

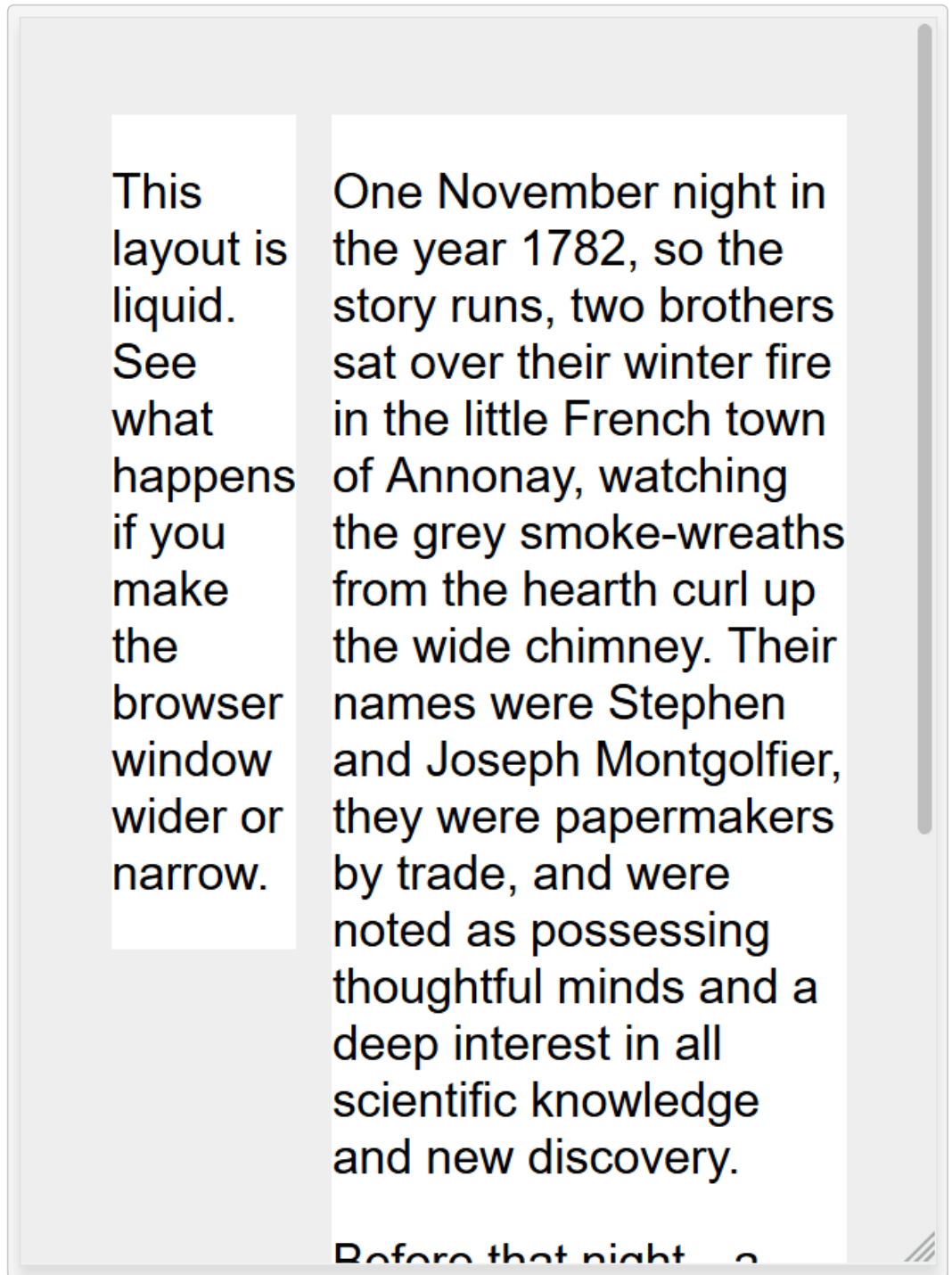
响应式设计

响应式网页设计(RWD) 是一种网页设计方法，可使网页在所有屏幕尺寸和分辨率上都能很好地呈现，同时确保良好的可用性。这是为多设备网络设计的方式。在本文中，我们将帮助您了解一些可用于掌握它的技术。

先决条件：	HTML 基础知识（研究 HTML 简介 ），以及 CSS 工作原理的概念（研究 CSS 第一步 和 CSS 构建块 。）
客观的：	了解用于实现响应式设计的基本目的和 CSS 功能。

HTML 从根本上说是响应式的，或者说*是流畅的*。如果您创建一个仅包含 HTML 而没有 CSS 的网页，并调整窗口大小，浏览器将自动重排文本以适应视口。

虽然默认的响应行为听起来似乎不需要解决方案，但在宽屏显示器上全屏显示的长行文本可能难以阅读。如果使用 CSS 减少了宽屏幕行的长度，例如通过创建列或添加大量填充，对于缩小浏览器窗口或在移动设备上打开网站的用户来说，该网站可能看起来被压扁了。



通过设置固定宽度来创建不可调整大小的网页也不起作用；这会导致在窄设备上出现滚动条，在宽屏幕上出现过多的空白区域。

响应式网页设计或 RWD 是一种解决设备和设备尺寸范围的设计方法，可以自动适应屏幕，无论是在平板电脑、手机、电视还是手表上查看内容。

响应式网页设计不是一项独立的技术——它是一种方法。它是一个术语，用于描述用于创建布局的一组最佳实践，该布局可以响应用于查看内容的任何设备。

术语[响应式设计](#)由 [Ethan Marcotte 在 2010 年创造](#)，描述了使用流体网格、流体图像和媒体查询来创建响应式内容，如 Zoe Mickley Gillenwater 的书[灵活网页设计](#)中所讨论的。

当时推荐的是 `float` 布局用 CSS，媒体查询查询浏览器宽度，创建不同断点的布局。流体图像设置为不超过其容器的宽度；他们将他们的 `max-width` 财产设置为 `100%`。流体图像在其包含的列变窄时按比例缩小，但在列增长时不会增长到大于其固有尺寸。这使图像能够缩小以适合其内容，而不是溢出它，但如果容器变得比图像更宽，也不会变大并变得像素化。

现代 CSS 布局方法本质上是响应式的，并且自 Gillenwater 的书和 Marcotte 的文章出版以来，我们在 Web 平台中内置了大量功能，使设计响应式网站变得更加容易。

本文的其余部分将向您介绍在创建响应式网站时可能希望使用的各种 Web 平台功能。

媒体查询

[媒体查询](#)允许我们运行一系列测试（例如，用户的屏幕是否大于特定宽度或特定分辨率）并有选择地应用 CSS 以根据用户需要适当地设置页面样式。

例如，以下媒体查询测试当前网页是否显示为屏幕媒体（因此不是打印文档）并且视口至少是 `80rem` 宽的。选择器的 CSS `.container` 只有在满足这两个条件时才会应用。

```
@media screen and (min-width: 80rem) {  
  .container {  
    margin: 1em 2em;  
  }  
}
```

您可以在样式表中添加多个媒体查询，调整整个布局或其中的一部分以最适合各种屏幕尺寸。引入媒体查询和更改布局的点称为[断点](#)。

使用媒体查询时的一种常见方法是为窄屏设备（例如手机）创建一个简单的单列布局，然后检查更宽的屏幕并在您知道有足够的屏幕宽度可以处理时实

施多列布局它。移动优先设计被称为**移动优先**设计。

[如果使用断点，最佳实践鼓励使用相对单位](#)而不是单个设备的绝对大小来定义媒体查询断点。

媒体查询块中定义的样式有不同的方法；从使用媒体查询到 [<link>](#) 基于浏览器大小范围的样式表，再到仅包括自定义属性变量以存储与每个断点关联的值。

[在媒体查询](#)的 MDN 文档中找到更多信息。

媒体查询可以帮助 RWD，但不是必需的。无需查询即可使用灵活的网格、相对单位以及最小和最大单位值。

响应式布局技术

响应式网站建立在灵活的网格上，这意味着您无需使用像素完美布局来针对所有可能的设备尺寸。

通过使用灵活的网格，您可以更改功能或添加断点，并在内容开始看起来很差的地方更改设计。例如，为了确保行长度不会随着屏幕尺寸的增加而变得不可读，您可以使用 [columns](#)；如果一个框在变窄时每行有两个单词被压扁，您可以设置一个断点。

多种布局方法，包括[Multiple-column layout](#)、[Flexbox](#)和[Grid](#)在默认情况下都是响应式的。他们都假设您正在尝试创建一个灵活的网格，并为您提供更简单的方法。

多元醇

使用 multicol，您可以指定 a column-count 来指示您希望将内容拆分成的最大列数。然后浏览器会计算出这些的大小，该大小会根据屏幕大小而变化。

```
.container {  
  column-count: 3;  
}
```

如果您改为指定 `a column-width`，则您指定的是**最小**宽度。浏览器将创建尽可能多的该宽度的列，以便轻松地放入容器中，然后在所有列之间共享剩余空间。因此，列数将根据空间大小而变化。

```
.container {  
  column-width: 10em;  
}
```

您可以使用 [columns](#) 简写来提供最大列数和最小列宽。这可以确保行长度不会随着屏幕尺寸的增加而变得难以阅读，或者随着屏幕尺寸的减小而变得太窄。

弹性盒

在 Flexbox 中，弹性项目收缩或增长，根据容器中的空间在项目之间分配空间。`flex-grow` 通过更改 `flex-grow` 的值，`flex-shrink` 您可以指示项目在遇到周围更多或更少空间时的行为方式。

在下面的示例中，每个弹性项目将在弹性容器中占据相同的空间，使用 `flex: 1` 布局主题[Flexbox 中描述的简写：弹性项目的灵活大小](#)。

```
.container {  
  display: flex;  
}  
  
.item {  
  flex: 1;  
}
```

注意：作为示例，我们在上面使用 flexbox 构建了一个简单的响应式布局。我们使用断点在屏幕变大时切换到多个列，并限制主要内容的大小 `max-width`： [示例](#) ， [源代码](#) 。

CSS 网格

在 CSS 网格布局中，该 `fr` 单元允许跨网格轨道分配可用空间。下一个示例创建一个网格容器，其中包含三个大小为 `1fr` 的轨道。这将创建三个列

轨道，每个轨道占用容器中可用空间的一部分。[您可以在使用 fr 单元的灵活网格](#)下的学习布局网格主题中找到有关此创建网格方法的更多信息。

```
.container {  
  display: grid;  
  grid-template-columns: 1fr 1fr 1fr;  
}
```

注意：网格布局版本更简单，因为我们可以直接在 `.wrapper` 上定义列：[示例](#)，[源代码](#)。

响应图像

为确保媒体永远不会大于其响应容器，可以使用以下方法：

```
img,  
picture,  
video {  
  max-width: 100%;  
}
```

这会缩放媒体以确保它们永远不会溢出容器。使用单个大图像并将其缩小以适合小型设备会下载比需要的更大的图像，从而浪费带宽。

响应式图像，使用 [<picture>](#) 元素和属性可以提供针对用户视口和设备分辨率的图像。例如，您可以为移动设备添加方形图片，但在桌面设备上显示与横向图片相同的场景。[srcset sizes](#)

该 `<picture>` 元素可以提供多种尺寸以及“提示”（描述图像最适合的屏幕尺寸和分辨率的元数据），浏览器将为每个设备选择最合适的图像，确保用户下载图像尺寸适用于他们正在使用的设备。使用 `<picture>` along with `max-width` 消除了使用媒体查询调整图像大小的需要。它可以将具有不同宽高比的图像定位到不同的视口尺寸。

您还可以使用不同尺寸的艺术直接图像，从而为不同的屏幕尺寸提供不同的裁剪或完全不同的图像。

[您可以在 MDN 上的学习 HTML 部分](#)找到有关响应式图像的详细指南。

响应式排版

响应式排版描述了在媒体查询中更改字体大小或使用视口单位来反映更少或更多的屏幕空间。

使用媒体查询进行响应式排版

在此示例中，我们希望将 1 级标题设置为 4rem，这意味着它将是我们基本字体大小的四倍。这是一个非常大的标题！我们只希望在更大的屏幕尺寸上显示这个巨型标题，因此我们首先创建一个较小的标题，然后如果我们知道用户的屏幕尺寸至少为 1200px。

```
html {  
  font-size: 1em;  
}  
  
h1 {  
  font-size: 2rem;  
}  
  
@media (min-width: 1200px) {  
  h1 {  
    font-size: 4rem;  
  }  
}
```

我们已经编辑了上面的响应式网格示例，以使用概述的方法也包括响应式类型。您可以看到标题如何随着布局变为两列版本而切换大小。

在手机上标题较小：

Watch my size!

This layout is responsive. See what happens if you make the browser window wider or narrow.

One November night in the year 1782, so the story runs, two brothers sat over their winter fire in the little French town of Annonay, watching the grey smoke-wreaths from the hearth curl up the wide chimney. Their names were Stephen and Joseph Montgolfier, they were papermakers by trade, and were noted as possessing thoughtful minds and a deep interest in all scientific knowledge and new discovery.

Before that night—a memorable night,

然而，在桌面上，我们看到更大的标题尺寸：

Watch my size!

This layout is responsive. See what happens if you make the browser window wider or narrow.

One November night in the year 1782, so the story runs, two brothers sat over their winter fire in the little French town of Annonay, watching the grey smoke-wreaths from the hearth curl up the wide chimney. Their names were Stephen and Joseph Montgolfier, they were papermakers by trade, and were noted as possessing thoughtful minds and a deep interest in all scientific knowledge and new discovery.

Before that night—a memorable night, as it was to prove—hundreds of millions of people had watched the rising smoke-wreaths of their fires without drawing any special inspiration from the fact.”

注意： 查看此示例的实际操作：[示例](#) ， [源代码](#) 。

正如这种排版方法所示，您无需将媒体查询限制为仅更改页面布局。它们可用于调整任何元素，使其在其他屏幕尺寸下更有用或更具吸引力。

使用视口单位进行响应式排版

视口单元 `vw` 还可用于启用响应式排版，而无需使用媒体查询设置断点。

`1vw` 等于视口宽度的百分之一，这意味着如果您使用 `设置字体大小 vw`，它将始终与视口的大小相关。

```
h1 {  
  font-size: 6vw;  
}
```

执行上述操作的问题是用户失去了使用该单位缩放任何文本集的能力 `vw`，因为该文本始终与视口的大小相关。**因此，永远不要单独使用视口单位来设置文本。**

有一个解决方案，它涉及使用 [calc\(\)](#)。如果您将 `vw` 单位添加到使用固定大小（例如 `em s` 或 `rem s`）的值集，那么文本仍然可以缩放。本质上，该 `vw` 单位在该缩放值之上添加：

```
h1 {  
  font-size: calc(1.5rem + 3vw);  
}
```

这意味着我们只需要为标题指定一次字体大小，而不是为移动设备设置它并在媒体查询中重新定义它。随着视口尺寸的增加，字体会逐渐变大。

注意： 请参阅实际示例：[示例](#) ， [源代码](#) 。

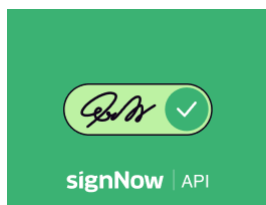
视口元标记

如果您查看响应式页面的 HTML 源代码，您通常会 [<meta>](#) 在 <head> 文档的 中看到以下标记。

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

这个[视口](#)元标记告诉移动浏览器，他们应该将视口的宽度设置为设备宽度，并将文档缩放到其预期大小的 100%，这会以您想要的移动优化大小显示文档。

为什么需要这个？因为移动浏览器倾向于谎报其视口宽度。



借助电子签名的力量
为您的应用程序增色
不少。免费试用
signNow API。

[Mozilla 广告](#)

不想看广告？

设备进行优化。因此，移动浏览器会将视口宽度设置为 980 像素，以该宽度呈现页面，并将结果显示为桌面布局的缩小版本。用户可以放大并在网站周围平移以查看他们感兴趣的部分，但看起来很糟糕。

通过设置， `width=device-width` 您将使用设备的 `width=980px` 实际宽度覆盖移动设备的默认值，例如 Apple 的默认值。没有它，带有断点和媒体查询的响应式设计可能无法在移动浏览器上正常工作。如果您的窄屏布局以 480 像素或更小的视口宽度启动，但设备显示其宽度为 980 像素，则该用户将看不到您的窄屏布局。

因此，您应该始终在文档的头部包含视口元标记。

概括

响应式设计是指响应其查看环境的站点或应用程序设计。它包含许多 CSS 和 HTML 特性和技术，现在基本上就是我们默认构建网站的方式。考虑一下您在手机上访问的网站 - 遇到桌面版本缩小的网站，或者您需要横向滚动才能找到内容的网站，这可能是相当不寻常的。这是因为网络已经转向这种响应式设计方法。

借助您在这些课程中学到的布局方法，实现响应式设计也变得更加容易。如果您现在是 web 开发的新手，那么与响应式设计的早期相比，您可以使用更多的工具。因此，值得检查您正在使用的任何材料的使用年限。虽然历史

文章仍然有用，但现代使用 CSS 和 HTML 可以更轻松地创建优雅和有用的设计，无论您的访问者使用什么设备查看网站。

也可以看看

- [媒体查询的 CSS 技巧指南](#)

此页面最后修改于 2023 年 3 月 22 日由[MDN 贡献者](#)提供。