

文档和网站结构

除了定义页面的各个部分（例如“段落”或“图像”）之外，[HTML](#)还拥有许多用于定义网站区域的块级元素（例如“标题”、“导航”菜单”，“主要内容栏”）。本文研究如何规划一个基本的网站结构，并编写 HTML 来表示该结构。

先决条件:	基本的 HTML 熟悉度，如 HTML 入门 中所述。 HTML 文本格式，如HTML 文本基础知识 所述。 超链接如何工作，如创建超链接 中所述。
客观的:	了解如何使用语义标签构建文档，以及如何构建简单网站的结构。

文档的基本部分

网页可以并且将看起来彼此完全不同，但它们都倾向于共享相似的标准组件，除非该页面正在显示全屏视频或游戏，是某种艺术项目的一部分，或者只是结构不当：

标头：

通常在顶部有一个大条带，上面有一个大标题、徽标，也许还有一条标语。这通常从一个网页到另一个网页保持不变。

导航栏：

网站主要部分的链接；通常由菜单按钮、链接或选项卡表示。与标题一样，此内容通常从一个网页到另一个网页保持一致——在您的网站上不一致的导航只会导致用户感到困惑、沮丧。许多网页设计师认为导航栏是标题的一部分，而不是一个单独的组件，但这不是必需的；事实上，有些人还争辩说，将两者分开对[可访问性](#)更好，因为屏幕阅读器如果分开可以更好地阅读这两个功能。

主要内容：

中心的一个大区域，包含给定网页的大部分独特内容，例如，您想要观看的视频，或者您正在阅读的主要故事，或者您想要查看的地图，或者新闻标题，等等。这是网站的一部分，肯定会因页面而异！

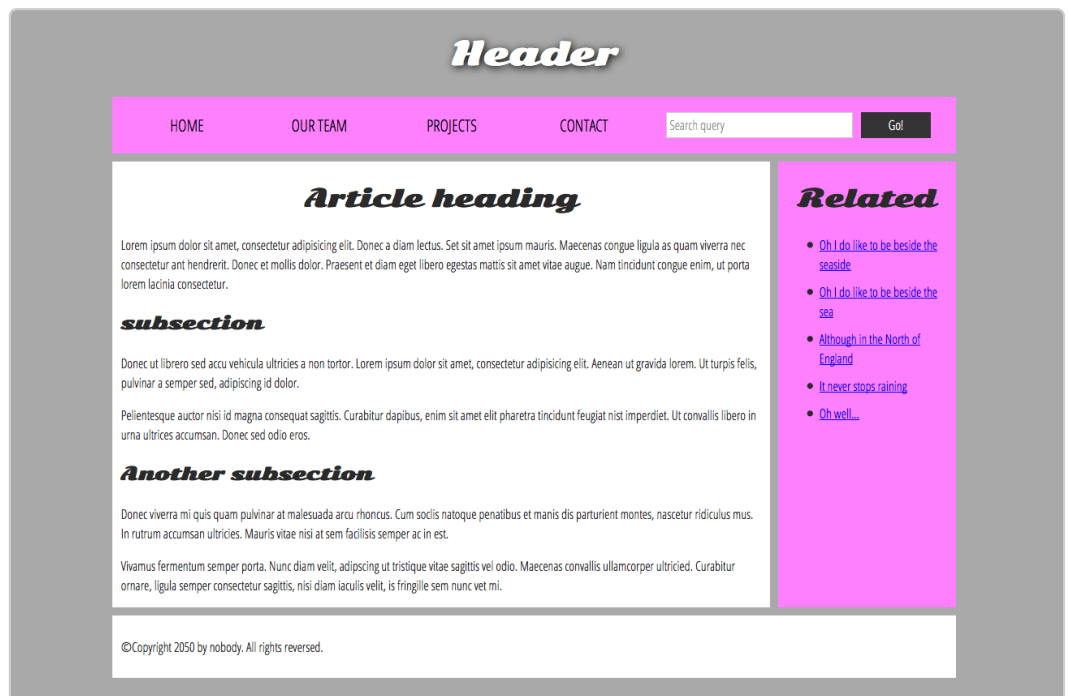
边栏：

一些外围信息、链接、引用、广告等。通常，这与主要内容中包含的内容相关（例如，在新闻文章页面上，侧边栏可能包含作者的简介或相关文章的链接），但也有在某些情况下，您还会发现一些重复出现的元素，例如辅助导航系统。

页脚：

页面底部的条带，通常包含细则、版权声明或联系信息。这是放置公共信息（如标题）的地方，但通常，该信息对于网站本身并不重要或次要。页脚有时也用于[SEO](#)目的，通过提供快速访问流行内容的链接。

一个“典型网站”的结构可能是这样的：



注意：上图说明了文档的主要部分，您可以使用 HTML 对其进行定义。但是，此处显示的页面外观（包括布局、颜色和字体）是通过将[CSS](#)应用于HTML 来实现的。

在本模块中，我们不教授 CSS，但一旦您了解了 HTML 的基础知识，请尝试深入了解我们的[CSS 第一步](#)模块，开始学习如何设计您的网站样式。

用于构建内容的 HTML

上面显示的简单示例并不漂亮，但它非常适合说明典型的网站布局示例。有些网站有更多的栏目，有些则复杂得多，但你明白了。使用正确的 CSS，您可以使用几乎所有元素来环绕不同的部分并使其看起来像您想要的那样，但如前所述，我们需要尊重语义并为正确的工作使用正确的元素。

这是因为视觉效果并不能说明整个故事。我们使用颜色和字体大小将视力正常的用户的注意力吸引到内容中最有用的部分，例如导航菜单和相关链接，但是对于视障人士来说，例如，他们可能找不到“粉红色”和“大号”等概念字体“很有用”。

注：[大约 8% 的男性和 0.5% 的女性](#) 是色盲；或者，换句话说，大约每 12 名男性中就有 1 名，每 200 名女性中就有 1 名。盲人和视障人士约占世界人口的 4-5%（2015 年有[9.4 亿人有一定程度的视力丧失](#)，而总人口约为[75 亿](#)）。

在您的 HTML 代码中，您可以根据内容的功能标记部分内容— 您可以使用明确表示上述内容部分的元素，屏幕阅读器等辅助技术可以识别这些元素并帮助完成诸如“找到主要内容”之类的任务导航，或“查找主要内容”。[正如我们在课程前面提到的，没有为正确的工作使用正确的元素结构和语义](#)会产生许多后果。

为实现此类语义标记，HTML 提供了可用于表示此类部分的专用标记，例如：

- 标头：[<header>](#)。
- 导航栏：[<nav>](#)。
- 主要内容：[<main>](#)，包含由[<article>](#)、[<section>](#)和[<div>](#)元素表示的各种内容小节。

- 边栏: [<aside>](#) ; 常放在里面 [<main>](#) 。
- 页脚: [<footer>](#) 。

主动学习：探索我们示例的代码

我们上面看到的示例由以下代码表示（您也可以[在我们的 GitHub 存储库中找到该示例](#)）。我们希望您查看上面的示例，然后查看下面的列表，看看哪些部分构成了视觉的哪些部分。

```
<!DOCTYPE html>
<html lang="en-US">
  <head>
    <meta charset="utf-8" />
    <meta name="viewport" content="width=device-width" />

    <title>My page title</title>
    <link
      href="https://fonts.googleapis.com/css?
family=Open+Sans+Condensed:300|Sonsie+One"
      rel="stylesheet" />
    <link rel="stylesheet" href="style.css" />
  </head>

  <body>
    <!-- Here is our main header that is used across all the
pages of our website -->

    <header>
      <h1>Header</h1>
    </header>

    <nav>
      <ul>
        <li><a href="#">Home</a></li>
        <li><a href="#">Our team</a></li>
        <li><a href="#">Projects</a></li>
        <li><a href="#">Contact</a></li>
      </ul>

      <!-- A Search form is another common non-linear way to
navigate through a website. -->
```

```
<form>
  <input type="search" name="q" placeholder="Search query"
/>

  <input type="submit" value="Go!" />
</form>
</nav>

<!-- Here is our page's main content -->
<main>
  <!-- It contains an article -->
  <article>
    <h2>Article heading</h2>

    <p>
      Lorem ipsum dolor sit amet, consectetur adipisicing
elit. Donec a diam
      lectus. Set sit amet ipsum mauris. Maecenas congue
ligula as quam
      viverra nec consectetur ant hendrerit. Donec et mollis
dolor. Praesent
      et diam eget libero egestas mattis sit amet vitae
augue. Nam tincidunt
      congue enim, ut porta lorem lacinia consectetur.
    </p>

    <h3>Subsection</h3>

    <p>
      Donec ut libero sed accu vehicula ultricies a non
tortor. Lorem ipsum
      dolor sit amet, consectetur adipisicing elit. Aenean
ut gravida lorem.
      Ut turpis felis, pulvinar a semper sed, adipiscing id
dolor.
    </p>

    <p>
      Pelientesque auctor nisi id magna consequat sagittis.
Curabitur
      dapibus, enim sit amet elit pharetra tincidunt feugiat
nisi imperdiet.
      Ut convallis libero in urna ultrices accumsan. Donec
sed odio eros.
    </p>
```

```
<h3>Another subsection</h3>

<p>
    Donec viverra mi quis quam pulvinar at malesuada arcu
    rhoncus. Cum
        soclis natoque penatibus et manis dis parturient
    montes, nascetur
        ridiculus mus. In rutrum accumsan ultricies. Mauris
    vitae nisi at sem
        facilisis semper ac in est.
</p>

<p>
    Vivamus fermentum semper porta. Nunc diam velit,
    adipiscing ut
        tristique vitae sagittis vel odio. Maecenas convallis
    ullamcorper
        ultricies. Curabitur ornare, ligula semper consectetur
    sagittis, nisi
        diam iaculis velit, is fringille sem nunc vet mi.
</p>
</article>

<!-- the aside content can also be nested within the main
content -->
<aside>
    <h2>Related</h2>

    <ul>
        <li><a href="#">Oh I do like to be beside the
    seaside</a></li>
        <li><a href="#">Oh I do like to be beside the sea</a>
    </li>
        <li><a href="#">Although in the North of England</a>
    </li>
        <li><a href="#">It never stops raining</a></li>
        <li><a href="#">Oh well...</a></li>
    </ul>
</aside>
</main>

<!-- And here is our main footer that is used across all the
pages of our website -->
```

```
<footer>
  <p>©Copyright 2050 by nobody. All rights reversed.</p>
</footer>
</body>
</html>
```

花一些时间查看代码并理解它——代码中的注释也应该有助于您理解它。在本文中我们并没有要求您做太多其他事情，因为理解文档布局的关键是编写合理的 HTML 结构，然后使用 CSS 对其进行布局。我们将等到您开始将 CSS 布局作为 CSS 主题的一部分进行研究。

更详细的 HTML 布局元素

详细了解所有 HTML 分段元素的总体含义是很好的——随着您开始获得更多 Web 开发经验，您将逐渐了解这一点。[您可以通过阅读我们的HTML元素参考](#)找到很多详细信息。现在，这些是您应该尝试理解的主要定义：

- [<main>](#) 适用于此页面独有的内容。<main> 每页只用一次，直接放在里面 [<body>](#)。理想情况下，这不应嵌套在其他元素中。
- [<article>](#) 包含一块相关内容，这些内容本身有意义，没有页面的其余部分（例如，一篇博客文章）。
- [<section>](#) 类似于 <article>，但它更多的是将页面的单个部分组合在一起，构成一个单一的功能（例如，一个迷你地图，或一组文章标题和摘要），或一个主题。[每个部分都以标题](#)开头被认为是最佳做法；另请注意，您可以根据上下文将 <article> s 分解为不同的 <section> s，或 <section> 将 s 分解为不同的s。<article>
- [<aside>](#) 包含与主要内容没有直接关系但可以提供与其间接相关的附加信息（词汇表条目、作者传记、相关链接等）的内容。
- [<header>](#) 代表一组介绍内容。如果它是它的子项 [<body>](#)，则它定义网页的全局标题，但如果它是 [or](#) 的子项，则 [<article>](#) 它 [<section>](#) 为该部分定义特定的标题（尽量不要将其与[titles 和 headings](#)混淆）。
- [<nav>](#) 包含页面的主要导航功能。辅助链接等不会出现在导航中。
- [<footer>](#) 表示页面的一组结束内容。

可以单击上述每个元素以阅读“HTML 元素参考”部分中的相应文章，提供有关每个元素的更多详细信息。

非语义包装器

有时您会遇到无法找到理想的语义元素来将某些项目组合在一起或包装某些内容的情况。有时您可能只想将一组元素组合在一起，以通过一些[CSS](#)或[JavaScript](#)将它们作为单个实体来影响它们。对于这些情况，HTML 提供了 [<div>](#) 和 [](#) 元素。您最好将它们与合适的 [class](#) 属性一起使用，为它们提供某种标签，以便轻松定位它们。

[](#) 是一个内联非语义元素，只有在您想不出更好的语义文本元素来包装您的内容，或者不想添加任何特定含义时才应使用它。例如：

```
<p>
  The King walked drunkenly back to his room at 01:00, the beer
  doing nothing to
  aid him as he staggered through the door
  <span class="editor-note">
    [Editor's note: At this point in the play, the lights should
    be down low]
  </span>.
</p>
```

在这种情况下，编者按应该只是为该剧的导演提供额外的指导；它不应该有额外的语义。对于有视力的用户，可能会使用 CSS 将注释与正文稍微拉开距离。

[<div>](#) 是一个块级非语义元素，只有当你想不出更好的语义块元素来使用，或者不想添加任何特定含义时，才应该使用它。例如，假设有一个购物车小部件，您可以在访问电子商务网站期间随时选择将其拉出：

```
<div class="shopping-cart">
  <h2>Shopping cart</h2>
  <ul>
    <li>
      <p>
        <a href=""><strong>Silver earrings</strong></a>:
        $99.95.
```



```
</p>

</li>
<li>...</li>
</ul>
<p>Total cost: $237.89</p>
</div>
```

这不是真正的 `<aside>`，因为它不一定与页面的主要内容相关（您希望它可以从任何地方查看）。它甚至没有特别保证使用 `<section>`，因为它不是页面主要内容的一部分。所以 `<div>` 在这种情况下 a 很好。我们包含一个标题作为路标，以帮助屏幕阅读器用户找到它。

警告： Div 使用起来非常方便，很容易用得太多。由于它们没有语义价值，它们只会使您的 HTML 代码混乱。请注意只有在没有更好的语义解决方案时才使用它们，并尽量将它们的使用减少到最低限度，否则您将很难更新和维护您的文档。

换行符和水平规则

您偶尔会使用并想了解的两个元素是 `
` 和 `<hr>`。

`
`：换行元素

`
` 在段落中创建换行符；在您想要一系列固定的短行的情况下（例如在邮政地址或诗歌中），这是强制使用刚性结构的唯一方法。例如：

```
<p>
  There once was a man named O'Dell<br />
  Who loved to write HTML<br />
  But his structure was bad, his semantics were sad<br />
  and his markup didn't read very well.
</p>
```

如果没有这些 `
` 元素，该段落将只呈现在一长行中（正如我们在课程前面所说的，[HTML 会忽略大部分空格](#)）；使用 `
` 代码中的元素，标记呈现如下：

There once was a man named O'Dell
Who loved to write HTML
But his structure was bad, his semantics were sad
and his markup didn't read very well.

<hr>：主题中断元素

<hr> 元素在文档中创建一条水平线，表示文本中的主题变化（例如主题或场景的变化）。从视觉上看，它只是一条水平线。举个例子：

```
<p>  
    Ron was backed into a corner by the marauding netherbeasts.  
    Scared, but  
    determined to protect his friends, he raised his wand and  
    prepared to do  
    battle, hoping that his distress call had made it through.  
</p>  
<hr />  
<p>  
    Meanwhile, Harry was sitting at home, staring at his royalty  
    statement and  
    pondering when the next spin off series would come out, when  
    an enchanted  
    distress letter flew through his window and landed in his lap.  
    He read it  
    hazily and sighed; "better get back to work then", he mused.  
</p>
```

会像这样渲染：

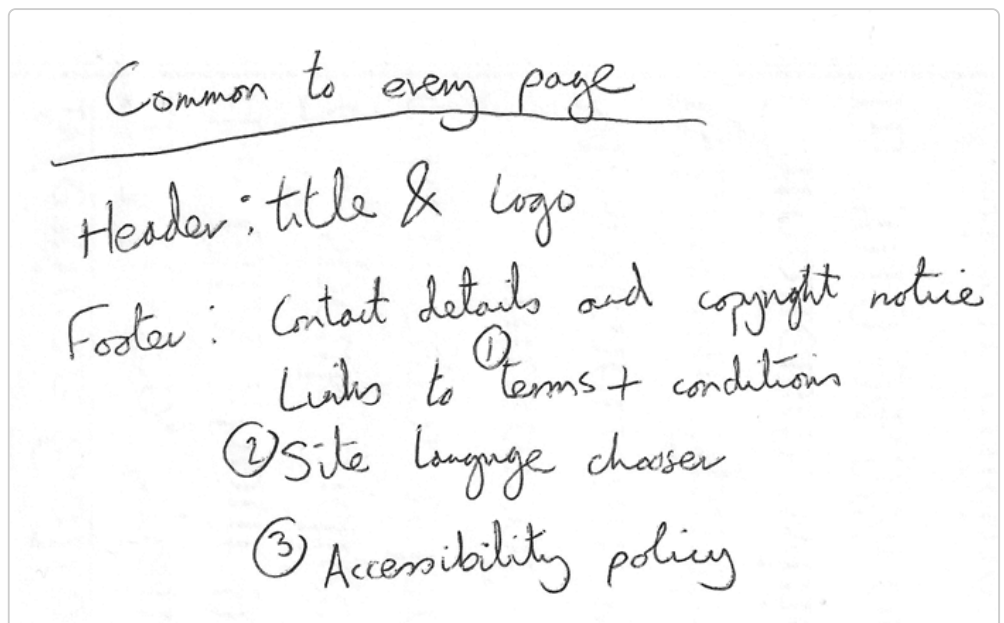
Ron was backed into a corner by the marauding netherbeasts. Scared, but determined to protect his friends, he raised his wand and prepared to do battle, hoping that his distress call had made it through.

Meanwhile, Harry was sitting at home, staring at

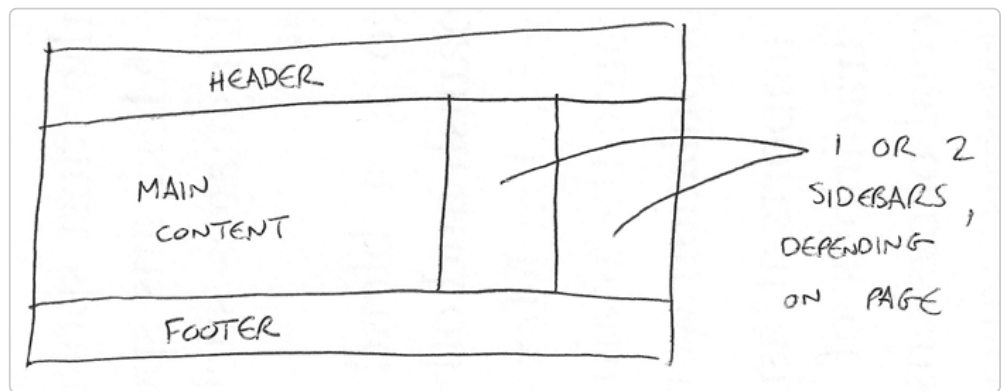
规划一个简单的网站

一旦你计划好了一个简单网页的结构，下一步就是尝试弄清楚你想在整个网站上放什么内容，你需要什么页面，以及它们应该如何排列和相互链接以获得最佳的用户体验。这称为[信息架构](#)。在一个大型、复杂的网站中，这个过程中可以进行很多规划，但对于一个只有几页的简单网站，这可能相当简单，也很有趣！

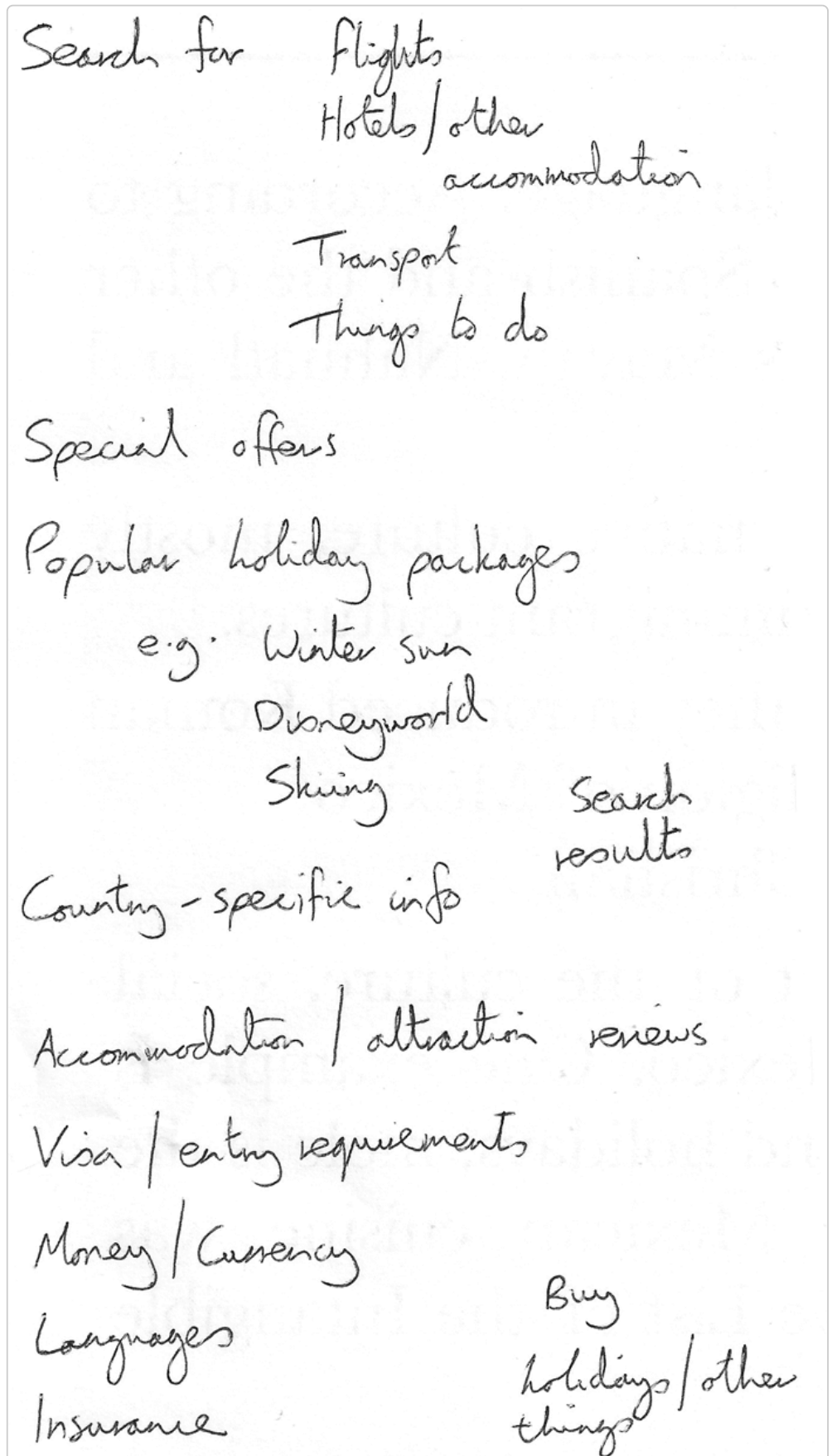
1. 请记住，您将拥有大多数（如果不是全部）页面共有的一些元素，例如导航菜单和页脚内容。例如，如果您的站点用于商业用途，最好在每页的页脚中提供您的联系信息。记下您希望每个页面都有的共同点。



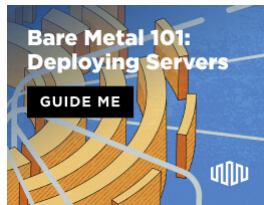
2. 接下来，画出您可能希望每个页面的结构看起来像什么的粗略草图（它可能看起来像我们上面的简单网站）。注意每个块将是什么。



3. 现在，集思广益您想要在您的网站上拥有的所有其他（并非每个页面都通用）内容 - 写下一个大列表。



4. 接下来，尝试将所有这些内容项分组，让您了解哪些部分可能一起出现在不同的页面上。[这与称为卡片分类的技术非常相似。](#)



有关如何使用
Equinix Metal 部署裸
机服务器的简单分步
指南

[Mozilla 广告](#)

不想看广告?

Search

- Flights
- Hotels
- Other accommodation
- Transport
- Things to do

Specials

- Special offers
- Popular holidays

Search results

Also include search functionality here to refine results

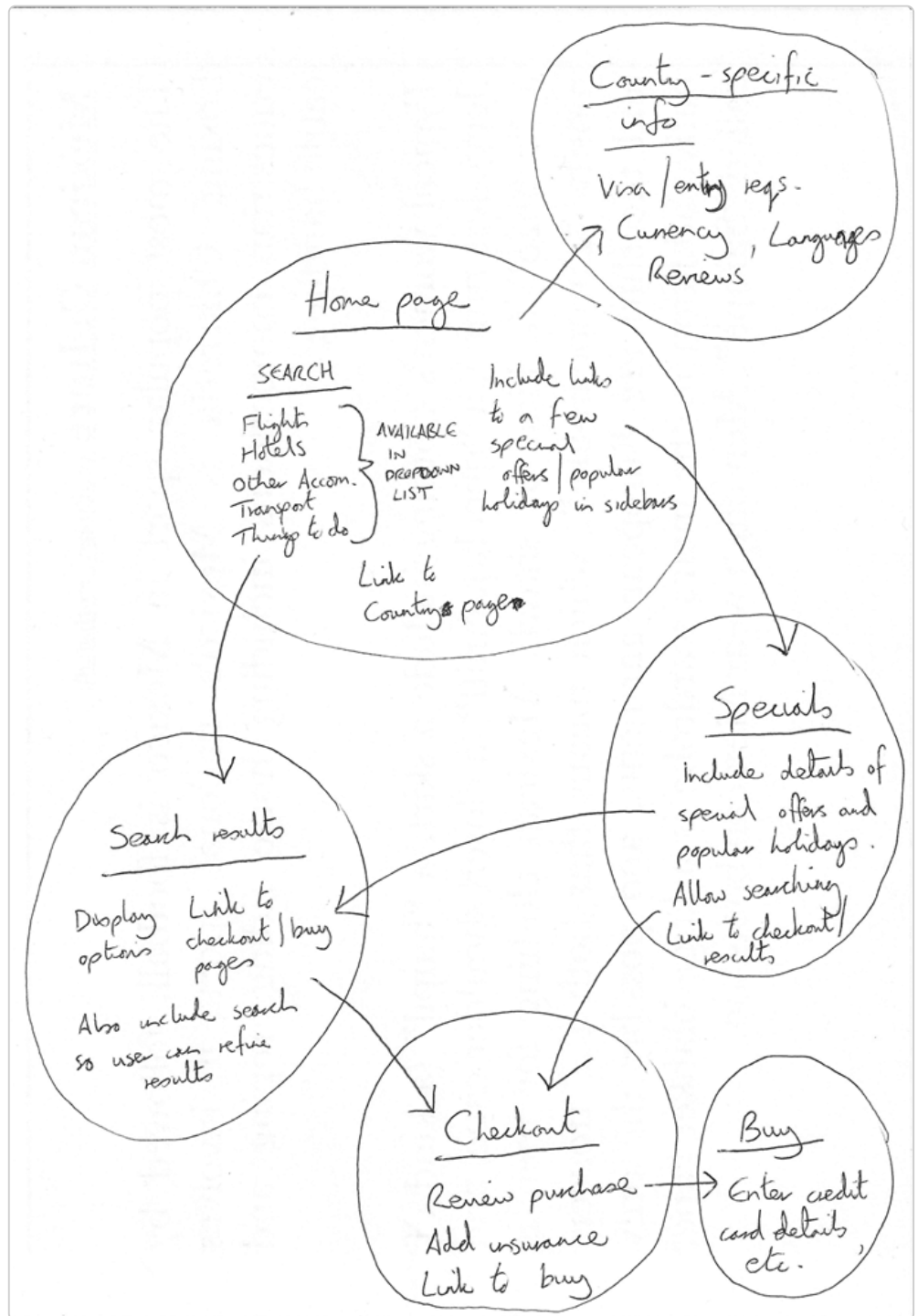
Country-specific info

- Reviews
- Visa / entry requirements
- Money / currency
- Languages

Buy things

- Shopping cart
- Checkout
- Insurance

5. 现在尝试绘制一个粗略的站点地图——为站点上的每个页面画一个气泡，并绘制线条以显示页面之间的典型工作流程。主页可能位于中心，并链接到大多数（如果不是全部）其他页面；小型站点中的大部分页面都应该可以从主导航栏访问，但也有例外。您可能还想包括有关如何呈现事物的注释。



主动学习：创建自己的站点地图

尝试为您自己创建的网站执行上述练习。你想做一个关于什么的网站？

注意： 将您的工作保存在某处；你以后可能需要它。

概括

此时，您应该对如何构建网页/站点有了更好的了解。在本模块的最后一篇文章中，我们将学习如何[调试 HTML](#)。

此页面最后修改于 2023 年 3 月 21 日由[MDN 贡献者](#)提供。