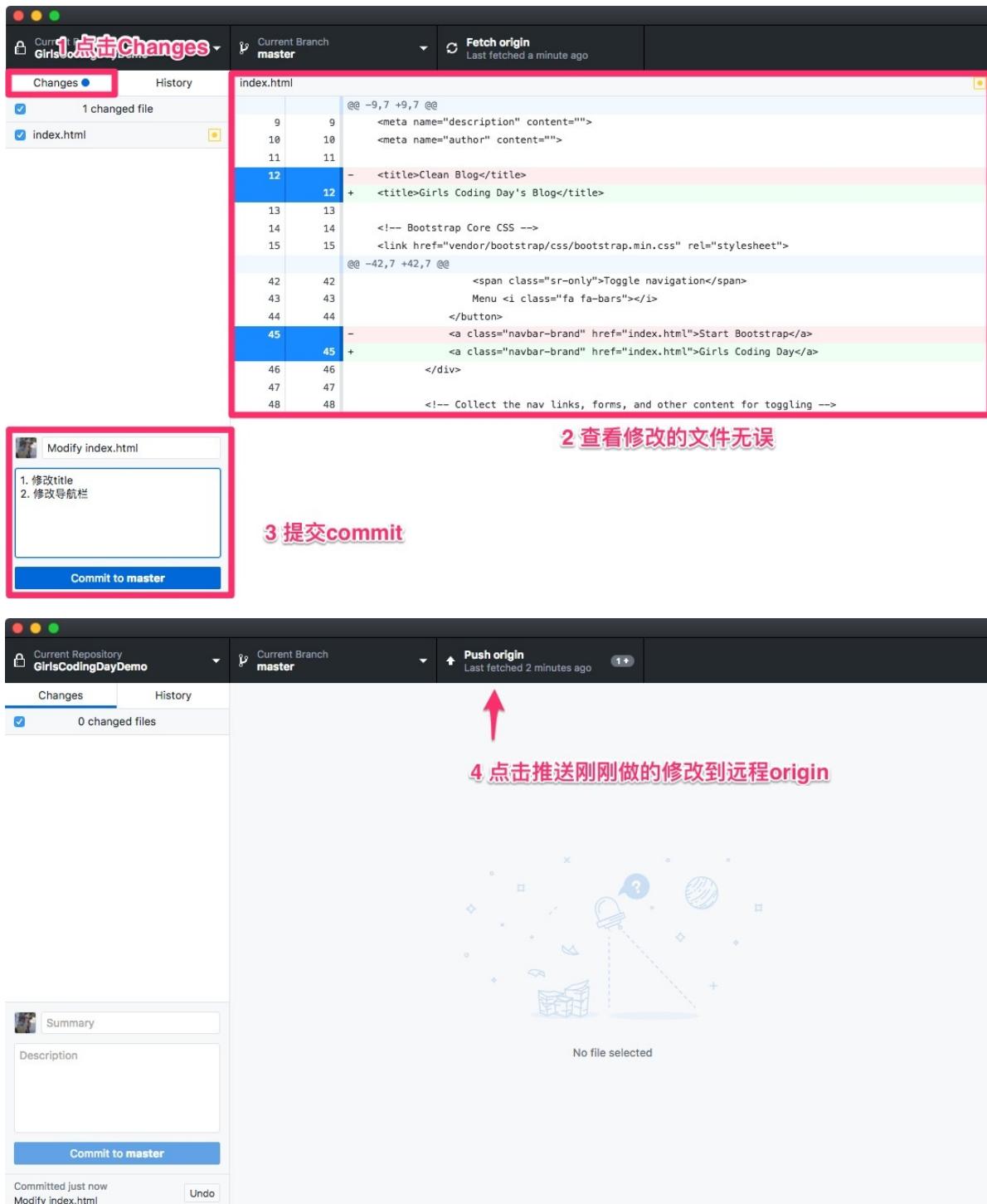
Problems look mighty small from 150 miles up

Posted by Start Bootstrap on September 24, 2014

**I believe every human has a finite
number of heartbeats. I don't intend to**

1.2 软件和工具

- 回到Github Desktop，可以看到修改的内容。提交commit，再点击Push origin同步到Github Pages。



- 打开github pages的链接，可以看到自己刚刚修改的内容。

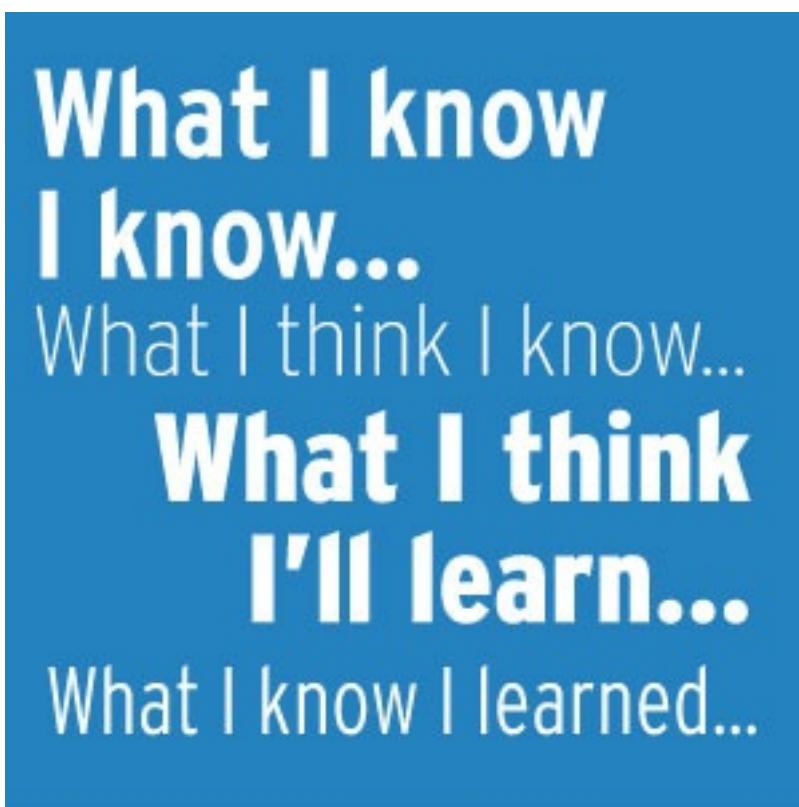
1.3 教学方法介绍

整个课程设计基于学习参与的三种方式：

1. 带着问题，探究式学习。
2. 自发动机，自主学习。
3. 指南不如匠人，社会化学习。

探究式学习

什么是定向思维？



定向思维，与发散思维相对，以基础为“经验”，在动力定型驱使下的按照既定方向或程序进行思维的活动过程。

但淋雨或、洗澡或开车时候在我们脑海中闪现的随机想法称作不定向思维，它不同于有目的的定向思维。

虽然，许多失败都缘于定向思维，但技能学习不同。在新进入技能领域之前，带着一个有经验前辈写下的关键问题，新手不容易迷茫并有能在浩瀚的概念中有定向前行。

- 带着问题流量使用工具、软件的网站（例如上面安装软件，我们并不是简单地让大家下载个软件而已，而是注入了需要思考的问题）
- 带着问题看阅读材料
- 带着问题做练习

1.3 教学方法

- 带着问题观察身边其他人

自主学习

如果你本身在互联网行业，恰巧身边有很多程序员。你会发现一个事实：『**程序员的编程技能都是自学成才！**』。不管是科班出身还是非科班出身，他们更多的人愿意将自己获得的编程技能途径定义为「自学」。

正如Alvin Toffler所说

21世纪的文盲不再是那些不会读书写字的人，而是那些不会学习，忘记所学，以及不能温故知新的人。

如果，你有这样的口头禅「告诉我，你想要我学什么，我就学给你看」，你可能不适合编程，因为我们希望你能「自己教自己」。

当你证明自己可以成为自主学习者，可以独立思考，可以找到和发现有效运用所学知识解决显示世界中难题的方法，你的自信心会得到极大的提升。自我获得力量增强的感觉，驱使自己为了未来而继续学习。

但要是失败了呢？失败是痛苦的，但是却也可以转化成一种有意义的学习经历。当你认识到做错了事情并知道下次怎样可以做得更好，你也获得了一种强有力终身学习的技能。

自主学习的八项技能

- 找出并评估有用的信息来源：信息源头不正确会导致一系列错误，例如我们阅读材料尽量只是W3School、维基百科等第一手文档，而不是其他二手加工过的资料。
- 在有用的信息源中辨别出重要的信息：设定阅读目标、找出中心思想和重要细节
- 以有意义的方式组织信息：用Cmap画概念图
- 撰写日志和博文：写作输出是最好的知识管理
- 安排好时间：第一要义是**时间哲学**，可参考文章 [「感觉身体被掏空」到掌控世界 -- 上班族一周如何拿出15小时学习](#)
- 背诵最小核心知识：把经过历史和经验反复验证的知识概念深入到自主心智。如果一个成年人还跟小学一年级的学生一样，需要去理解为什么三加四等于七，而不是把这个概念作为本能反应出来，那么他是无法和别人竞争的，因为效能太低。而我们为了能够提高学习效率，必须在学习过程中记住一些关键知识。
- 运用问题解决体系
- 监督自己的学习（元认知）

社会化学习

教是最高的形式的理解。

1.3 教学方法

告诉我，我会忘记。给我看，我会记住。让我教，我会理解。

如果仅仅依靠自己的行动、通过试错来学习，那么学习绝对是一件非常辛苦、费力的事情，而且可能是危险或有害的。

回归学习的本质，人类大多数行为是通过模仿和观察习得的，通过交流、分享、应用、反馈、争论、甚至争辩，最终帮助个体对于某个事物形成较为全面的认知。

因此我们非常有必要，大家坐在一个屋子里，共同完成一些学习。哪怕没有言语交流，只是彼此坐在一起，也是非常棒的“交流”。

三个提高思维能力的方法

- 概念图
- 刻意练习
- 费曼技巧

1.4 WEB基本知识和历史

Cooking time: 30min active / 30-45min passive

Learning by Reading(10mins): [网络是如何工作的 - 学习 Web 开发 | MDN](#)

本小节会覆盖的主题

- 托管主机: 计算机上的文件如何成为公共网站
- 主机的组件和网站架构
- 各种可选主机提供商以及如何选择

本地开发环境

- 打开Atom软件, 在桌面创建文件 `index.html`, 并用文本编辑器写入你最喜欢的诗篇。
- 现在, `index.html` 只是一个保存在你自己电脑上面的文件.
- 你可以通过文件夹 (Windows 电脑) 或Finder (Mac电脑) 看到文件的地址类似为
`/Users/yangbwen/Desktop/index.html`

主机的作用是给这个新创立的文件一个地方, 好让互联网上的任何人都可以看见。

这个地方可以是你的电脑吗?

默认设置互联网上随随便便一个人是不能访问你电脑的, 但你可以给他们权限。

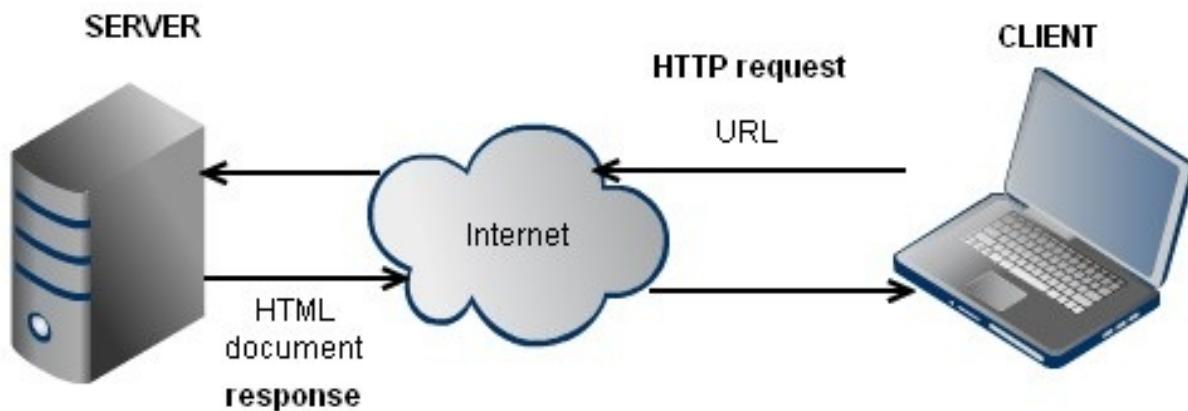
- 例如: 你可以在你的电脑上面设置一个互联网服务器 (webserver) 。
- 缺点: 如果你的电脑关机了别人就不能访问了。
- 通常做法: 把你的这些文件放在一个一直开机的地方。

服务器 (Server) 与主机 (Hosting)

托管主机: 把你文件的地方通过托管在一台由其他人管理电脑的服务器 (server) 上。

1. 阅读 [服务器 - 维基百科, 自由的百科全书](#)
2. 阅读后回答: 服务器是软件还是硬件?

认识名词: 单机、Client/Server、Browser/Server



客户端提出请求，服务器端满足客户端（通常）。请求就像问题：客户端询问他们，服务器回答。对于我们的web讨论，client = browser。虽然任何类型的计算机可以用作服务器，但服务器通常比其他计算机更大和更强大。客户端应用程序在与您交互的客户端计算机上运行（如您的浏览器），服务器应用程序在其他地方的机器上运行，这样一来客户端或客户端应用程序就可以和服务器应用程序通话

1. 使用搜索引擎搜索上述关键词「Client/Server、Browser/Server」，阅读自己搜索到的文章。
2. 教练带大家讨论C/S 和 B/S。
3. 问题：怎么去判断是B/S 还是C/S？例如小程序是哪种模式？微信手机端是哪种模式？微信网页端是哪种模式？

托管客户端应用程序

您的 `index.html` 和 `style.css` 文件都是客户端代码（client code），浏览器运行这些文件就会显示页面。为了使任何人的浏览器都可以加载它们，我们将它们放在其他地方（主机）的计算机上，该计算机上的服务器（server）将是看门人，让任何人的浏览器加载您的文件（客户端代码）。在这个例子中，你的代码是客户端代码，服务器只是作为看门人。

托管主机提供商

类别	说明 (what)	什么情况下用 (when)	比较 (why)
文件存储	放置文件的地方;示例： S3, Github Pages, Dropbox, 百度盘, 七牛。	当你只需要静态文件即可。	适用于静态页面或客户端应用程序，不足以用于服务器应用
软件	可以获取域名，可选择预	当你需要一个廉价方法来做网站（例如，为一个小企业），并	想要创建和管理内容的用户，对于想

包服务	wordpress.org, bluehost.com。	要一切“能工作而已”，并不需要定制，那么你可以使用此类型。	人员或者想要使用软件组件不在允许列表上
自建	可以设置和管理您自己的所有网站组件，包括数据库，服务器，域名等。例如：Amazon EC2, Heroku, Linode, Digital Ocean, 阿里云，青云等。	当你需要完全控制你的软件或业务要求时；例如，大多数科技公司都是建立自己的网站架构和托管解决方案。	更耗时，往往更便宜。需要技术人员来实现和维护，但完全可控，部署/托管任何类型的客户端或服务器应用程序的能力

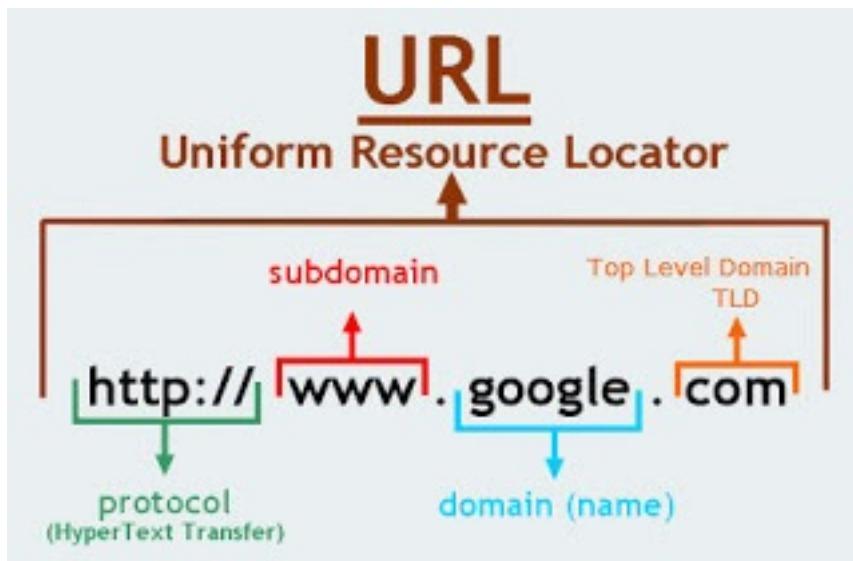
域名托管 Domain hosting

- IP addresses

IP被在线电脑用来互相识别，通常是四组数字。通过简单的方法查找您的IP地址 [What Is My IP Address?](#)

阅读 [IP地址 - 维基百科](#), 自由的百科全书

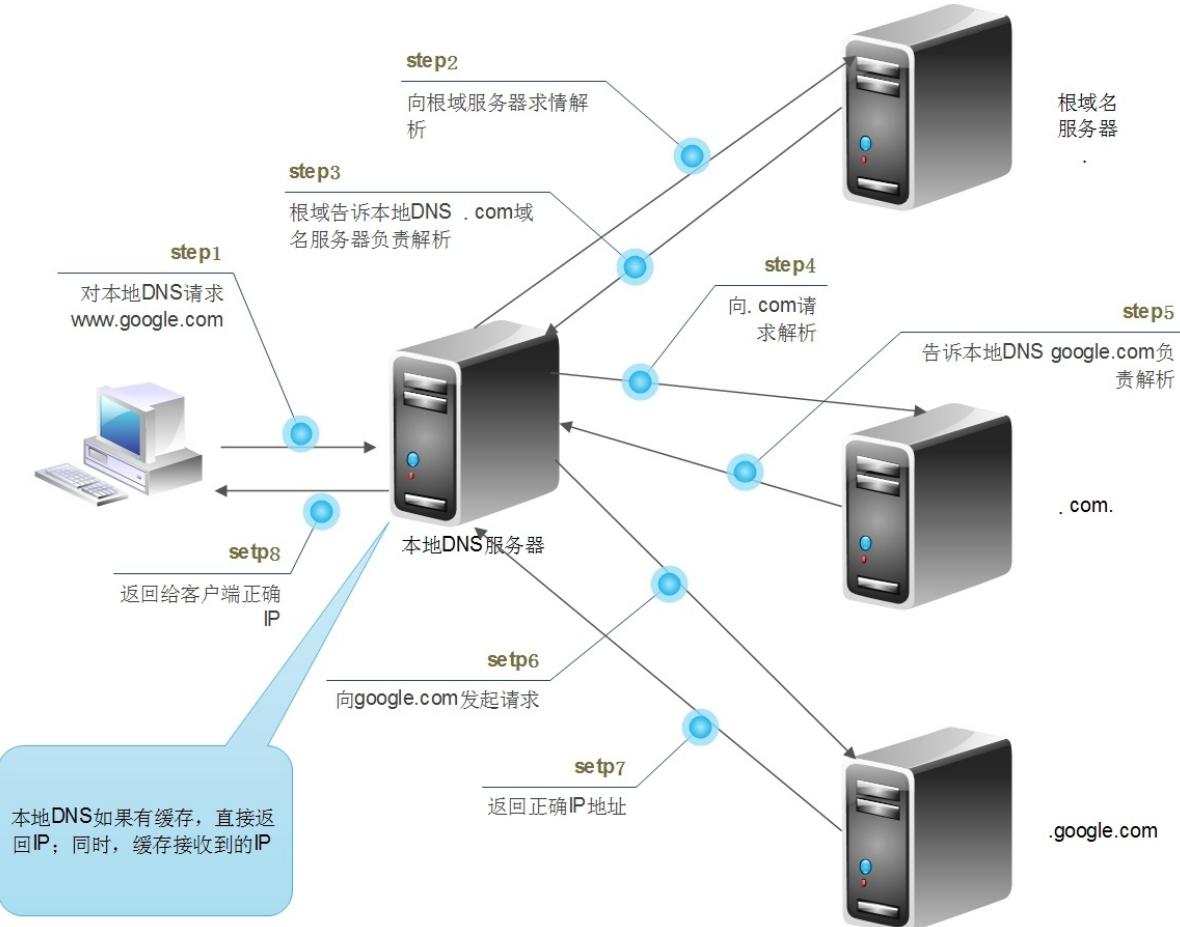
- Domains



- Domain Name System (DNS)

将域名转换为允许计算机相互通信的IP地址。通过键入终端（Mac）或命令提示符（PC）查找GCD的IP地址：nslookup girlscodingday.org

Advice for Tutors: 教练解释下图



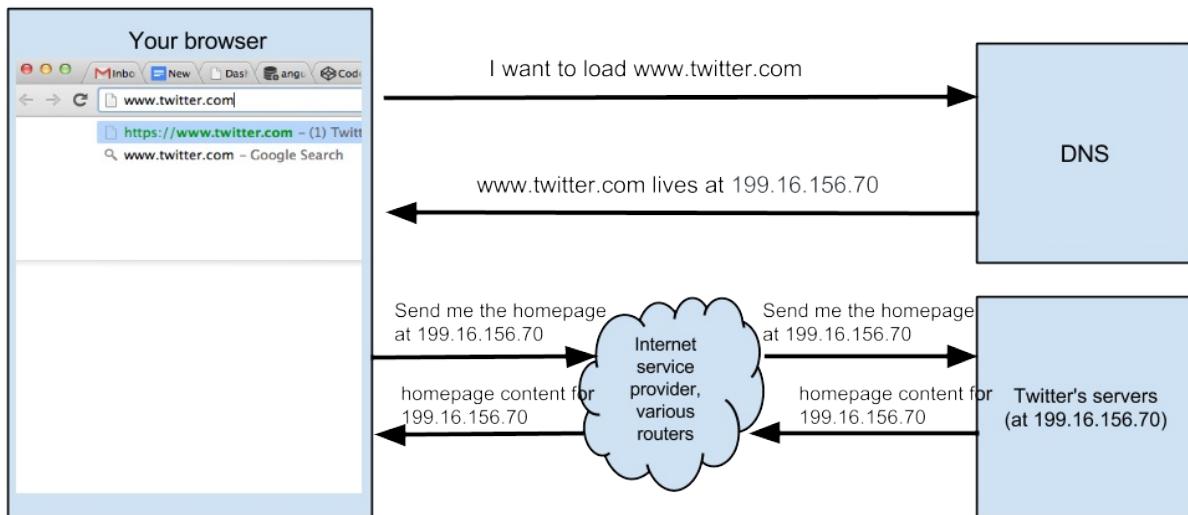
• 注册域名 Registrars

通常用来购买域名。良好的客服、易于使用的管理界面、价格实惠（无隐藏费）、允许将名称转让给其他注册商、ICANN认证

Advice for Tutors: 教练说说国内、国外都有哪些买域名的代理机构？国外和国内买域名有哪些优缺点？什么是备案（可选）？

一个网站的日常

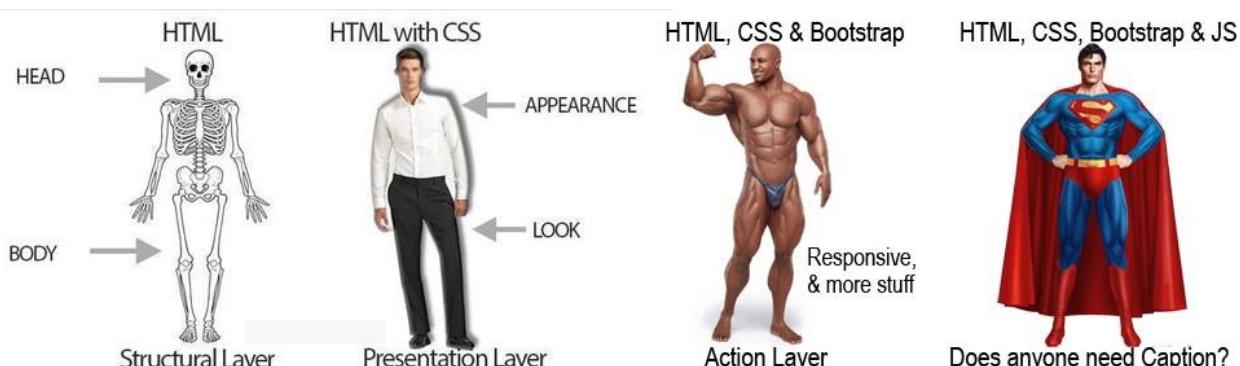
1. 在浏览器地址栏中输入网址
2. DNS 将您连接到托管服务器
3. 然后将文件发送回您的计算机进行显示
4. 有时，代码在发给你之前必须进行编译。



HTML/CSS/Bootstrap/Javascript 对于WEB分别意味着什么？



Advice for Tutors: 教练简单讲讲 HTML/CSS/JavaScript 分别是什么以及三者之间的关系。



1.4 WEB基本知识和历史

Advice for Tutors: 教练讲讲

- 什么是Bootstrap?
- Web 1.0 vs Web 2.0 ?
- 浏览器大战

Tasks For Tutors:

1. 给学员演示如何使用atom
 - 打开一个项目
 - 找到index.html
 - 修改html常用标签: title/h1/h3/li/ol
2. 给学员演示Chome Developer Tools的简单用法
 - 快捷键调用开发者工具
 - 怎么看定位页面上某个HTML元素
 - 怎么查看某个HTML元素的CSS样式, 以及在开发者工具中调试样式属性

1.5 课前作业

- 从你的同组学员中，找一个你喜欢的或许也喜欢你的小伙伴，结对！
 - 阅读下列资料
 - 阅读[HTML 教程](#) 到 「HTML 内联框架」章节
 - 阅读[CSS 教程](#) 到 「CSS 伪元素」章节
- 手机上有 W3C School 的 App，大家可自行搜索下载。充分利用上下班通勤时间学习。

Tasks For Tutors:

1. 每天（根据各组情况也可以两天一次，我们鼓励每天都联系）晚上找一个合适的时间，使用软件 [碰碰](#) 或者其他软件跟大家视频开[站会](#)，了解学员的学习情况，鼓励她们积极学习。
 - 站会一般不超过15分钟；
 - 站会经常问的三个问题（更多信息 [Daily Scrum - Scrum Inc](#)）：
 - What did I do yesterday that helped the Team meet the Goal?
 - What will I do today to help the Team meet the Goal?
 - Do I see any impediment that prevents me or the Team from meeting the Goal?
2. 如果已经完成上述材料的阅读，鼓励学员完成本教程后续内容。

Intro to HTML

本章节目标

1. 了解一个网站的结构
2. 能阅读和理解所给源码中的HTML
3. 学会使用文本编辑器Atom阅读、修改HTML源码

初识HTML

Inquiry Learning

尝试向你的同学解释下列概念：

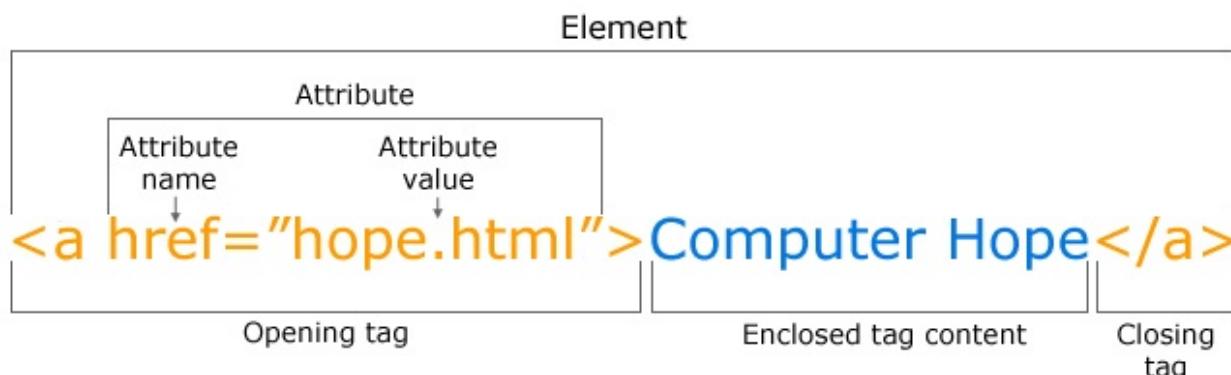
1. 什么是文本 (Text) ?
2. 什么是超文本 (HyperText) ?
3. 什么是标记语言 (Markup Language) ?
4. 什么是网页 (也叫HTML 文档) ?

Learning by Reading

阅读 [W3C School HTML 教程](#) 到「HTML 内联框架」章节

HTML常用术语

Breakdown of an HTML Tag



元素 - Element

Inquiry Learning

- 什么是HTML元素(HTML Element)?
- HTML文档和HTML元素什么关系?

标签 - Tag

Inquiry Learning

什么是HTML标签 (HTML Tag) ?

属性 - Attribute

Inquiry Learning

- 什么是属性?
- 标签和属性是什么关系?
- 能否识别属性的name/value形式?

两个重要属性

- class

Learning by Reading

HTML class 属性 & HTML 类

- id

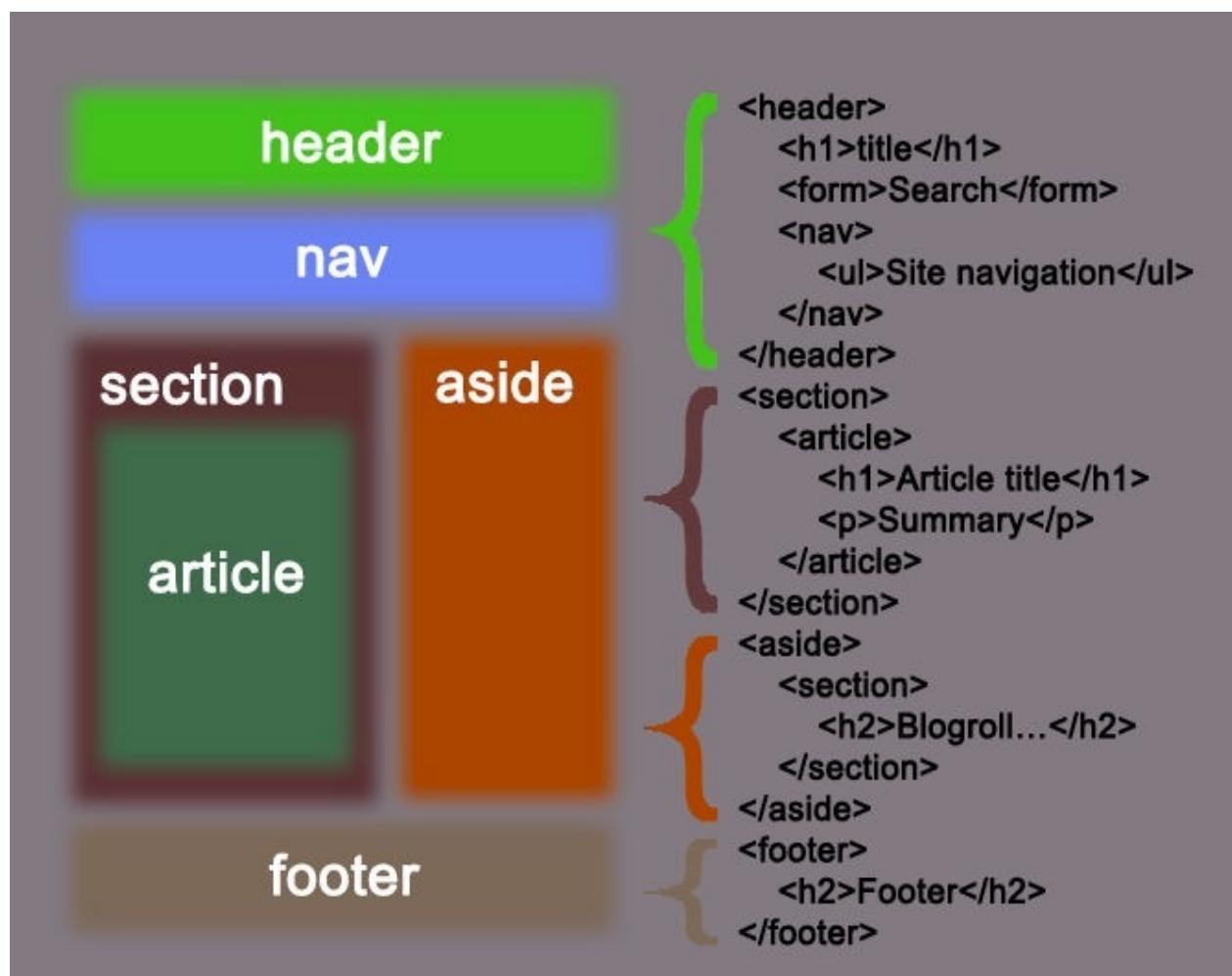
2.2 HTML常用术语

Learning by Reading

HTML id 属性

解剖网站

HTML5 Structure



Inquiry Learning

尝试向你的同学解释下列概念：

`<header>` vs. `<head>` vs. `<h1>` 三者的区别?

Advice for Tutor: 向同学们解释下图，使学员脑海中网页结构的画面与HTML元素一一对应起来

2.3 解剖网站

The screenshot shows a web browser displaying the 'Clean Blog' theme by Start Bootstrap. The page includes a header with a background image of a desk setup, a navigation bar with links like HOME, ABOUT, SAMPLE POST, and CONTACT, and a main content area featuring blog posts. Annotations with question marks and arrows point to specific HTML elements: 'Page Title' points to the title bar, 'Website URL' points to the address bar, 'Header' points to the top section, 'Background Image' points to the image in the header, 'Top Navigation' points to the navigation bar, 'li tag' points to a menu item, 'h1 tag' points to the main heading, and 'span tag' points to a span element in the content.

Content

Man must explore, and this is exploration at its greatest

Problems look mighty small from 150 miles up

Posted by Start Bootstrap on September 24, 2014

I believe every human has a finite number of heartbeats. I don't intend to waste any of mine.

Posted by Start Bootstrap on September 18, 2014

Science has not yet mastered prophecy

Learning by Doing

1. 找一个你喜欢的网站，调出Chrome开发者工具，向跟你结对的同学互相解剖此网站结构。
2. 打开网站 <https://codepen.io/pen/>，在HTML编辑框建立自己的第一个HTML文档，并请教练审阅。
 - 所有HTML文档都必须符合特定的结构，要求至少包含以下声明和元素：`<DOCTYPE html>`，`<html>`，`<head>` 和 `<body>`。
 - 此练习必须包含以下HTML标签及其属性。
 - `h1`
 - `p`
 - `li`
 - `body`
 - `meta` (`charset`属性为`utf-8`)
 - `title`
 - `head`
 - `DOCTYPE`
 - `html`

Advice for Tutors:

1. 聊聊HTML的元素嵌套规则。

讲解如何使用Chrome开发者工具定位HTML元素，查看HTML元素的嵌套实例。

2. 聊聊 自闭合元素 (self-closing elements) 。

例如上面第二个练习题 就只有开标签，没有闭标签。其他常见自闭合标签：`
`

```
<embed> <hr> <img> <input> <link> <meta> <param> <source> <wbr>
```

3. 聊聊三种标签元素的区别与特点：

块元素

```
div, ul, li, dl, dt, dd, p, h1-h6, blockquote
```

块级元素的特点：

- 总在新行上开始，占据一整行
- 默认情况下，其宽度自动填满其父元素宽度
- 宽度始终是与浏览器宽度一样，与内容无关
- 它可以容纳内联元素和其他块元素
- `display`属性为`block`
- 块级元素的垂直相邻外边距`margin`会合并。

内联元素

```
a, b, span, img, input, strong, select, label, em, button, textarea
```

内联元素的特点：

- 和其他元素都在一行上
- 高，行高及外边距和内边距部分可改变
- 宽度只与内容有关
- 行内元素只能容纳文本或者其他行内元素
- `display`属性为`inline`
- 水平方向的`padding-left`、`padding-right`、`margin-left`、`margin-right`都产生边距效果，但竖直方向的`padding-top`、`padding-bottom`、`margin-top`、`margin-bottom`却不会产生边距效果。
- 不可以设置宽高，其宽度随着内容增加，高度随字体大小而改变，内联元素可以设置外边界，但是外边界不对上下起作用，只能对左右起作用。

空元素

2.3 解剖网站

```
br,meta,hr,link,input,img
```

空元素的特点：

- 没有内容的 HTML 内容被称为空元素。空元素是在开始标签中关闭的。

动手 - 修改更多HTML元素



1. 把导航栏的英文换成中文，并添加一列「报名 Girls Coding Day」，链接到 <https://srcodingday.org/> (考察知识点：考察学员 Lists, 参考 [HTML 列表](#))
2. 修改 `<header>` 元素中的内容 (考察知识点：Text-Based Element, 参考 [HTML 文本格式化](#))
 - 修改 `<h1>` 元素中内容为「Girls Coding Day」
 - 增加 `<h3>` 标签，其内容为「我也愿意贡献力量帮助更多的女性学习编程」 (考察知识点：Headings系列)
 - 在 `` 标签嵌套一个 `<a>` 标签 (链接为 Bootstrap模板下载地址 <https://startbootstrap.com/template-overviews/clean-blog/>) (考察知识点：Paragraphs, Opening Links in a New Window, 参考资料 [HTML 链接](#))
 - 将「博客」二字加粗 (考察知识点：Bold Text with Strong, 参考资料 [HTML 文本格式化](#))
 - 将「制作」二字变成斜体 (考察知识点：Italicize Text with Emphasis, 参考资料 [HTML 文本格式化](#))

Inquiry Learning :

- i. 什么是块元素?
- ii. 什么是内联元素?
- iii. Div元素和span元素哪个是块元素? 哪个是内联元素?

Learning by Reading : 阅读 [HTML `<div>`](#) 和 [](#) , 理解块元素和内联元素

2.4 动手 - 修改更多HTML

- 去 <https://unsplash.com/> 下载图片到 img 文件夹，修改图片名字，替换掉背景图
(考察知识点：Image，参考资料 [HTML 图像](#))

3. 修改页面最下端 `<footer>` 元素中的内容如下图所示



Copyright © Girls Coding Day 2017

- 修改twitter图标为微信图标
- facebook图标为微博图标，如果有微博请链接到自己的微博

Advice for tutor

- 什么是 Font Awesome (参考：[Font Awesome 中文网](#)) ?
- 怎么使用 Font Awesome ?
- 添加邮箱图标，点击图标可以发送邮件。邮件需要包含主题 (subject)和正文 (body) (参考：[Mailto Links | CSS-Tricks](#)) (备注：Linking to an Email Address)
- 修改copyright段落内容

Intro to CSS

本章节目标

1. 引用样式CSS
2. 了解CSS基本知识练习其基本用法
3. 使用编辑器Atom重置CSS样式

初识CSS

Inquiry Learning

尝试向你的同学解释下列概念：

1. 什么是层叠样式表（Cascading Style Sheets）？
2. CSS 规则的两个主要构成部分是什么？
3. HTML和CSS什么关系？

Learning by Reading

阅读[CSS 教程](#)到「CSS 伪元素」章节

引用 CSS

1. HTML 外部样式表

为了让CSS效果能在HTML中显示出来，我们需要在HTML文档中引用CSS文件。引用CSS最好的办法是把所有的样式代码放在额外的一个文件中，在HTML文档中的元素中引用此文件。

Inquiry Learning：想一想这样引用CSS的好处？

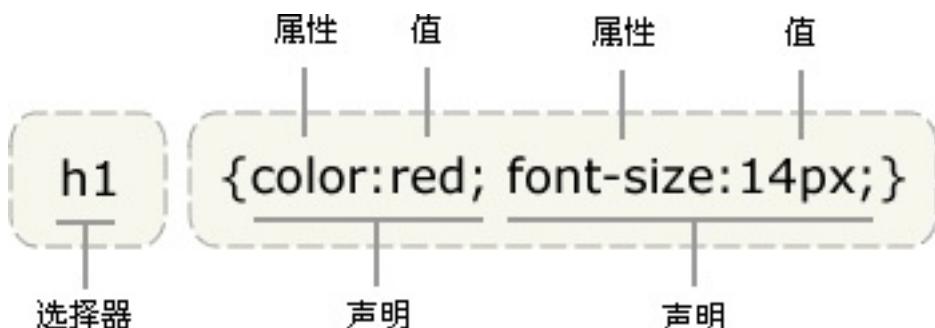
2. 另外两种引用CSS的方法：

- 内部样式表
- 内联样式

Inquiry Learning：当同一个HTML元素被不止被以上三种方法定义时，会使用哪个样式呢？

Learning by Reading：阅读[HTML CSS & CSS 简介 & 如何创建 CSS](#)

CSS常用术语



选择器 - Selector

Inquiry Learning

1. 选择器跟HTML元素什么关系?
2. 如何把相同的声明赋给多个HTML元素? 提示: 选择器分组
3. 子元素会继承父元素的样式, 子元素不想继承, 该怎么办呢?
4. 什么是「派生选择器、*id* 选择器、类选择器、属性选择器、伪类」?
5. 类选择器 可以在单个HTML文档中出现多次, *id*选择器 也可是在单个HTML文档中出现多次吗?
6. *id*选择器 和 派生选择器 可以混合使用吗?

声明 - Declaration

Inquiry Learning

1. 属性和值用哪个英文标点符合隔开呢?
2. 一个属性可以包含多个值吗?
3. 如果要定义不止一个声明, 需要用哪个英文标点符号将每个声明分开?
4. 属性和值一定要英文小写吗?

属性 - Property

大概了解各个属性的效果, 不要求熟练掌握。

Inquiry Learning

- 你知道下列CSS背景属性都有什么样式效果吗?

属性	描述
background	简写属性, 作用是将背景属性设置在一个声明中。
background-attachment	背景图像是否固定或者随着页面的其余部分滚动。

3.2 CSS常用术语

background-color	设置元素的背景颜色。
background-image	把图像设置为背景。
background-position	设置背景图像的起始位置。
background-repeat	设置背景图像是否及如何重复。

- 你知道下列CSS文本属性的效果吗？

属性	描述
color	设置文本颜色
direction	设置文本方向。
line-height	设置行高。
letter-spacing	设置字符间距。
text-align	对齐元素中的文本。
text-decoration	向文本添加修饰。
text-indent	缩进元素中文本的首行。

- 你知道下列CSS字体属性的效果吗？

属性	描述
font	简写属性。作用是把所有针对字体的属性设置在一个声明中。
font-family	设置字体系列。
font-size	设置字体的尺寸。
font-size-adjust	当首选字体不可用时，对替换字体进行智能缩放。（CSS2.1 已删除该属性。）
font-stretch	对字体进行水平拉伸。（CSS2.1 已删除该属性。）
font-style	设置字体风格。
font-variant	以小型大写字体或者正常字体显示文本。
font-weight	设置字体的粗细。

- 你知道如何设置链接样式吗？
- 你知道下列CSS列表属性的效果吗？

属性	描述
	简写属性。用于把所有用于列表的属性设置于一个声明中。

3.2 CSS常用术语

	简写属性。用于把所有用于列表的属性设置于一个声明中。
list-style-image	将图象设置为列表项标志。
list-style-position	设置列表中列表项标志的位置。
list-style-type	设置列表项标志的类型。

- 你知道下列CSS表格属性的效果吗？

属性	描述
border-collapse	设置是否把表格边框合并为单一的边框。
border-spacing	设置分隔单元格边框的距离。
caption-side	设置表格标题的位置。
empty-cells	设置是否显示表格中的空单元格。
table-layout	设置显示单元、行和列的算法。

值 - Value

Inquiry Learning

值为若干单词，则要给值加引号。用双引号还是单引号呢？

Learning by Reading

其他常用术语 [CSS Terms and Definitions](#)

盒子模型 - Box Model

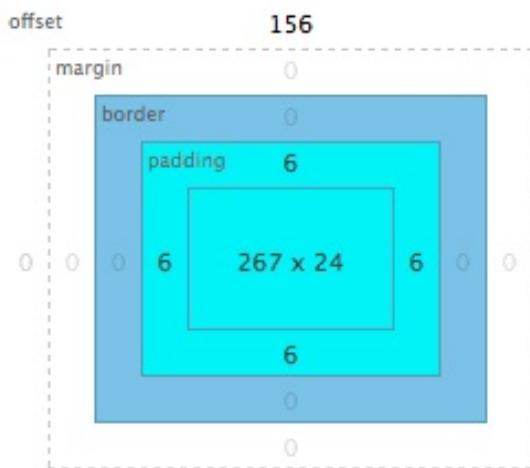
我们已经具备一些基本的HTML和CSS知识，知道一个网页大概长什么样子。现在我们深入讨论盒子模型，看看HTML元素到底如何在浏览器中展示的。

根据[CSS 框模型概述](#)，网页面中所有的元素都是矩形的盒子（rectangular box），有宽度（width）、高度（height）、内边距（padding）、边框（border）、外边距（margin）。

重要的事情说三遍：网页中的每个元素都是矩形的盒子。网页中的每个元素都是矩形的盒子。网页中的每个元素都是矩形的盒子。

Advice for tutor

1. 请教练以下图和代码为例，讲讲盒子的大小



*Firebug's Box Model Display
(not to scale, but very useful)*

```
div {
    width: 267px;
    height: 24px;
    padding: 6px;
    border: 0px;
    margin: 0px;
}
```

Margin 比较特别，它不会影响盒子本身的大小，但是它会影响和盒子有关的其他内容，因此 margin 是盒模型的一个重要的组成部分。

盒子本身的大小是这样计算的：

盒子大小	计算方法
Width	<code>width + padding-left + padding-right + border-left + border-right</code>

3.3 CSS盒子模型

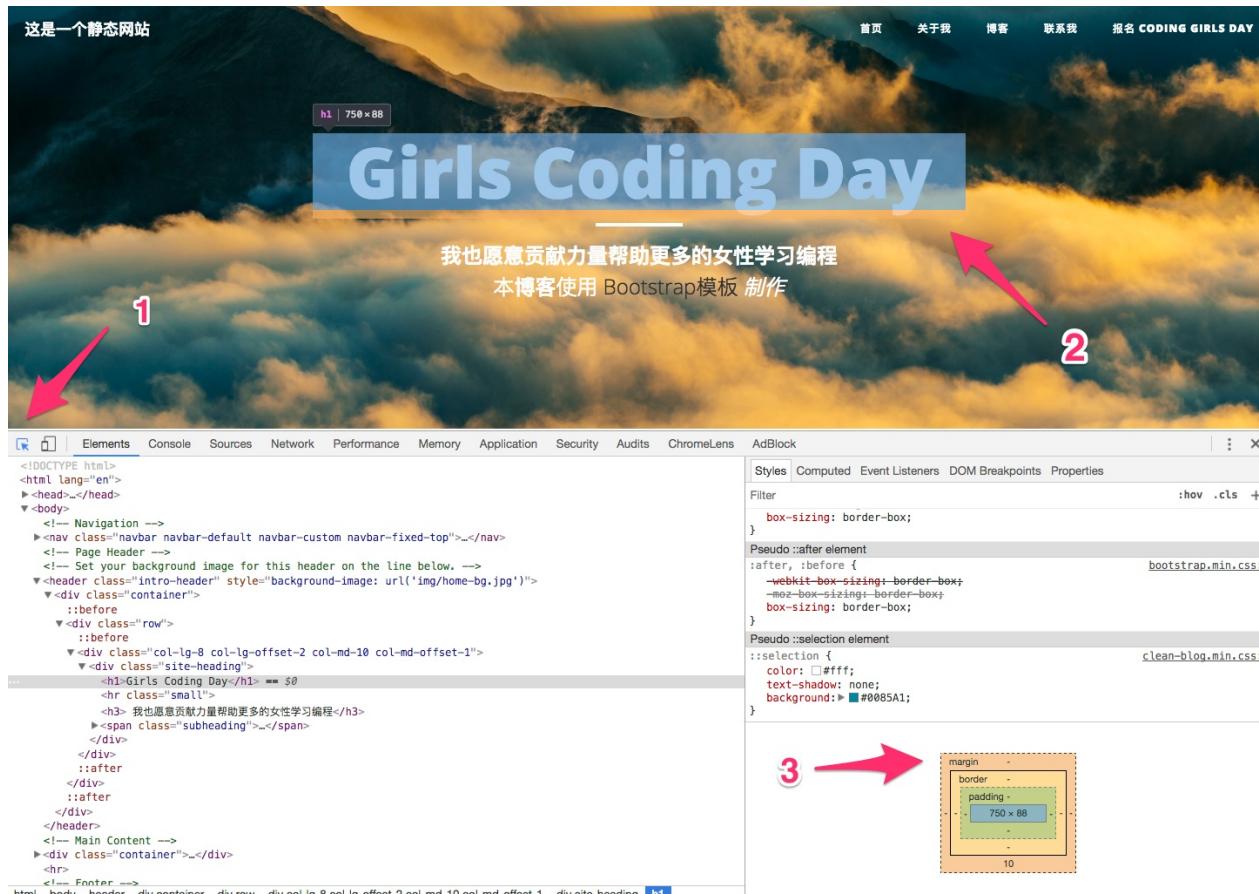
Height

height + padding-top + padding-bottom + border-top + border-bottom

使用Chrome开发者工具调试样式

Advice for tutor

请教练使用Chrome开发者工具，简单演示如何调试盒子模型属性以及其他样式。



动手 - 修改更多CSS样式

- 在 css 文件夹中添加一个新文件 `custom.css` , 并在 `index.html` 文件中的元素引用, 如下图

```

17      <!-- Theme CSS -->
18      <link href="css/clean-blog.min.css" rel="stylesheet">
19
20      <!-- Custom CSS -->
21      <link href="css/custom.css" rel="stylesheet"> |
22

```

- 使用类选择器、*id*选择器、派生选择器给 `index.html` 中的元素添加样式。

观察 `index.html` 中的文章预览元素(仅截取 `<div class="post-preview">` 标签中的内容):

```

<div class="post-preview">
    <a href="post.html">
        <h2 class="post-title">
            Man must explore, and this is exploration at its greatest
        </h2>
        <h3 class="post-subtitle">
            Problems look mighty small from 150 miles up
        </h3>
    </a>
    <p class="post-meta">Posted by <a href="#">Start Bootstrap</a> on September 24, 2014</p>
</div>

```

- 把 `<div class="post-preview">` 改为 `<div class="container" id="post">`
- 请在 `custom.css` 中添加下列内容并保存, 然后我们来看看CSS中如何使用选择器 (注意阅读注释内容) :

```

/*这里的“*”是通配符, 意思是该页面中所有的元素都会被加上*后面大括号里的样式,
对应图中页面中所有的标签都有红色的上边框*/
*{
    border:1px solid red;
}

/*类选择器: 这里使用.post-title的方式选中所有具有类名为post-title的元素, 这里的样式
使他们大小一样, 都有绿色的边框*/
.post-title{
    background-color: #777;
    border: 10px solid green;
}

/*id选择器: 这里使用#post的方式选中具有id属性值为post的元素, 添加样式, 改变它的背景色
为白色, 边框为绿色点线*/
#post{

```

3.5 动手 - 修改更多CSS样式

```
border: 5px dotted green;
background-color: #FFF;
}

/*派生选择器：这里的空格代表元素层级关系，选中了class属性值为“post-meta”的元素包含链接的元素，使用派生选择器会使得你的css代码层次更清晰，更有可读性*/
.post-meta a {
    background-color: blue
}
```

3. 自己练习更多CSS属性

什么是Bootstrap?

1. Bootstrap 是由Twitter推出的一个用于快速开发 Web 应用程序和网站的前端框架。
2. 基于html5、css3的bootstrap，具有下面这些诱人特性：
 - 移动设备优先；
 - 漂亮的设计；
 - 友好的学习曲线；
 - 卓越的兼容性；
 - 响应式设计；
 - 12列响应式栅格结构；
 - 样式向导文档。
3. 自定义JQuery插件，完整的类库，基于Less、Sass等

安装Bootstrap

Bootstrap 提供以下几种方式帮你快速上手，每一种方式针对具有不同技能等级的开发者和不同的使用场景。我们使用方法一，减少操作。

1. 使用 Bootstrap 提供的免费 CDN 加速服务

英文官方网站和中文网站的地址不同，由于众所周知的墙的原因，我们采用中文网站）。

复制下面的

```
<!-- 新 Bootstrap 核心 CSS 文件 -->
<link rel="stylesheet" href="http://cdn.bootcss.com/bootstrap/3.3.0/css/bootstrap.min.css">
```

把这个JavaScript插件以及JQuery放在你的网页的末尾附近，就在</body>标签前面。记住需要先添加jQuery，因为我们的代码依赖于它。

```
<!-- jQuery文件。务必在bootstrap.min.js 之前引入 -->
<script src="http://cdn.bootcss.com/jquery/1.11.1/jquery.min.js"></script>

<!-- 最新的 Bootstrap 核心 JavaScript 文件 -->
<script src="http://cdn.bootcss.com/bootstrap/3.3.0/js/bootstrap.min.js"></script>
```

2. 下载「用于生产环境的 Bootstrap」到本地

到 [Bootstrap中文网站](#)下载用于生产环境的 Bootstrap，然后解压缩。

3. 其他工具下载

现在还无需掌握，知道有这种安装方法即可

- 通过 Bower 进行安装
- 通过 npm 进行安装
- 其他

起始模板

See the Pen [Bootstrap标准模板](#) by Yang Wen (@sundevilyang) on CodePen.

全局 CSS 样式概览

设置全局 CSS 样式；基本的 HTML 元素均可以通过 class 设置样式并得到增强效果；还有先进的栅格系统。

一边参考[全局 CSS 样式 · Bootstrap 中文文档 概览部分](#)，一边阅读以下部分

1. 为什么要将页面设置为 HTML5 文档类型？

答：Bootstrap 使用到的某些 HTML 元素和 CSS 属性只有 HTML5 提供。

相关问题：如何设置页面为 HTML4 文档类型？（如果不知道，可以看 W3C 的文档，或者用 Google 搜索）

2. 怎么理解「移动设备优先」？

先满足小屏幕移动设备（手机、平板），再去扩展里面的 components 去满足大屏幕设备（笔记本、台式机）。

移动优先策略

i. 内容 Content

- 决定什么是最重要的

ii. 布局 Layout

- 优先设计更小的宽度。

- Base CSS 强调移动设备；中等关注平板、电脑

iii. 渐进增强 Progressive Enhancement

- 随着屏幕大小增加而添加元素。

3. Bootstrap 排版、链接样式设置了基本的全局样式是什么？

Bootstrap 保留和坚持部分浏览器的基础样式，解决部分潜在的问题，提升一些细节的体验，在排版、链接样式设置了基本的全局样式。具体说明如下：

Intro to Bootstrap

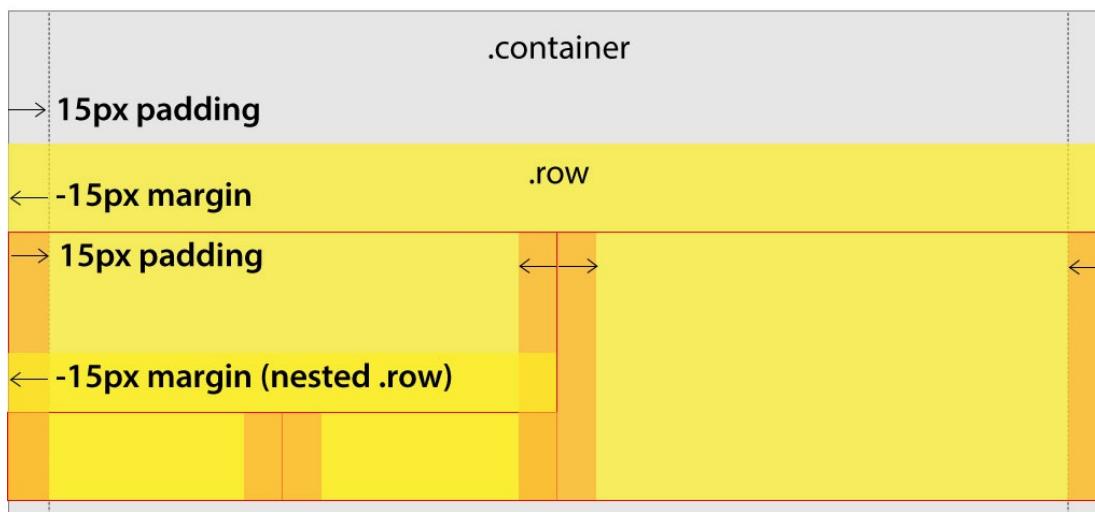
- 移除body的margin声明
- 设置body的背景色为白色 `background-color: #fff;`
- 为排版设置了基本的字体、字号和行高
- 设置全局链接颜色，且当链接处于悬浮 `:hover` 状态时才会显示下划线样式

4. Bootstrap 如何在不同的浏览器表现出一样的效果？

Bootstrap框架的核心是轻量的CSS基础代码库，他并没有一味的重置样式，而是注重各浏览器基础表现，降低开发难度。为了增强跨浏览器表现的一致性，Bootstrap使用了 [Normalize.css](#)。但没有一味全部使用该重置样式，而是在此基础之上进行了一些改良，让其更加符合Bootstrap的设计思想。

5. 布局容器（container）是什么鬼？

顾名思义，就是放Bootstrap代码的容器，只有放在container里面的html代码才会被Bootstrap识别。



注意 我们要把Bootstrap的HTML的代码都放在 带 `container` or `container-fluid` 类的 `<div>` 标签中。

6. .container 和 .container-fluid 都是响应式，那有什么差别？

看源码bootstrap.css可以看到container和container-fluid类本身没有设置width属性值，也就是默认100%，但是container类还设置了媒体查询：

```
@media (min-width: 768px) {  
    .container {  
        width: 750px;  
    }  
}  
  
@media (min-width: 992px) {  
    .container {  
        width: 970px;  
    }  
}  
  
@media (min-width: 1200px) {
```

```
.container {  
    width: 1170px;  
}  
}
```

也就是说container类在浏览器viewport宽度768~992px区间宽度固定为750px，在992~1200px区间宽度固定为970px，大于1200px的时候宽度固定为1170px，所以container类在viewport大于768px的时候会有变化的左右margin，而container-fluid类没有媒体查询设置，所以始终是100%宽度，没有左右margin。

网络搜到一些资料：

- [Bootstrap中container与container-fluid的区别 - 简书](#)
- [bootstrap中container类和container-fluid类的区别 - so what - 博客频道 - CSDN.NET](#)
- [bootstrap中container跟container-fluid的样式是一样的，为什么要分开写？ - 知乎](#)
- [Bootstrap 栅格系统 理解和总结 | JeryTao](#)

作业

1. 快速阅读[全局 CSS 样式 · Bootstrap 中文文档](#)，先只了解有哪些整体知识点。看到[响应式工具](#)即可，后面的Less和Sass可以先不看。
2. 通读，知道这些知识点如何定义。
3. 精度，知道这些知识点如何用代码实现。

下面，我们会进入实战环节，写一些常用的Bootstrap模块，将来大家自己写页面就可以像搭积木一样写出自己的页面。

布局

Options for structuring your pages with Bootstrap, including global styles, required scaffolding, grid system, and more.

自学任务：

1. 自己先快速阅读 [栅格系统](#) 文档。
2. 到 [Bootstrap可视化布局系统](#) 玩一玩，快速体验Bootstrap

1. 背景介绍

栅格系统是Bootstrap中的核心，正是因为栅格系统的存在，Bootstrap才能有着如此强大的响应式布局方案。下面是官方文档中的解说：

Bootstrap 提供了一套响应式、移动设备优先的流式栅格系统，随着屏幕或视口（viewport）尺寸的增加，系统会自动分为最多12列。包含了用于简单的布局选项的预定义类，也包含了用于生成更多语义布局的功能强大的混合类。

我们来理解一下这一段话，可以发现其中最重要的部分就是**移动设备优先**，那么什么是移动设备优先呢？

Bootstrap的基础CSS代码默认从小屏幕设备（比如移动设备、平板电脑）开始，然后使用媒体查询扩展到大屏幕设备（比如笔记本电脑、台式电脑）上的组件和网格。

有着如下策略：

1. 内容：决定什么是最重要的。
2. 布局：优先设计更小的宽度。
3. 渐进增强：随着屏幕大小增加而添加元素。

2. 知识点讲解

1. 数据行（.row）必须包含在容器.container（固定宽度）或.container-fluid（100%宽度）中，以便为其赋予合适的排列（alignment）和内填充（padding）。如：

```
<div class="container"><!-- 水平居中，两边有margin，最小屏幕时，充满父元素 -->
<div class="row"></div>
</div>
<!-- 或者 --
<div class="container-fluid"><!-- 默认一直充满整个父元素 -->
<div class="row"></div>
```

4.1 布局

```
</div>
```

- 在数据行 (.row) 中可以添加列 (column)，但列数之和不能超过平分的总列数 (在超过时，多余部分会换行显示)，默认12。

```
<div class="container">
  <div class="row">
    <div class="col-md-2"></div>
    <div class="col-md-6"></div>
    <div class="col-md-4"></div>
  </div>
</div>
```

- 页面上的具体内容应当放置于列 (column) 内，并且只有列 (column) 可以作为数据行 .row 容器的直接子元素。
- 预定义的网格类，比如 .row 和 .col-xs-4，可用于快速创建网格布局。
- 栅格系统中的列是通过指定**1到12**的值来表示其跨越的范围。例如，三个等宽的列可以使用三个 .col-xs-4 来创建。
 - 如果一“行 (row)”中包含了的“列 (column)”大于 12，多余的“列 (column)”所在的元素将被作为一个整体另起一行排列。
- 下面媒体查询 (media query) 的基本结构，根据查看网页的设备的某些重要信息（比如屏幕大小、分辨率、颜色位深等），页面可以分别应用不同的样式甚至替换整个样式表。

```
@media (media-feature-name: value) {
  /* 符合条件时应用的样式 */
}
```

如果浏览器当前的条件与圆括号中的条件匹配，它就会采用花括号中的那些样式。如果不匹配，浏览器会忽略这些样式。

注意：浏览器始终会采用位于 **@media** 代码块之外的样式。满足条件时的媒体查询样式是在其他样式基础上应用的。为此，条件式媒体查询样式经常要覆盖其他样式，比如隐藏之前的可见元素，把区块移动位置，调整字号等。

3. 用法

4.1 布局

一图胜千言，通过下表可以详细查看 Bootstrap 的栅格系统是如何在多种屏幕设备上工作的。

	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 大桌面显示器 (≥1200px)
栅格系统行为	总是水平排列	开始是堆叠在一起的，当大于这些阈值时将变为水平排列	C	
.container 最大宽度	None (自动)	750px	970px	1170px
类前缀	.col-xs-	.col-sm-	.col-md-	.col-lg-
列 (column) 数	12			
最大列 (column) 宽	自动	~62px	~81px	~97px
槽 (gutter) 宽	30px (每列左右均有 15px)			
可嵌套	是			
偏移 (Offsets)	是			
列排序	是			

3.1 基础

那么我们就来看看一些示例吧，下面这种方式是最基本的用法：

See the Pen [grid1](#) by Yang Wen (@sundevilyang) on [CodePen](#).

点击上图右上角的Edit on CODEPEN看视图效果

任务1

你来试试：制作一个网格，效果图如下：

.col-md-4	.col-md-8	
col-md-4	col-md-4	col-md-4
col-md-3	.col-md-6	.col-md-3

3.2 列偏移 (offset)

在某些情况下，我们不希望相邻的列紧靠在一起，如果你希望通过额外的margin或其他的手段来实现的话，Bootstrap内置为我们提供了列偏移 (offset)，这一系列的类来帮助我们实现想要的效果。

只需要给需要偏移的列元素上添加类名 `col-md-offset-*` (`*`代表要偏移的列组合数)，那么具有这个类名的列就会向右偏移。这些类实际是通过使用 `*{margin-left: 15px;}` 选择器为当前元素增加了左侧的边距 (margin)。例如：在列元素中添加 `.col-md-offset-6` 类将 `.col-md-6` 元素向右侧偏移了6个列 (column) 的宽度。

现在我们的代码是这样的：

See the Pen [grid列偏移](#) by Yang Wen (@sundevilyang) on [CodePen](#).

4.1 布局

点击上图右上角的Edit on CODEPEN看视图效果

从实现的效果我们就能发现一些东西，注意第二段的显示效果与代码，从那里我们可以发现：使用 `col-md-offset-*` 对列进行向右偏移时，要保证列与偏移列的总数不超过12，不然会致列断行显示。其实原因也很简单：因为该类是对于列设置 `margin-left`，并且我们在上面的源码展示中，也可以看有每一列都有着 `float:left` 的属性，从这些地方我们就很难发现在（偏移+列宽）超过12时，为何会换行显示了。

任务2

要求：

1. 一个有4列网格。
2. 第一列网格占2格宽不偏移。
3. 第二列、第三列网格占2格宽，并且偏移1个网格。
4. 第四列网格占3格宽，并且偏移1个网格。

效果图如下：



3.3 列嵌套

Bootstrap框架的网格系统还支持列的嵌套。你可以在一个列中添加一个或者多个行（`.row`）容器，然后在这个行容器中插入列（像前面介绍的一样使用列）。但在列容器中的行容器（`.row`），宽度为100%时，就是当前外部列的宽度。（其实就是在列中嵌套多个列，下面会有实际效果展示）

See the Pen [grid列嵌套](#) by Yang Wen (@sundevilyang) on CodePen.

点击上图右上角的Edit on CODEPEN看视图效果

注意：嵌套的列总数也需要遵循不超过12列。不然会造成末位列换行显示。

任务3

要求：

1. 创建一个8-4列网格。（备注：以中屏md（970px）为例）。
2. 在第一个8列网格中插入8-4列网格。
3. 在第二个4列网格中插入9-3列网格。

效果图如下：



3.4 列排序

列排序其实就是在改变列的方向，就是改变左右浮动，并且设置浮动的距离。在Bootstrap框架的网格系统中是通过添加类名 `col-md-push-*` 和 `col-md-pull-*` (其中星号代表移动的列组合数)。

我们来看一个简单的示例：

See the Pen [grid列排序](#) by Yang Wen (@sundevilyang) on CodePen.

点击上图右上角的Edit on CODEPEN看视图效果

“`col-md-4`”居左，“`col-md-8`”居右，如果要互换位置，需要将“`col-md-4`”向右移动8个列的距离，也就是8个offset，也就是在“

”添加类名“`col-md-push-8`”，调用其样式。

也要将“`col-md-8`”向左移动4个列的距离，也就是4个offset，在“

”上添加类名“`col-md-pull-4`”：

4. 实例

4.1 移动设备和桌面屏幕

是否不希望在小屏幕设备上所有列都堆叠在一起？那就使用针对超小屏幕和中等屏幕设备所定义的类吧，即 `.col-xs-*` 和 `.col-md-*`。我把Bootstrap中文官网的代码稍微做了一点修改。主要是将中间的`.col-md-4`和`.col-xs-6`两个类调换了一下位置（CSS层叠，说到这应该就不需要做更多的说明了吧。），代码如下：

```
<div class="container">
  <div class="row">
    <div class="col-xs-12 col-md-8">.col-xs-12 .col-md-8</div>
    <div class="col-xs-6 col-md-4">.col-xs-6 .col-md-4</div>
  </div>
  <div class="row">
    <div class="col-md-4 col-xs-6">.col-md-4 .col-xs-6</div>
    <div class="col-md-4 col-xs-6">.col-md-4 .col-xs-6</div>
    <div class="col-md-4 col-xs-6">.col-md-4 .col-xs-6</div>
  </div>
  <div class="row">
    <div class="col-xs-6">.col-xs-6</div>
    <div class="col-xs-6">.col-xs-6</div>
  </div>
</div>
```

4.1 布局

宽屏显示效果如下：

.col-xs-12 .col-md-8	.col-md-4 .col-xs-6	.col-xs-6 .col-md-4
.col-md-4 .col-xs-6	.col-xs-6	.col-md-4 .col-xs-6

窄屏显示效果如下：

.col-xs-12 .col-md-8	
.col-xs-6 .col-md-4	
.col-md-4 .col-xs-6	.col-md-4 .col-xs-6
.col-md-4 .col-xs-6	
.col-xs-6	.col-xs-6

其实从上面代码和实际的显示效果，我们就可以看出一些问题来：

1. 在宽屏时，使用 `col-md-*` 的布局方式，在窄屏时使用 `col-xs-*` 布局
2. 响应式的列重置与列中设置的类的排列顺序无关。
3. 响应式列重置，在一个 `.row` 仍然存在列超过12时，`col-xs-6 3=18`。多余列的元素将作为一个整体单元被另起一行排列。2. 在了解了上面的这些问题之后是不是明了了一些呢？

举一反三，我们再来看一个更加复杂的例子

4.2 手机、平板、桌面

在上面案例的基础上，通过增加使用针对平板设备的 `.col-sm-*` 类，我们来创建更加动态和强大的布局吧。

```
<div class="container">
  <div class="row">
    <div class="col-xs-12 col-sm-6 col-md-8">.col-xs-12 .col-sm-6 .col-md-8</div>
    <div class="col-xs-6 col-sm-12 col-md-4">.col-xs-6 .col-sm-12 .col-md-4</div>
  </div>
  <div class="row">
    <div class="col-xs-6 col-sm-4">.col-xs-6 .col-sm-4</div>
    <div class="col-xs-6 col-sm-4">.col-xs-6 .col-sm-4</div>
    <div class="col-xs-6 col-sm-4">.col-xs-6 .col-sm-4</div>
  </div>
</div>
```

这一次会比原来的情况更加复杂，一个会有3种情况出现：

宽屏：桌面

.col-xs-12 .col-sm-6 .col-md-8	.col-xs-6 .col-sm-12 .col-md-4
.col-xs-6 .col-sm-4	.col-xs-6 .col-sm-4

中屏：平板

4.1 布局

.col-xs-12 .col-sm-6 .col-md-8		
.col-xs-6 .col-sm-12 .col-md-4		
.col-xs-6 .col-sm-4	.col-xs-6 .col-sm-4	.col-xs-6 .col-sm-4

窄屏：手机

.col-xs-12 .col-sm-6 .col-md-8		
.col-xs-6 .col-sm-12 .col-md-4		
.col-xs-6 .col-sm-4	.col-xs-6 .col-sm-4	
.col-xs-6 .col-sm-4		

然后我们从这一部分代码与实际效果，再加上上面做的一点小总结，对比之后又能发现一些问题：

1. Bootstrap中对于栅格系统的展示总是优先展示**对应屏幕大小范围的类**（`.col-xs-6` 和 `.col-sm-4` 等等）。
2. 在为不同大小屏幕，添加多个类一定要记住在基本用法处的表格中的内容，它决定了在不同大小屏幕下的显示与排列。
3. 若还要继续添加在超大屏幕下的特定显示效果，只需要对列再添加一个`.col-lg-*` 的类，就能达到你所想要的。

4.3 响应式列重置

除了在上面讲到的内容，在官方文档中还提到了**响应式列重置**，在我看来其实就是以下几点（或者说在这样的条件下可以使用）：

1. 需要对于不同屏幕大小进行处理，添加多个对于列宽进行处理的类。
2. 在缩小屏幕时，同一数据行（`.row`）针对小屏幕设置的类的宽度（`.col-sm-*` 等等）加起来超过12，余下部分进行换行显示。
3. 在不同列的高度，有比较大的差异。可以在未超过12与超过12的部分，两者之间添加下面这样的代码：`<div class="clearfix visible-xs-block"></div>`。（`.visible-xs-block` 这一系列的类，会在后面进行讲解）它最主要的作用是：它是响应式的，会占据未超过12的部分的宽高，且只有在发生列换行显示（堆叠）才进行设置。
4. 为了便于大家理解，我加一句：把上面那段代码复制到数据行`.row`容器内的最末端，它所占据的宽高和`.row`是一样的。（内容中有多段这样的代码的时候不成立，原因大家应该都能看出来了）

基础Bootstrap CSS

设置全局 CSS 样式；基本的 HTML 元素均可以通过 class 设置样式并得到漂亮的效果。

一、排版 Typography

1. 标题

在Bootstrap中使用标题的方式和常规html一样：从 `<h1>` 到 `<h6>` 分别从大到小安排标题元素，只不过在Bootstrap中重新设置了其默认样式，从源码中可以看到有如下共性：

字体颜色与字体样式继承自父元素，字体粗细为500，且行高全部设为1.1（也就是`font-size`的1.1倍）

```
{
  font-family: inherit;
  font-weight: 500;
  line-height: 1.1;
  color: inherit;
}
```

在不同大小的标题中有着如下区别：

在Bootstrap中对于不同级别标题字体大小设置为：**`h1=36px, h2=30px, h3=24px, h4=18px, h5=14px和h6=12px`**。重新设置了 `.margin-top` 和 `.margin-bottom` 的值，**`h1~h3`** 重置后的值都是`20px`；**`h4~h6`**重置后的值都是`10px`。

除此之外在Bootstrap中为了让非标题元素和标题使用相同的样式，还特意定义了 `.h1~.h6` 六个类名。如下所示：

在Bootstrap中，下面的代码，显示效果相同。

```
<!--Bootstrap中的标题-->
<h1>Bootstrap标题一</h1>
<h2>Bootstrap标题二</h2>
<h3>Bootstrap标题三</h3>
<h4>Bootstrap标题四</h4>
<h5>Bootstrap标题五</h5>
<h6>Bootstrap标题六</h6>
<!--Bootstrap中让非标题元素和标题使用相同的样式-->
<div class="h1">Bootstrap标题一</div>
<div class="h2">Bootstrap标题二</div>
<div class="h3">Bootstrap标题三</div>
```

4.2 基础Bootstrap CSS

```
<div class="h4">Bootstrap标题四</div>
<div class="h5">Bootstrap标题五</div>
<div class="h6">Bootstrap标题六</div>
```

2. 文本格式

2.1 正文文本

Bootstrap中的文本全局样式如下：

1. 字体大小为14px
2. 行高为1.42867143 (约等于20px)
3. 字体颜色为#333
4. p标签外部有着10个像素的下外边距margin-bottom:10px;

2.2 强调

字体风格

```
<p class="lead">lead</p><!-- 字体变大，行高变大，下外边距变大 -->
<i>i</i><!-- 无特殊意义， 斜体 -->
<small>small</small><!-- 小号字体-->
<strong>strong</strong><!-- 语气强烈的强调，粗体 -->
<em>em</em><!-- 强调，斜体 -->
<mark>mark</mark>
<del>del</del>
<s>s</s>
<ins>ins</ins>
<u>u</u>
```

强调相关的类

```
<p class="text-muted">提示，使用浅灰色 (#999) </p>
<p class="text-primary">主要，使用蓝色 (#428bca) </p>
<p class="text-success">成功，使用浅绿色 (#3c763d) </p>
<p class="text-info">通知信息，使用浅蓝色 (#31708f) </p>
<p class="text-warning">警告，使用黄色 (#8a6d3b) </p>
<p class="text-danger">危险，使用褐色 (#a94442) </p>
```

利用「整体学习法」中的「可视化技巧」，把颜色和这6个单词的意义在大脑中联想起来吧：）

2.3 文本对齐

4.2 基础Bootstrap CSS

```
<p class="text-left">我居左</p>
<p class="text-center">我居中</p>
<p class="text-right">我居右</p>
<p class="text-justify">我两端对齐</p>
```

请自行阅读官方文档有关概念：改变大小写、缩略语、地址、引用。

2.4 列表

在Bootstrap中对于列表的设置与原生的html基本一致，需要注意的有：

1. 在列表之间有10px的下外边距
2. 在嵌套列表中，不含有下边距

Bootstrap当然不会就这么简单做一点点修改，它定义了一些关于列表的类给我们使用。

2.4.1 去点列表 .list-unstyled

```
list-unstyled {
    padding-left: 0;
    list-style: none;
}
```

从源码中，我们可以看到这样的信息，它除了将项目编号去除之外，还将默认的左边距也清除掉了。

```
<ol>
    <li class="list-unstyled">
        项目列表
        <ul class="list-unstyled">
            <li>带有项目编号</li>
            <li>带有项目编号</li>
        </ul>
    </li>
</ol>
```

在这样一段代码中，三个列表项会整齐的排列在一起，且都没有项目符号。

2.4.2 内联列表 .list-inline

把垂直列表换成水平列表，而且去掉项目符号（编号），保持水平显示。内联列表就是为制作水平导航而生。

小任务：自己在codepen上面试一试

2.4.3 水平定义列表 .dl-horizontal

4.2 基础Bootstrap CSS

```
<dl class="dl-horizontal">
  <dt>标题一: </dt>
  <dd>描述内容, 我很喜欢前端, 也很喜欢响应式布局, 它能在个不同大小的屏幕上提供很好的体验, 我现在是初学者, 但是我会越来越强的</dd>
  <dt>标题二: 标题二: 标题二: 标题二: </dt>
  <dd>描述内容</dd>
</dl>
```

浏览器全屏 和 屏幕变小 显示效果不同, 在遇到一个临界值时 (小屏), 水平定义列表将回复到原始的状态。

这是为什么呢? 我们去看看源码吧! 原来在这里添加了一个媒体查询, 只有屏幕大于 768px 的时候, 添加类名 `.dl-horizontal` 才具有水平定义列表效果。其实现主要方式:

1. 将`dt`设置了一个左浮动, 并且设置了一个宽度为160px
2. 将`dd`设置一个`margin-left`的值为180px, 达到水平的效果
3. 当标题宽度超过160px时, 将会显示三个省略号

2.5 代码 Code

在Bootstrap主要提供了三种代码风格:

1. 使用 `<code></code>` 来显示单行内联代码——针对于单个单词或单个句子的代码
2. 使用 `<pre></pre>` 来显示多行块代码——针对于多行代码 (也就是成块的代码)
3. 使用 `<kbd></kbd>` 来显示用户输入代码——表示用户要通过键盘输入的内容

需要注意的是, 不管使用哪种代码风格, 在代码中碰到小于号 (<) 和大于号 (>) 都需要使用转义字符来替代

```
code风格:
<div>Bootstrap的代码风格有三种: <code>&lt;code&gt;</code>、<code>&lt;pre&gt;</code>和<code>&lt;kbd&gt;</code></div>

pre风格:
<div>
<pre>
&lt;ul&gt;
  &lt;li&gt;...&lt;/li&gt;
&lt;/ul&gt;
</pre>
</div>

kbd风格:
<div>请输入<kbd>ctrl+c</kbd>来复制代码, 然后使用<kbd>ctrl+v</kbd>来粘贴代码</div>
```

小任务: 自己在Codepen上面看看效果

3. 表格 Table

4.2 基础Bootstrap CSS

Bootstrap为表格提供了1种基础样式和4种附加样式以及1个支持响应式的表格。

类名	表格样式
.table	基础表格
.table-striped	斑马线表格
.table-bordered	带边框的表格
.table-hover	鼠标悬停高亮的表格
.table-condensed	紧凑型表格
.table-responsive	响应式表格

下面说一些注意事项：

1. 后几种表格附加样式，必须在基础样式
2. 响应式表格：其原理是在表格外部添加容器把普通表格包裹起来，下面进行详细说明：

```
<div class="table-responsive"><!-- 关键！容器包裹后实现响应式 -->
  <table class="table table-bordered"><!-- 设置表格样式，带边框的表格 -->
    <thead><!-- 一个表格应该有表头，若直接写tr>td这样的结构，浏览器会自动创建一个tbody包裹 -->

      </thead>
      <tbody><!--与上同理，即使你不创建，浏览器也会自动添加tbody包裹你的代码 -->
        </tbody>
    </table>
</div>
```

二、表单Forms

关于表单的应用自然不需要多说，只要您的网站可以用户登录，那么不可能不用到表单！表单主要功能是用来与用户做交流的一个网页控件，JavaScript发明之初最大的作用也就是用来进行表单操作。所以表单是每一个前端开发者必须要熟练掌握的东西。

良好的表单设计能够让网页与用户更好的沟通。表单中常见的元素主要包括：**文本输入框、下拉选择框、单选按钮、复选按钮、文本域和按钮等**。其中每个控件所起的作用都各不相同，而且不同的浏览器对表单控件渲染的风格都各有不同。传统的HTML表单就不在这样进行讲解了，下面就开始Bootstrap表单的学习吧！

1. 基础表单

在Bootstrap对于基础表单没有做太多的定制效果，仅仅是对于一些元素的**margin**、**padding**和**border**进行了一些细化设置。当然，也不会那么简单，在Bootstrap框架中，通过定制了一个类名 `form-control`，也就是说，使用了类名 `form-control`，将会实现一些设计上的定制效果。从源码中（2356行，嗯，善用搜索）可以得出结论如下：

1. 宽度变成了100%
2. 设置了一个浅灰色（#ccc）的边框
3. 具有4px的圆角
4. 设置阴影效果，并且元素得到焦点之时，阴影和边框效果会有所变化
5. 设置了placeholder(输入框的提示文字)的颜色为 #999 这个类一般用于输入框，对于其他的类型使用，嗯，你可以试试自己尝试一下。

```
<form role="form">
  <div class="form-group"><!-- form-group这个类在在这里用于控制表单之间的间隔，之后会有另外
介绍 -->
    <label for="exampleInputEmail1">邮箱: </label>
    <input type="email" class="form-control" id="exampleInputEmail1" placeholder="请输入您的邮箱地址">
  </div>
  <div class="form-group">
    <label for="exampleInputPassword1">密码</label>
    <input type="password" class="form-control" id="exampleInputPassword1" placeholder="请输入您的邮箱密码">
  </div>
</form>
```

效果如下：

邮箱：

请输入您的邮箱地址

密码

请输入您的邮箱密码

内联表单

还是上面那一段代码：我们只需要给form加上下面 `.form-inline` 这个类就可以实现表单元素在一排排列：

邮箱：

请输入您的邮箱地址

密码

请输入您的邮箱密码

4.2 基础Bootstrap CSS

水平表单

Bootstrap框架默认的表单是垂直显示风格，除了上面的内联表达之外，很多时候我们需要的是水平表单风格。在Bootstrap框架中要实现水平表单效果，必须满足以下两个条件：

1. 在

元素是使用类名.form-horizontal，作用如下：

- 设置表单控件padding和margin值。
- 改变“form-group”的表现形式，类似于栅格系统的“row”。

2. 配合Bootstrap框架的栅格系统。（栅格布局会在以后的章节中详细讲解）

- 例如：使用

将我们的input进行包裹，具体情况到栅格布局的时候再说 - 可以实现下面的效果，（同样处于媒体查询中）

The image shows two horizontal input fields. The first field is labeled '邮箱' (Email) and contains the placeholder '请输入您的邮箱地址'. The second field is labeled '密码' (Password) and contains the placeholder '请输入您的邮箱密码'.

2. 表单控件

输入框input

单行文本框在html编写时，我们都需要设置type属性为text，在Bootstrap中也必须正确的添加type的类型，因为在这里是使用CSS的属性选择器如：`input[type="text"]` 来进行样式设置的！为了让控件在各种表单风格中样式不出错，需要添加类名 `form-control`，如：`<input type="email" class="form-control" placeholder="请输入您的邮箱">`

任务：自行阅读Bootstrap文档以下内容

- 下拉选择框select
- 文本域textarea
- 复选框checkbox和单选按钮radio
- 表单控件大小
- 表单控件状态
- 验证状态
- 表单提示信息

3. 按钮 Buttons

我们创建的按钮通常通过下面这几种形式：

4.2 基础Bootstrap CSS

- `input[type="submit"]` : 提交按钮
- `input[type="button"]` : 普通按钮
- `input[type="reset"]` : 重置按钮
- `<button></button>` : 按钮元素, (默认行为是提交)

在Bootstrap框架中的按钮都是采用 `<button>` 来实现。

1. 情景按钮

也正如我们前面所说, Bootstrap对于按钮也做了不同情景下的类, 用不同的颜色展示很好的描述了当前的状态。

类名	用途	描述	颜色		效果
<code>.btn</code>	基础按钮	Bootstrap 框架的基础按钮风格	背景	默认浅灰色	
			边框	透明色	
			文本	#000	
<code>.btn-default</code>	默认按钮	Bootstrap 默认按钮风格	背景	#fff	
			边框	#ccc	
			文本	#333	

4.2 基础Bootstrap CSS

.btn-primary	主要按钮	提示浏览器当前按钮在一系 列按的按钮中 为主要操作	背景	#3276b1	主要按钮.btn-primary
			边框	#285e8e	
			文本	#fff	
.btn-success	成功按钮	表示成功或积 极的动作	背景	#5cb85c	成功按钮.btn-success
			边框	#4cae4c	
			文本	#fff	
.btn-info	信息按钮	可以用来替换 默认按钮	背景	#5bc0de	信息按钮.btn-info
			边框	#46b8da	
			文本	#fff	
.btn-warning	警告按钮	提醒用户操作 应该谨慎	背景	#f0ad4e	警告按钮.btn-warning
			边框	#eea236	
			文本	#fff	
.btn-danger	危险按钮	提醒用户当前 操作非常危险	背景	#d9534f	危险按钮.btn-danger
			边框	#d43f3a	
			文本	#fff	
.btn-link	链接按钮	把按钮转化成 链接风格，让其 看起来就像一 个链接	背景	无	链接按钮.btn-link
			边框	透明	
			文本	#428bca	

记住一点！Bootstrap中的按钮，第一前提是必须添加 `btn` 这个类，其他的各种类，都只能添加到其后。所以一个按钮应该是这个样子的：`<button class="btn btn-info">信息按钮</button>`，若还有其他的样式，往 `btn-info` 后添加，充分利用css的层叠特性。

2. 按钮大小

Bootstrap有着很对相似的类（学习完后总结一下吧颜色，大小等等），比如现在要说的，改变按钮的大小，当然你可以自定义，但是Bootstrap默认为我们提供了一些，添加到 `btn` 类后就行！

4.2 基础Bootstrap CSS

类名	描述	控制参数		效果
<code>.btn-lg</code>	变大	padding	10px 16px	
		font-size	18px	
		line-height	1.33	
		border-radius	6px	
	正常	padding	6px 12px	
		font-size	14px	
		line-height	1.42857143	
		border-radius	4px	
<code>.btn-sm</code>	变小	padding	5px 10px	
		font-size	12px	
		line-height	1.5	
		border-radius	3px	
<code>.btn-xs</code>	超小	padding	1px 5px	
		font-size	12px	
		line-height	1.5	
		border-radius	3px	

3. 块状按钮

Bootstrap框架中提供了一个类名btn-block。使用这个类名就可以让按钮充满整个容器，并且这个按钮不会有任何的padding和margin值。在实际当中，常把这种按钮称为块状按钮。同样添加到btn类后即可生效

4. 活动与禁用状态

Bootstrap框架针对按钮的状态做了一些特殊处理。主要分为两种：活动状态和禁用状态。

活动状态

Bootstrap按钮的活动状态主要包括按钮的悬浮状态(`:hover`)，点击状态(`:active`)和焦点状态 (`:focus`) 几种。

通过在源码中我们可以看到，Bootstrap对于每一种风格的按钮都做了单独的定制。需要进行修改的话可以参考源码！

禁用状态

在Bootstrap框架中，要禁用按钮有两种实现方式：

- 在标签中添加disabled属性
- 在元素标签中添加类名 .disabled

三、图片 Images

在Bootstrap框架中对于图像的样式风格提供以下几种风格：

- .img-responsive : 响应式图片，主要针对于响应式设计
- .img-rounded : 圆角图片
- .img-circle : 圆形图片
- .img-thumbnail : 缩略图片，相框效果

直接把类名添加到img标签上就行了！

注意：设置图片大小时，由于样式没有对图片做大小上的样式限制，所以在实际使用的时候，需要通过其他的方式来处理图片大小。比如说控制图片容器大小。（注意不可以通过css样式直接修改img图片的大小，这样操作就不响应了）

图标 Icons by Glyphicons

这里说的图标就是Web制作中常看到的小icon图标，包括250多个来自 Glyphicon Halflings 的字体图标，Bootstrap框架中图标都是glyphicons.com这个商业网站提供的，并且免费授权给Bootstrap框架使用。

不过，大家普遍更愿意使用Font Awesome，使用方法非常简单：

1. 在页面顶部添加 `<link rel="stylesheet" href="//maxcdn.bootstrapcdn.com/font-awesome/4.5.0/css/font-awesome.min.css"/>`
2. 添加 `.fa .icon_name` 到 `<i>` 标签。这里的icon_name 到Font Awesome网站查询：<http://fontawesome.io/icons/> 如下列代码：

```
<div class="list-group">
    <a class="list-group-item" href="#"><i class="fa fa-home fa-fw" aria-hidden="true"></i>&ampnbsp Home</a>
    <a class="list-group-item" href="#"><i class="fa fa-book fa-fw" aria-hidden="true"></i>&ampnbsp Library</a>
    <a class="list-group-item" href="#"><i class="fa fa-pencil fa-fw" aria-hidden="true"></i>&ampnbsp Applications</a>
    <a class="list-group-item" href="#"><i class="fa fa-cog fa-fw" aria-hidden="true"></i>&ampnbsp Settings</a>
</div>
```


组件

无数可复用的组件，包括字体图标、下拉菜单、导航、警告框、弹出框等更多功能。

我们接下来要学习的东西呢，是Bootstrap为我们提供的一些组件，这些组件的使用非常简单，可以方便快捷帮助我们搭建网站，并且能达到一个不错的显示效果。我们最需要注意的就是：HTML的结构和bootstrap提供添加到结构上的类以及属性。这些组件的交互功能是依赖于jQuery库实现的，所以还必须引入jQuery.js，且必须在Bootstrap.js之前。正式上线的话直接使用压缩版本就可以了如下：

```
<script src="../js/jquery-min-1.11.3.js"></script>
<script src="../js/bootstrap.min.js"></script>
```

一、菜单

1. 基本用法

在官方文档中，我们看到的下拉菜单组件是这样的：

```
<div class="dropdown">
  <button class="btn btn-default dropdown-toggle" type="button" id="dropdownMenu1" data-toggle="dropdown" aria-haspopup="true" aria-expanded="true">
    下拉菜单<span class="caret"></span>
  </button>
  <ul class="dropdown-menu" aria-labelledby="dropdownMenu1">
    <li><a href="#">菜单项1</a></li>
    <li><a href="#">菜单项2</a></li>
  </ul>
</div>
```

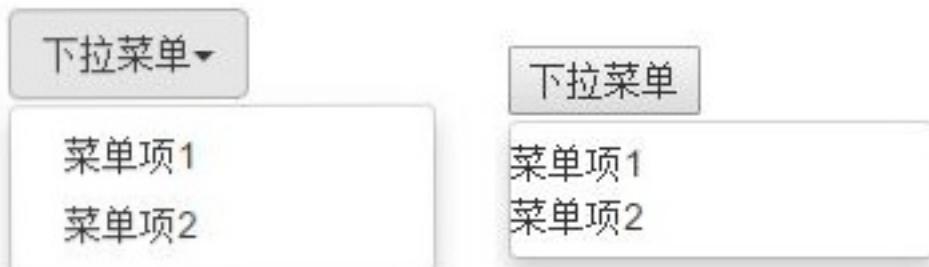
那么，到底哪一些属性是必须的、用来控制行为的呢？又有哪一部分是起到修饰作用的？

我们来自己简化一下这一段代码：

```
<div class="dropdown">
  <button class="dropdown-toggle" data-toggle="dropdown">下拉菜单</button>
  <ul class="dropdown-menu" >
    <li>菜单项1</li>
    <li>菜单项2</li>
  </ul>
</div>
```

4.3 组件

我们来看看简化前后的对比图：



在使用Bootstrap框架中的下拉菜单组件时，其结构运用的正确与否非常的重要，如果结构和类名未使用正确，直接影响组件是否能正常运用。经过我们上面的简化，最终可以看到核心要求有以下几点：

1. 使用一个名为 `.dropdown` 的容器包裹了整个下拉菜单元素，示例中为：`<div class="dropdown"></div>`
2. 使用了一个 `<button>` 按钮做为父菜单，并且定义类名 `.dropdown-toggle` 和自定义 `data-toggle` 属性，且值必须和最外容器类名一致，此示例为：`data-toggle="dropdown"`
3. 下拉菜单项使用一个`ul`列表，并且定义一个类名为 `.dropdown-menu`，此示例为：`<ul class="dropdown-menu">`

也就是说，我们仅仅需要牢记这3个步骤我们就可以使下拉菜单生效，`<button>` 内的 `` 标签，也仅仅是用来进行图标的展示，让菜单更加清晰。不过虽然如此，我建议在实际的使用过程中，把Bootstrap框架中的下拉菜单的**其它属性也一并加上**，这样更加友好的为不同的客户如：**屏幕阅读器** 提供了一个优质的体验。

那么这些是怎么起作用的呢？

查看CSS源码可以看到 `.dropdown-menu` 有一个 `display:none`，所以下拉列表项默认隐藏的；我们继续打开chrome开发者工具（F12），可以看到每一次在列表项展现出来时，`.dropdown` 类后还增加了一个 `.open` 的类；不用多说想肯定就是这个类在起作用，我们再继续回到CSS源码中去可以看到：

```
.open > .dropdown-menu {  
    display: block;  
}
```

4.3 组件

看到这里也应该不用过多说明了，添加了这个 `.open` 类时，`.dropdown-menu` 自然的就显示出来了。

2. 下拉菜单修饰

2.1 下拉分隔线

在Bootstrap框架中的下拉菜单还提供了下拉分隔线，假设下拉菜单有两个组，那么组与组之间可以通过添加一个空的 ``，并且给这个 `` 添加类名 `.divider` 来实现添加下拉分隔线的功能。`<li class="divider">` 对应的样式代码：

```
.dropdown-menu .divider {  
    height: 1px;  
    margin: 9px 0;  
    overflow: hidden;  
    background-color: #e5e5e5;  
}
```

2.2 菜单标题

上一小节讲解通过添加 `divider` 可以将下拉菜单分组，为了让这个分组更明显，还可以给每个组添加一个头部（标题）：`<li class="dropdown-header"> 第一部分菜单头部 `。

从源码中比较关键的两个点是：标题的字体颜色是：`color: #777;`，还有内填充 `padding: 3px 20px;`

2.3 对齐方式

Bootstrap框架中下拉菜单默认是左对齐也就是添加`.dropdown-menu-left`类所能达到的效果，如果你想让下拉菜单相对于父容器右对齐时，可以在 `.dropdown-menu` 上（也就是你的菜单项的ul）添加一个 `.pull-right` 或者 `.dropdown-menu-right` 类名，如下所示：`<ul class="dropdown-menu pull-right" >` 源码如下：

```
.dropdown-menu.pull-right,  
.dropdown-menu-right /*两个类自选*/  
right: 0;  
left: auto;  
}
```

当然你只进行这样的设置的话，可能还会有一些奇怪的情况出现，比如列表项跑到屏幕最右边去了，所以同时一定要为 `.dropdown` 添加 `float:left` 样式。

```
.dropdown{  
    float: left;  
}
```



2.4 菜单项状态

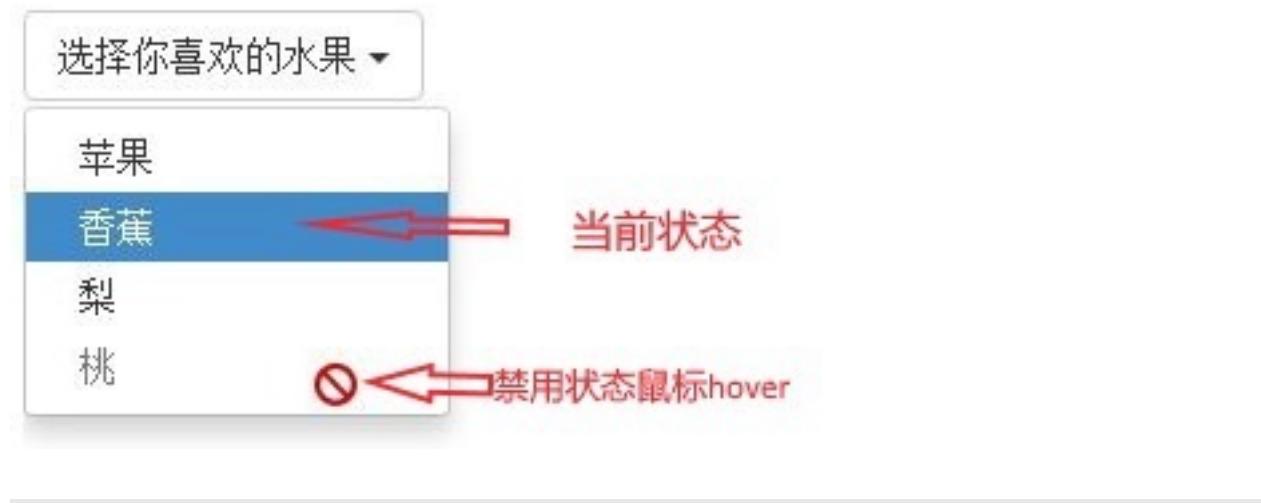
下拉菜单项的默认的状态（不用设置）有悬浮状态（:hover）和焦点状态（:focus）：
(包括下面提到的状态都必须在li中包含a标签才有--)

下拉菜单项除了上面两种状态，还有当前状态（.active）和禁用状态（.disabled）。这两种状态使用方法只需要在对应的菜单项上添加对应的类名如：

- 桃

4.3 组件

我们可以实现下面这样的效果：



二、按钮

1. 按钮组及工具栏

单个按钮的运用有时候并不能满足我们的业务需求，常常会看到多个按钮组合使用，比如富文本编辑器里的一组小图标按钮等。

对于结构方面，非常的简单。使用一个名为 `.btn-group` 的容器，把多个按钮放到这个容器中。如：

```
<div class="btn-group">
  <button type="button" class="btn btn-default">向左</button>
  <button type="button" class="btn btn-danger">中间</button>
  <button type="button" class="btn btn-default">向右</button>
</div>
```

如何转化成工具栏呢？只需要再用一个 `.btn-toolbar` 的内容，将我们的 `.btn-group` 再进行一次包裹，同时，我们还可以使用 `.btn-group-lg` 大按钮组、`.btn-group-xs` :超小按钮组这一系列的类添加到 `.btn-group` 容器来进行对一组按钮进行大小控制如下：

```
<div class= "btn-toolbar">
  <!-- ..... -->
  <div class= "btn-group ">
    <button type="button" class="btn btn-default">向左</button>
    <button type="button" class="btn btn-danger">中间</button>
    <button type="button" class="btn btn-default">向右</button>
  </div>
  <div class= "btn-group btn-group-sm">
    <button type="button" class="btn btn-default">向左</button>
    <button type="button" class="btn btn-danger">中间</button>
    <button type="button" class="btn btn-default">向右</button>
  </div>
</div>
```

4.3 组件

```
</div>
<!-- ..... -->
</div>
```

上面两段代码效果如下：



2. 嵌套分组

这里的嵌套分组其实就是在分组按钮中嵌套了一个下拉菜单。使用的时候，只需要把当初制作下拉菜单的 `.dropdown` 的容器换成 `.btn-group`，并且和普通的按钮放在同一级。如下所示：

```
<div class="btn-group">
    <button class="btn btn-default" type="button">首页</button>
    <div class="btn-group">
        <button class="btn btn-default dropdown-toggle" data-toggle="dropdown" type="button">产品展示<span class="caret"></span></button>
        <ul class="dropdown-menu">
            <li><a href="#">公司简介</a></li>
            <li><a href="#">企业文化</a></li>
            <li><a href="#">组织结构</a></li>
            <li><a href="#">客服服务</a></li>
        </ul>
    </div>
    <button class="btn btn-default" type="button">关于</button>
</div>
```

当然我们还可以从水平按钮组变成垂直分组，只需要在水平分组的最外层容器的类名 `.btn-group` 替换成 `.btn-group-vertical`，就可以实现垂直分组的效果。

按钮的向下向上三角形

我们在下拉菜单和嵌套分组都都看到了一个按钮的向下三角形，我们是通过在 `<button>` 标签中添加一个 `` 标签元素，并且命名为 `caret`，这个三角形完全是通过CSS代码来实现的：

```
.caret {
    display: inline-block;
    width: 0;
    height: 0;
```

4.3 组件

```
margin-left: 2px;
vertical-align: middle;
border-top: 4px solid;
border-right: 4px solid transparent;
border-left: 4px solid transparent;
}
```

有的时候我们的下拉菜单会向上弹起，这个时候我们的三角方向需要朝上显示，实现方法：需要在 `.btn-group` 类上追加 `dropdown` 类名（这也是做向上弹起下拉菜单要用的类名）。

```
.dropdown .caret{
  content: "";
  border-top: 0; /*可以看到这里top与bottom的变化*/
  border-bottom: 4px solid;
}
```

那么现在就来对比一下，上面代码以及类的添加可以达到的不同效果：



三、导航

可以说，每一个网页里面都会有导航的存在，便于用户查找网站所提供的各项功能服务。那么如何使用Bootstrap框架制作各式各样的导航呢？

1. 基础样式

Bootstrap框架中制作导航条主要通过 `.nav` 样式。默认的 `.nav` 样式不提供默认的导航样式，必须附加另外一个样式才会有效，比如`nav-tabs`、`nav-pills`之类。

```
<ul class="nav nav-tabs">
  <li><a href="#">HTML5</a></li>
  <li><a href="#">CSS3</a></li>
  <li><a href="#">JavaScript</a></li>
</ul>
```

当然他们同样也有当前状态（.active）和禁用状态（.disabled），我们给第一个导航添加当前状态，最后一个导航添加禁用状态，实现效果如下：



2. 修饰

2.1 垂直堆叠的导航

在实际运用当中，除了水平导航之外，还有垂直导航，就类似前面介绍的垂直排列按钮一样。制作垂直堆叠导航只需要在 .nav-pills 的基础上添加一个 .nav-stacked 类名即可，同时，我们和下拉菜单一样，同样可以使用组之间的分割线例，只需要在导航项之间添加 `<li class="nav-divider">`，如：

```
<ul class="nav nav-pills nav-stacked">
  <li class="active"><a href="#">HTML5</a></li>
  <li><a href="#">CSS3</a></li>
  <li><a href="#">JavaScript</a></li>
  <li class="nav-divider"></li>
</ul>
```

复制到CODEPEN中看看效果吧！

你应该已经发现了在垂直导航中，每一个导航项都是占文字居左，占100%的父容器宽度的，如果你想有响应式的效果，应该和栅格系统一起使用，另外为什么不试试使用nav-tabs来实现垂直导航呢？看看效果，你就会明白为什么不使用了。

2.2 自适应导航

4.3 组件

自适应导航和前面制作自适应按钮是一样的，不过更换了一个类名，`.nav-justified`。当然他需要和`.nav-tabs`或者`.nav-pills`配合在一起使用。它是响应式的，在宽度足够时，给多个导航项，均分空间；在达到一个临界值（768px）时，它会和垂直导航一样：每一项都充满父容器的宽度，不同的是，他的文字是居中显示的

2.3 导航中加入下拉菜单（二级导航）

前面介绍的都是使用制作一级导航，但很多时候，在Web页面常常会用到二级导航的效果。在Bootstrap框架中制作二级导航就更容易了。只需要将li当作父容器，使用类名`.dropdown`，同时在`li`中嵌套另一个列表`ul`

```
<ul class="nav nav-pills">
    <li><a href="#">HTML5</a></li>
    <li><a href="#">CSS3</a></li>
    <li class="dropdown">
        <a href="#" class="dropdown-toggle" data-toggle="dropdown">JavaScript<span
            class="caret"></span></a>
        <ul class="dropdown-menu">
            <li><a href="#">JQuery</a></li>
            <li><a href="#">Vue</a></li>
            ...
        </ul>
    </li>
```

与上面的嵌套按钮效果一致。

2.4 面包屑式导航

面包屑(Breadcrumb)一般用于导航，主要是起的作用是告诉用户现在所处页面的位置（当前位置）。一般在文章、博客、列表处用的比较多，需要用到`.breadcrumb`类。

```
<ol class="breadcrumb">
    <li><a href="#">个人首页</a></li>
    <li><a href="#">标签</a></li>
    <li class="active">Bootstrap</li>
</ol>
```

个人首页 / 标签 / Bootstrap

可以实现的效果：

四、导航条

导航条（navbar）和我们在上一个笔记中介绍的导航（nav）相比，就多了一个“条”字。其实在Bootstrap框架中他们有着一定的区别。在导航条(navbar)中有一个背景色、而且导航条可以是纯链接（类似导航），也可以是表单，还有就是表单和导航一起结合等多种形式。它们在移动设备上可以折叠（并且可开可关），且在视口（viewport）宽度增加时逐渐变为水平展开模式。

1. 基础导航条

实际上，导航条和导航在外观上的差别并不是那么的大，但是在实际使用中导航条要比导航复杂得多。下面就来实现一个最基础的导航条。

See the Pen [basic-navbar](#) by Yang Wen (@sundevilyang) on [CodePen](#).

点击上图HTML，再点击Result看看效果。拉伸浏览器，分别看看在宽屏和窄屏的效果。

在制作一个基础导航条时，主要注意以下几点：

- 首先在制作导航的列表(`<ul class="nav">`)基础上添加类名`.navbar-nav`
- 在列表外部添加一个容器`<div>`在这里允许的话推荐使用`<nav>`，并且使用类名`.navbar`和`.navbar-default`
- `.navbar`类的主要功能就是设置左右padding和圆角等效果，通过`.navbar-default`来进行控制导航条的颜色比如还有`.navbar-inverse`。

2. 带标题、二级菜单

在Web页面制作中，一般都可以在菜单前面看到有一个标题（文字字号比其它文字稍大一些），其实在Bootstrap框架也为大家做了这方面考虑，其通过`.navbar-header`和`.navbar-brand`来实现，如：

```
<div class="navbar-header">
  <a href="#" class="navbar-brand">WEB开发</a>
</div>
```

当然也可以用来做logo，中文官网描述，在此不进行详细描述。

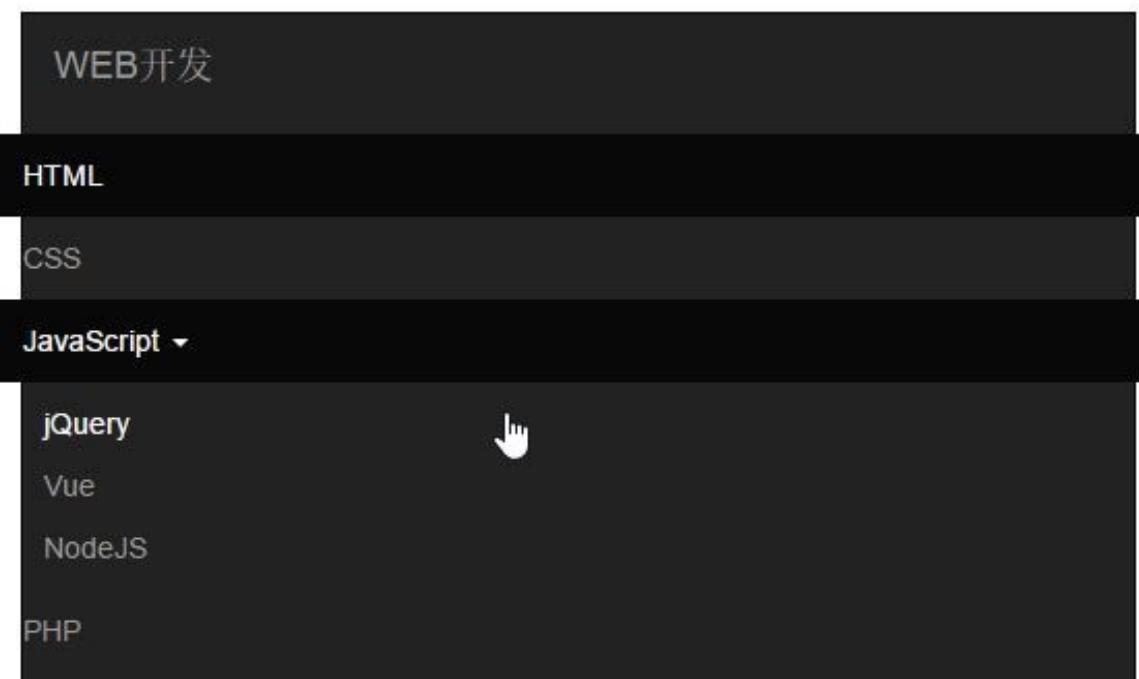
二级菜单自然不用多说，我们一起来看看下面这段代码以及能实现的效果：（注意这里使用的是`navbar-inverse`反色的导航条）

```
<div class="navbar navbar-inverse" role="navigation">
  <div class="navbar-header">
    <a href="#" class="navbar-brand">WEB开发</a>
  </div>
```

4.3 组件

```
<ul class="nav navbar-nav">
    <li class="active"><a href="#">HTML</a></li>
    <li><a href="#">CSS</a></li>
    <li class="dropdown" >
        <a href="#" class="dropdown-toggle" data-toggle="dropdown">JavaScript
        <span class="caret"></span></a>
        <ul class="dropdown-menu">
            <li><a href="#">jQuery</a></li>
            <li><a href="#">Vue</a></li>
            <li><a href="#">NodeJS</a></li>
        </ul>
    </li>
    <li ><a href="#">PHP</a></li>
</ul>
</div>
```

效果如下：



我们常常在导航条中都能看到搜索表单，比如知乎，新浪微博等等都具有，Bootstrap框架当然不可能缺掉这样的东西。使用方法也非常简单：

在Bootstrap框架中提供 `.navbar-form`，在 `.navbar` 容器中放置一个带有 `.navbar-form` 类名的表单；再添加 `.navbar-left` 让表单左浮动，更好实现对齐。在Bootstrap中，还提供了 `.navbar-right` 样式，让元素在导航条靠右对齐。

```
<form action="#" class="navbar-form navbar-left" role="search">
    <div class="form-group">
        <input type="text" class="form-control" placeholder="请输入关键词" />
    </div>
    <button type="submit" class="btn btn-default">搜索</button>
</form>
```

3. 按钮、文本和链接

除了上面的一些样式之外，bootstrap框架提供了三种其他样式：

1. 导航条中的按钮`navbar-btn`，使不在`<form>`中的按钮在导航条里垂直居中
1. 导航条中的文本`navbar-text`，一般使用`<p>`元素，达到最优的行距
1. 导航条中的普通链接`navbar-link`，非导航的链接，添加这个类，可以有正确的默认颜色与反色设置

但这三种样式在框架中使用时受到一定的限制，需要和`navbar-brand`、`navbar-nav`配合起来使用。而且对数量也有一定的限制，一般情况在使用一到两个不会有太大问题，超过两个就会有问题。

4. 固定导航条

有时，我们希望导航条固定在浏览器顶部或底部，Bootstrap为我们方便的提供了两个类来实现这样的样式：

.navbar-fixed-top：导航条固定在浏览器窗口顶部 .navbar-fixed-bottom：导航条固定在浏览器窗口底部 使用方法很简单，只需要在制作导航条最外部容器（`navbar`）上追加对应的类名即可。

```
<nav class="navbar navbar-default navbar-static-top">
  ...
</nav>
```

这个固定的导航条会遮住页面上的其它内容，除非你给元素底部设置了padding。提示：导航条的默认高度是 50px。如：`body { padding-bottom: 70px; }`，当然你也可以使用你自己的值。

5. 响应式导航条

在上面的例子中，导航条就仅仅能在大屏幕上有一个不错的效果，在屏幕变小的情况下，还直接把所有的导航项直接展现出来就不适合了。为了有一个更好的显示效果，响应式的导航条设计也就随之而来。我们先来看看效果，再一步一步分析它！

响应式导航条

窄屏时，点击右边的按钮图标可展开，效果与上面的展开效果除了`active`状态不再溢出容器外，其他基本相同。实现代码如下：

```
<div class="navbar navbar-default" role="navigation">
  <div class="navbar-header">
    <!-- .navbar-toggle 样式用于 toggle 收缩的内容，即 navbar-collapse collapse 样式所在元素 -->
```

4.3 组件

```
<button class="navbar-toggle" type="button" data-toggle="collapse" data-target=".navbar-responsive-collapse">
    <span class="sr-only">Toggle Navigation</span>
    <span class="icon-bar"></span>
    <span class="icon-bar"></span>
    <span class="icon-bar"></span>
</button>
<!-- 确保无论是宽屏还是窄屏，navbar-brand都显示 --&gt;
&lt;a href="#" class="navbar-brand"&gt;WEB开发&lt;/a&gt;
&lt;/div&gt;
<!-- 屏幕宽度小于768px时，div.navbar-responsive-collapse容器里的内容都会隐藏，显示icon-bar图标，当点击icon-bar图标时，再展开。屏幕大于768px时，默认显示。 --&gt;
&lt;div class="collapse navbar-collapse navbar-responsive-collapse"&gt;
    &lt;ul class="nav navbar-nav"&gt;
        &lt;li class="active"&gt;&lt;a href="#"&gt;HTML&lt;/a&gt;&lt;/li&gt;
        &lt;li&gt;&lt;a href="#"&gt;CSS&lt;/a&gt;&lt;/li&gt;
        &lt;li&gt;&lt;a href="#"&gt;JavaScript&lt;/a&gt;&lt;/li&gt;
        &lt;li&gt;&lt;a href="#"&gt;PHP&lt;/a&gt;&lt;/li&gt;
    &lt;/ul&gt;
&lt;/div&gt;
&lt;/div&gt;</pre>
```

分析：

和普通导航条一样，最外层使用 `<div class="navbar navbar-default" role="navigation">` 容器包裹，主要是 `.navbar` 和 `.navbar-default` 类。保证在窄屏时需要折叠的内容必须包裹在带一个div内，并且为这个div加入 `.collapse`、`.navbar-collapse` 两个类名。最后为这个 `div` 添加一个class类名或者id名。保证在窄屏时要显示的图标样式（固定写法）：

```
<button class="navbar-toggle" type="button" data-toggle="collapse">
    <span class="sr-only">Toggle Navigation</span>
    <span class="icon-bar"></span>
    <span class="icon-bar"></span>
    <span class="icon-bar"></span>
</button>
```

并为button添加`data-target=".类名/#id名"`，究竟是类名还是id名呢？由需要折叠的div来决定。（这里很关键，你可以把上面的代码在这里进行修改，会发现无法展开）

五、分页

分页导航几乎在所有的网站都可以看到，好的分页导航可以给用户带来更好的用户体验，还可以方便爬虫在网页中进行搜索。

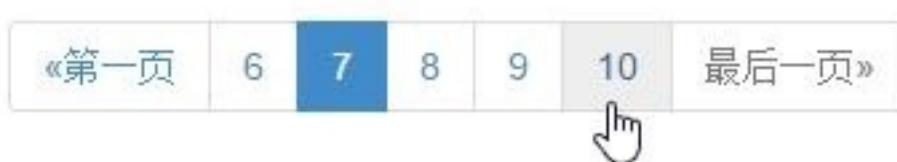
1 带页码的分页导航

4.3 组件

带页码的分页导航，应该是最常见的一种分页导航，特别是在列表页内容很多的时候，会给用户提供带页码的分页导航方式，在Bootstrap中使用的是 `ul>li>a` 这样的结构，在 `ul` 标签上加入 `pagination` 类：

```
<ul class="pagination">
    <li><a href="#">« 第一页 </a></li>
    <li><a href="#">6 </a></li>
    <li class="active"><a href="#">7 </a></li>
    <li><a href="#">8 </a></li>
    <li><a href="#">9 </a></li>
    <li><a href="#">10 </a></li>
    <li class="disabled"><a href="#">最后一页 » </a></li>
</ul>
```

实现效果如下：



可以看到同样会有 `hover`、`active`、`disabled` 等等状态，当然它也可以调整大小：在外层容器中的 `.pagination` 类后，添加 `.pagination-lg` 让分页导航变大；`.pagination-sm` 让分页导航变小：

2 翻页分页导航

其实这种方式，就是看不到具体的页码，只会提供一个上一页和下一页的按钮。

使用方式很简单，也就是在 `ul` 标签上添加 `pager` 类：

```
<ul class="pager">
    <li><a href="#">« 上一页 </a></li>
    <li><a href="#">下一页 » </a></li>
</ul>
```

它默认是居中显示的，如果我们需要一个居左一个居右显示的话，添加两个样式：`previous` 让“上一步”按钮居左；`next` 让“下一步”按钮居右。（同样可以使用 `disabled` 表示禁用状态）

```
<ul class="pager">
    <li class="previous disabled"><a href="#">« 上一页 </a></li>
    <li class="next"><a href="#">下一页 » </a></li>
</ul>
```

六、其他

标签

在一些Web页面中常常会添加一个标签用来告诉用户一些额外的信息，比如说在导航上添加了一个新导航项，可能就会加一个“new”标签，来告诉用户。这是新添加的导航项。在Bootstrap专门将这样的效果提取成为了单独的标签组件。可以使用 `` 这样的行内标签添加 `.label` 类来控制大小，再使用情景类来控制高亮显示的颜色如：`label-default`、`label-danger` 等等（在原来笔记中说到过这样的情景类，只不过这里需要加上不同的前缀）

`Default` `Primary` `Success` `Info` `Warning` `Danger`

徽章（角标）

其实和上面的标签效果很大部分是类似的，一般用作信息提示，例如剩余多少未读信息等等，也就是我们常常能在右上角看到的未读信息提示。在Bootstrap中，把这种效果称作为徽章效果，使用`badge`类来实现。

给链接、导航等元素嵌套 `` 元素，可以很醒目的展示新的或未读的信息条目。

```
<a href="#">Inbox <span class="badge">42</span></a>
<button class="btn btn-primary" type="button">
  Messages <span class="badge">4</span>
</button>
```

Inbox 42

Messages 4

通过查看源码，我们可以发现Bootstrap同样使用 `:empty` 伪元素，来设置当没有内容的时候隐藏：

```
.badge:empty {
  display: none;
}
```

需要注意的是IE8 不支持 `:empty` 选择符，所以不会默认隐藏；

Bootstrap 中的徽章会适配导航元素的激活状态。

4.3 组件

```
<ul class="nav nav-pills" role="tablist">
    <li role="presentation" class="active"><a href="#">Home <span class="badge">42</span></a></li>
    <li role="presentation"><a href="#">Profile</a></li>
    <li role="presentation"><a href="#">Messages <span class="badge">3</span></a></li>
</ul>
```



缩略图

其实在我们日常使用的网页中，随处都可以看到有缩略图的存在，比如淘宝的商品列表，慕课网上的课程列表等等，Bootstrap自然不会放过那么常见效果，它将这样的效果独立成为了一个模块组件，使用.thumbnail样式类，Bootstrap缩略图的默认设计仅需最少的标签就能展示带链接的图片。

我们需要配合Bootstrap的栅格系统来达到一个不错的显示效果。在Bootstrap中文网中，我们可以看到这样的示例代码：

```
<div class="container">
    <div class="row">
        <div class="col-xs-6 col-md-3">
            <a href="#" class="thumbnail">
                
                <!-- 此处应对图片做一些样式设置height: 180px; width: 100%; display: block; 等 -->
            </a>
        </div>
        ...
    </div>
</div>
```

我们将上面代码中的列复制3份，改变浏览器的大小，我可以看到因为栅格系统的存在，我们的缩略图已经变成了响应式的。这是栅格系统起到的效果那么.thumbnail又起到了什么效果呢？我们可以看到图片周围有了一圈边框，并且在边框与图片之间有着4px的padding，除此之外bootstrap还人性化的为我们的缩略图效果提供了hover效果。

添加标题、描述内容、按钮

4.3 组件

我们当然不可能仅仅只展示一张可点击的图片就完了，我们时常还会配上相关的说明文字以及一些按钮等等。在已有的缩略图的基础上，我们需要使用.caption容器，并且在这个容器中放置其它的内容，来达到统一的显示效果。

```
<!-- 添加到上面代码中的.thumbnail容器内, a标签后 -->
<div class="caption">
    <h2>说明文字</h2>
    <p>这里是说明文字</p>
    <button class="btn btn-info">按钮</button>
    <button class="btn btn-success">按钮</button>
</div>
```

看看效果吧：

缩略图展示

警示框

其实在我们的网页中，总是避免不了各种的提示信息，比如我们的登陆界面中，总是避免不了输入错误的情况，就需要使用警示框来提示用户错误信息。Bootstrap为我们提供了一组这样的样式类。

Bootstrap使用.alert来控制警示框的大小边距等样式，然后再用：成功：.alert-success、信息：.alert-info、警告：.alert-warning和错误：.alert-danger、这样的一组样式类，来表示不同的情景颜色。

```
<div class="alert alert-success" role="alert">恭喜您操作成功! </div>
<!-- ...等等 -->
```

缩略图展示

关闭

除了展示效果之外，如果你还引入了Bootstrap的js文件。那么我们还可以为我们的警示框非常简单的创造一些交互效果。

```
<div class="alert alert-success alert-dismissible">
    success操作成功
    <button class="close" data-dismiss="alert">
        <span class="glyphicon glyphicon-remove"></span>
    </button>
</div>
```

可以看到效果如下：（点击后面的icon便可以关闭该提示信息）

缩略图交互

4.3 组件

那么好的效果，在实现的时候需要注意一些什么呢？主要有三个步骤：

在普通的警示框.alert的基础上，添加.alert-dismissible类 在button标签中加入close类，实现警示框关闭按钮的样式（我在上面使用了Bootstrap自带的icon图标来实现关闭按钮）在关闭按钮上添加自定义属性data-dismiss="alert"（Bootstrap中是通过这样的自定义属性，再使用JS检测来实现关闭效果的） 警示框的链接

有时我警示框中除了错误信息之外，还会加入一个链接地址，以便于用户在新的页面中查找问题去处理他的错误。

Bootstrap仅仅使用了一个.alert-link的类就实现了这样的效果，主要是对于链接样式的高亮显示、加粗，并且颜色相应加深。（除此之外在hover上去后有下划线）

进度条

Bootstrap框架对于进度条提供了一个基本样式，一个100%宽度的背景色，然后一个子容器高亮，表示完成进度。其实制作这样进度条非常容易，一般使用两个容器，外容器具有一定的宽度，设置一个背景色，他的子元素设置一个宽度（父容器的宽度比例值如60%），同时设置一个高亮的背景色。

Bootstrap也是这样实现的，外层容器使用.progress类，子容器使用.progress-bar类（别忘记设置宽度）如：

```
<div class="progress">
    <div class="progress-bar" style="width:80%"></div>
</div>
```

同样它也提供了四个不同的情景颜色。.progress-bar-info..等，添加到子容器原有的类后即可

除了色彩，还可以使进度条的变为条纹的，只需要将progress-striped添加到原有的父容器类后，它同样具有彩色效果。更进一步的是，若在其后再添加一个.active类，可以实现动态的条纹进度条。（必须两个同时使用，使用css3动画实现）我们来试试：

```
<div class="progress progress-striped active">
    <div class="progress-bar progress-bar-danger" style="width:80%"></div>
</div>
<!-- . . . -->
```

条纹进度条

我们还可以在一个进度条内，将不同状态的进度条放置在一起，水平排列，形成叠加进度条。

4.3 组件

带标签的显示条 使用很简单，只需要在子容器中添加相应的数字就可以了，不过下面这段代码会有什么效果呢？不妨也自己去试试吧！(在源码中搜索aria-valuenow="0"可看到原因)

```
<div class="progress">
    <div class="progress-bar" role="progressbar" aria-valuenow="0" aria-valuemin="0"
        aria-valuemax="100">0%</div>
</div>
```

媒体对象

媒体对象一般包括以下几个部分：

- 媒体对像的容器：常使用.media类名表示，用来容纳媒体对象的所有内容
- 媒体对像的对象：常使用.media-object表示，就是媒体对象中的对象，常常是图片
- 媒体对象的主体：常使用.media-body表示，就是媒体对像中的主体内容，可以是任何元素，常常是图片侧边内容
- 媒体对象的标题：常使用.media-heading表示，就是用来描述对象的一个标题，此部分可选
- .pull-left或者.pull-right来控制媒体对象中的对象浮动方式。(如图片在左或者在右) 那么我们可以看到的结构如下：

```
<div class="media">
    <a class="pull-left" href="#">
        
    </a>
    <div class="media-body">
        <h4 class="media-heading">Bootstrap</h4>
        <div>默认样式的媒体对象组件允许在一个内容块的左边或右边展示一个多媒体内容（图像、视频、音频）。</div>
    </div>
</div>
```

可以实现最普通的图片在左，标题与介绍文字在右的媒体对象组。

我们在论坛中常常能看到和Bootstrap中文官网媒体对象组模块类似的阶梯状媒体对象，它是怎么实现的呢？其实很简单，我们只需要在上一级的.media-body容器最后添加新的额媒体对象容器.media就可以实现这样的阶梯效果

媒体对象

除了这样的阶梯形式，我们还可以在论坛评论系统中看到，评论是平铺着下来的，也就是媒体对象列表。

4.3 组件

我们只需要将原来的.media放在.media-list容器中就可以实现，不过既然都说了是列表，那我们还是直接使用ul>li的排列方式吧：

```
<ul class="media-list">
  <li class="media">...</li>
  <li class="media">...</li>
  <li class="media">...</li>
</ul>
```

Bootstrap同样帮我们清除了，媒体对象列表中的ul>li的默认样式，使展示更美观

列表组

列表组是Bootstrap框架新增的一个组件，可以用来制作列表清单、垂直导航等效果，也可以配合其他的组件制作出更漂亮的组件。它主要包括两个部分：

1. .list-group：列表组容器，常用的是ul元素，当然也可以是ol或者div元素
2. .list-group-item：列表项，常用的是li元素，当然也可以是div元素 如：

带徽章的列表组

带徽章的列表组其实就是将Bootstrap框架中的徽章组件和基础列表组结合在一起的一个效果。具体做法很简单，只需要在.list-group-item中添加徽章组件.badge：

```
<ul class="list-group">
  <li class="list-group-item">玩转Bootstrap <span class="badge">5</span></li>
  <li class="list-group-item">基础Bootstrap的网页开发 <span class="badge">12</span>
</li>
</ul>
```

可以实现效果如下：

带徽章的列表

如果在列表组中，我们希望列表项是链接的话，Bootstrap为我们提供了两种方案：

直接将链接置于li.list-group-item容器内，它会有链接下划线的存在 将ul.list-group>li.list-group-item的结构换成div.list-group>a.list-group-item，此时与默认的列表组显示无异（emmet会吧？那就能理解我的代码的意思...）

自定义列表组

除了上面默认的列表组之外，为了更清晰的展示内容，Bootstrap实现了自定义列表组，主要是新增了两个类：

4.3 组件

.list-group-item-heading：用来定义列表项头部样式 .list-group-item-text：用来定义列表项主要内容 结构如下编写：

```
<div class="list-group">
  <a href="#" class="list-group-item">
    <h4 class="list-group-item-heading">图解CSS3</h4>
    <p class="list-group-item-text">...</p>
  </a>
  <a href="#" class="list-group-item">
    <h4 class="list-group-item-heading">Sass中国</h4>
    <p class="list-group-item-text">...</p>
  </a>
</div>
```

列表项的状态设置

和其他的组件一样，列表组也两个状态效果：

.active：表示当前状态 .disabled：表示禁用状态 只需要在相应的列表项上添加。
class="list-group-item active">，主要对于该项的背景颜色进行高亮，以及其中的徽章进行了高亮处理

禁用状态，只是添加了一个灰色的背景色，以及文字的暗色处理

除了上面的两个最常用的状态之外，Bootstrap为我们的列表组也提供了不同的情景颜色，只需要在列表项中添加.list-group-item-danger这一系列的类。

面板

同样作为一个新增的组件，基础面板非常简单，就是一个div容器运用了.panel样式，产生一个具有边框的文本显示块。由于.panel不控制主题颜色，所以在.panel的基础上增加一个控制颜色的主题.panel-default，另外在里面添加了一个.div.panel-body来放置面板主体内容：

```
<div class="panel panel-default">
  <!-- panel-default就是那一系列的情景类，设置不同的可以设置不同的颜色 -->
  <div class="panel-body">我是一个基础面板，带有默认主题样式风格</div>
</div>
```

带有头和尾的面板

基础面板看上去太简单了，Bootstrap为了丰富面板的功能，特意为面板增加“面板头部”和“页面尾部”的效果：

- .panel-heading：用来设置面板头部样式

4.3 组件

- .panel-footer：用来设置面板尾部样式

```
<div class="panel panel-default">
    <!-- 这里对于不同的情景颜色，面板头的颜色改变十分明显 -->
    <div class="panel-heading">图解CSS3</div>
    <div class="panel-body">...</div>
    <div class="panel-footer">作者：大漠</div>
</div>
```

面板中嵌套表格

其实面板作为一个用于处理，别的组件完成不了的情况的组件，一般情况下可以把面板理解为一个区域，所以在使用面板的时候，都会在.panel-body放置需要的内容，可能是图片、表格或者列表等。

如果在面板中嵌套表格，只需要将编写好的表格直接放置在面板中即可，不过有着如下两种形式：

```
<div class="panel panel-default">
    <div class="panel-heading">图解CSS3</div>
    <div class="panel-body">...</div>
    <!-- 这里是放在panel-body后面，实际上我们还可以将table放在panel-body里面 -->
    <table class="table table-bordered">...</table>
    <div class="panel-footer">作者：大漠</div>
</div>
```

两个效果最大的差异是：

1. 放在.panel-body内的表格，实际上就是简单的嵌套，设置了一些padding值等。
2. 放在.panel-body后的表格，会与面板融合在一起，与面板相接的地方，不会有边框的展示，而是直接展示面板的边框

在面板中嵌套列表组时，同样是这样的两个差异

成果展示

1. 去挑选一个自己喜欢的Bootstrap 模板，根据先前的教程，重新在github建立仓库(repository)。以下免费Bootstrap模板供参考下载
 - [14 + Bootstrap Free Themes and Templates: Download @ Creative Tim](#)
 - [I AM X- Responsive Personal Portfolio vCard Template](#)
 - [Bootstrap resume themes on BootstrapZero](#)
 - [Bootstrap Themes at BootstrapZero](#)
 - [Free Bootstrap Templates | Brix.io](#)
 - [Responsive Bootstrap Themes & Templates - UX centred HTML5 Bootstrap themes and site templates made for developers and startups](#)
2. 修改成你想修改的样子
3. 与你的教练分享