

第七篇 响应网页开发

Contents

01.媒体查询

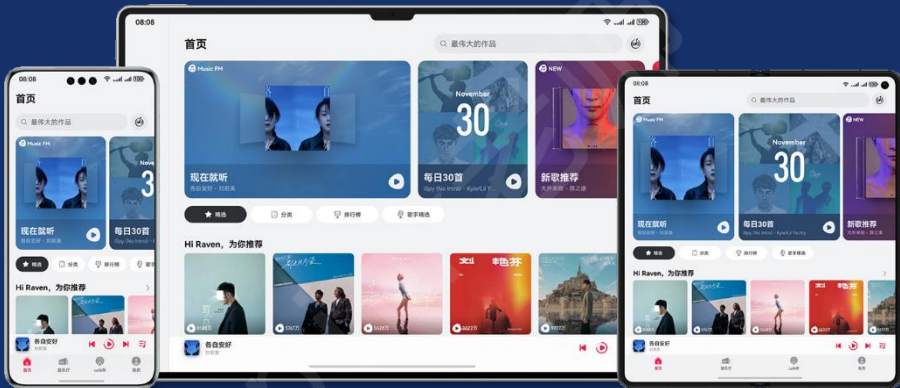
transform

02.Bootstrap响应式

case

响应式布局

响应式布局是一种网页设计方法，使网页能够根据用户设备的屏幕尺寸、分辨率、方向（横屏/竖屏）等特性自动调整布局和内容，提供一致且优化的用户体验。其核心目标是“一次设计，多端适配”。



优点	缺点
1. 一套代码适配多端设备	1. 复杂布局需较多调试
2. 节省开发和维护成本	2. 过度使用媒体查询可能影响性能
3. 用户体验一致性高	3. 需精准计算断点(什么屏幕下，怎么显示)

响应式布局

响应式布局核心技术：

布局方式：

- ❑ 弹性盒子 (Flexbox)
- ❑ 网格布局 (CSS Grid)
- ❑ 百分比布局(流式布局)
- ❑ 单位可以借助于vw实现

断点 (Breakpoints) 设置：

- ❑ 根据常见设备宽度设置断点
- ❑ 手机 < 768px
- ❑ 平板 768px~1024px
- ❑ PC > 1024px
- ❑ 根据不同设备宽度显示不同布局样式

断点名称	最小宽度	CSS 媒体查询	适用场景
xs	<576px	@media (min-width: 0)	超小设备 (手机竖屏)
sm	≥576px	@media (min-width: 576px)	平板竖屏/小屏手机横屏
md	≥768px	@media (min-width: 768px)	平板横屏/桌面小屏
lg	≥992px	@media (min-width: 992px)	桌面中屏
xl	≥1200px	@media (min-width: 1200px)	桌面大屏
xxl	≥1400px	@media (min-width: 1400px)	超宽屏 (4K 显示器等)

响应式布局

响应式布局实现方案：

1. 媒体查询：

媒体查询是 **CSS3** 的功能，允许根据设备的特性（如屏幕宽度、高度、分辨率、设备类型等）应用不同的样式规则。它是实现响应式设计的核心技术。

2. 第三方框架 Bootstrap：

预置样式和组件减少重复编码，缩短项目周期，适配 PC、移动端及主流浏览器。
后期还会有不同框架学习。

响应式布局

1. 媒体查询(Media Queries):

媒体查询是CSS3中的一个强大功能，它允许开发者根据设备的特定特性（如屏幕宽度、高度、方向、分辨率等）来应用不同的CSS样式。媒体查询是实现响应式网页设计（Responsive Web Design）的核心技术。

语法:

```
@media 媒体类型 and (媒体特征) {  
    /* 样式 */  
}
```

- ❑ @media : 媒体查询的起始声明
- ❑ 设备类型: 媒体类型（可选），如 screen, print等，省略默认 all
- ❑ 设备特性: 如 min-width, max-width 等

```
html {  
    font-size: 20px;  
    font-size: 5.33333vw  
}
```

正常文字大小

```
@media screen and (max-width: 320px) {  
    html {  
        font-size: 17.06667px  
    }  
}
```

当前屏幕最大是320px时候，
html文字修改17.06667px

```
@media screen and (min-width: 540px) {  
    html {  
        font-size: 28.8px  
    }  
}
```

当前屏幕最小是540px时
候，html文字修改28.8px

响应式布局

1. 任务一

当屏幕尺寸大于等于1200 像素， 页面背景显示为 绿色

当屏幕尺寸在 768像素到1200像素， 页面显示为 橙色

当屏幕尺寸小于 768 像素， 页面显示为 红色

响应式布局

2. 任务二：仿京东响应式布局

当屏幕尺寸大于等于1536 像素，一行放 6个 盒子

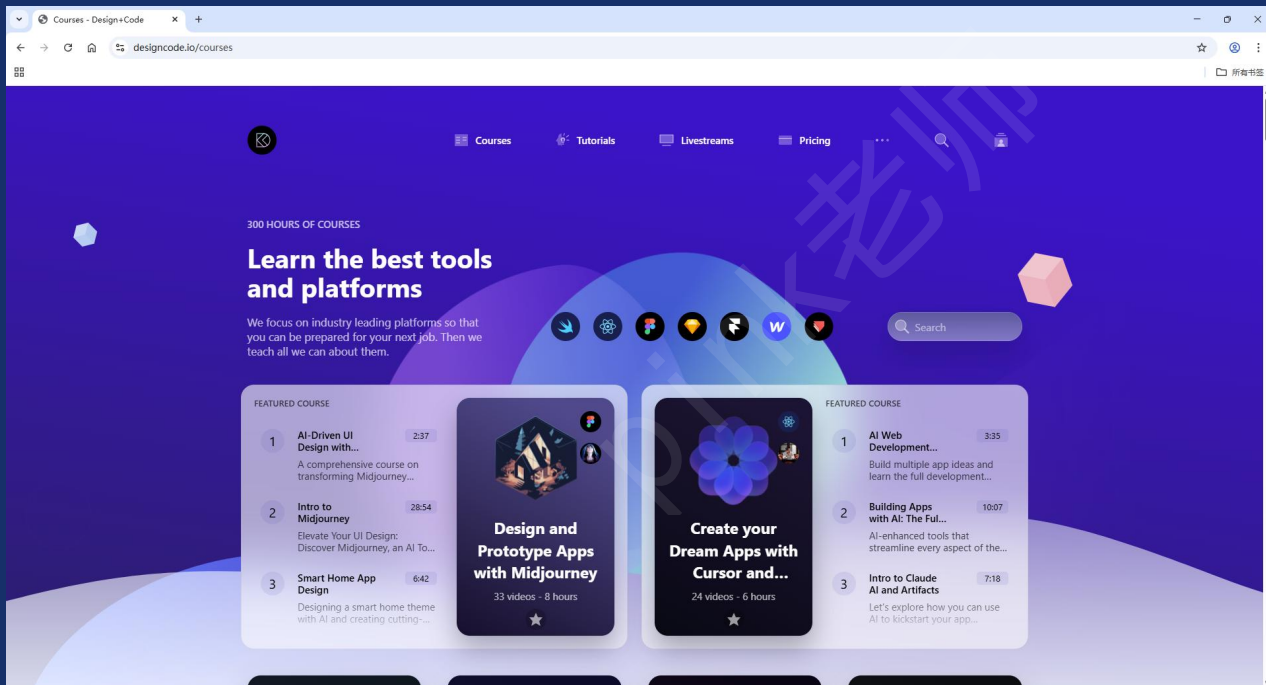
当屏幕尺寸在 1316像素到 1536像素，一行放 5个 盒子

当屏幕尺寸小于 1316 像素，一行放 4 个 盒子

父盒子最小尺寸是 1252 像素，最大尺寸是1780像素

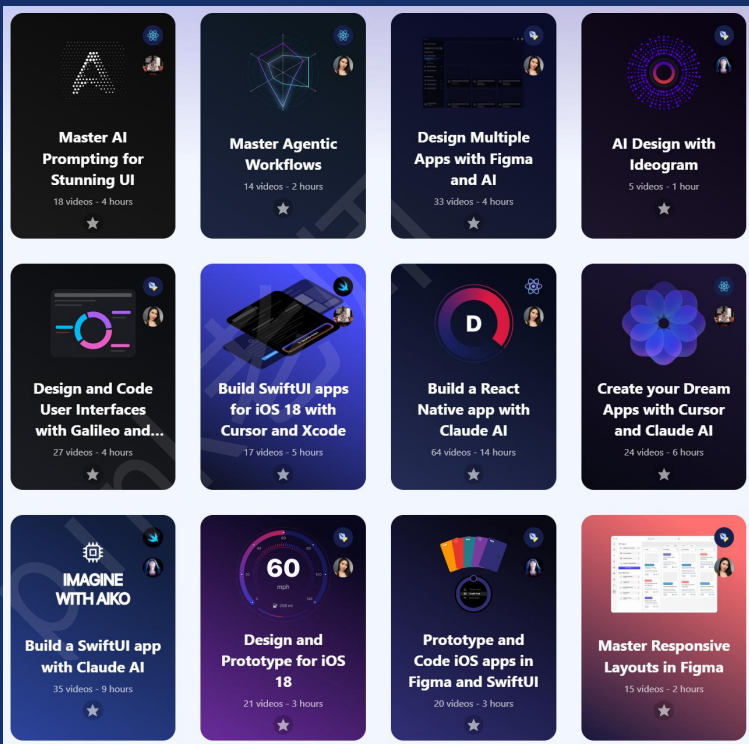
综合案例

<https://designcode.io/courses>



综合案例

<https://designcode.io/courses>



核心代码: `grid-template-columns: repeat(auto-fill,minmax(210px,1fr))`

综合案例

<https://designcode.io/courses>

svg 裁剪波浪效果



```
<svg width="1440" height="699" viewBox="0 0 1440 699"
fill="none" xmlns="http://www.w3.org/2000/svg">
  <defs>
    <clipPath id="wave2">
      <path
        d="M1192 63.1469C979.197 -18.7459 969.021 -3.8326
        747.038 15.147C525.055 34.1266 456.426 81.6931 291.
        038 63.147C125.65 44.6009 38.0215 68.2247 -41.1011
        88.0053C-41.1011 175.938 -41.1013 376.556 -41.1013
        376.556L1481 376.556L1481 63.1469C1481 63.1469 1405
        19 145.19 1192 63.1469Z"
        fill="url(#paint0_linear)"></path>
      </clipPath>
    </defs>
  </svg>
```

```
.bolang {
  position: absolute;
  left: 0;
  top: -200px;
  width: 100%;
  height: 600px;
  background: linear-gradient(□rgba(19, 12,
    62, 0.8) 0%, □rgb(30, 19, 87) 28%);
  /* 利用svg裁剪盒子 */
  clip-path: url(#wave2);
}
```

Contents

01.媒体查询

transform

02.Bootstrap响应式

case

Bootstrap框架

Bootstrap 是由 Twitter 团队开发的开源前端框架，用于快速构建响应式、移动优先的网站和 Web 应用程序。它提供了一套完整的 HTML、CSS 和 JavaScript 组件，使开发者可以轻松实现现代化的 UI 设计。

- 提供大量预定义样式和组件，减少重复代码。
- 开发者可以专注于业务逻辑，而不必从零开始写 CSS。



官网：<https://getbootstrap.com/>

Bootstrap框架

学习的目的除了做响应式网页外，最主要学会使用框架，后面我们还会有很多框架学习。

学习框架：

1. 官方文档。查看官网文档，学习核心概念、使用规范，然后做demo实现基础功能。
2. 实战项目。复刻经典项目，逐步添加自定义功能。
3. 原理剖析。阅读源码，理解原理与最佳实践。



官网：<https://getbootstrap.com/>

中文官网：<https://v5.bootcss.com/docs>

Bootstrap框架

学习的目的除了做响应式网页外，最主要学会使用框架，后面我们还会有很多框架学习。

Bootstrap使用：

1. 下载并引入相关文件。
2. 学习官方文档指引，学习框架相关语法。

下载 Bootstrap

下载 Bootstrap 以获得编译后的 CSS 和 JavaScript 文件。你可以使用你喜欢的软件包管理器，例如 npm、RubyGems 等，在你的项目中。


经过编译的 CSS 和 JS


下载 Bootstrap v5.3.0-alpha1 版本经过编译并立即可用的文件，以便直接使用。

- 编译并压缩过的 CSS 集成包 (参见 [CSS 文件比较](#))
- 编译并压缩过的 JavaScript 插件 (参见 [JS 文件比较](#))

不包括文档、源文件或任何可选的 JavaScript 依赖项 (例如 Popper)。

[下载 Bootstrap 生产文件](#)

 bootstrap.min.css

 bootstrap.min.js

```
<!-- 引入bootstrap css文件 -->
<link rel="stylesheet" href="./css/bootstrap.min.css">
<!-- 引入bootstrap js文件 -->
<script src="./js/bootstrap.min.js"></script>
```

Bootstrap框架

1. 内容 (content)

内容包含：字体排版比如标题、段落以及图片、表格、表单等。



使用方法：

1. 找到文档对应内容。
2. 查看样例，然后复制文档示例。
3. 根据需求调整。（重点学习类名）

Bootstrap框架

2. 实用工具 (content)

实用工具包含：CSS常见样式比如背景、边框、颜色等。

✓ 工具类

API

边框(Borders)

颜色(Colors)

Display属性

Flex布局

浮动(Float)

交互(Interactions)

溢出(Overflow)

定位(Position)

阴影(Shadows)

调整大小(Sizing)

间距(Spacing)

文本

垂直对齐

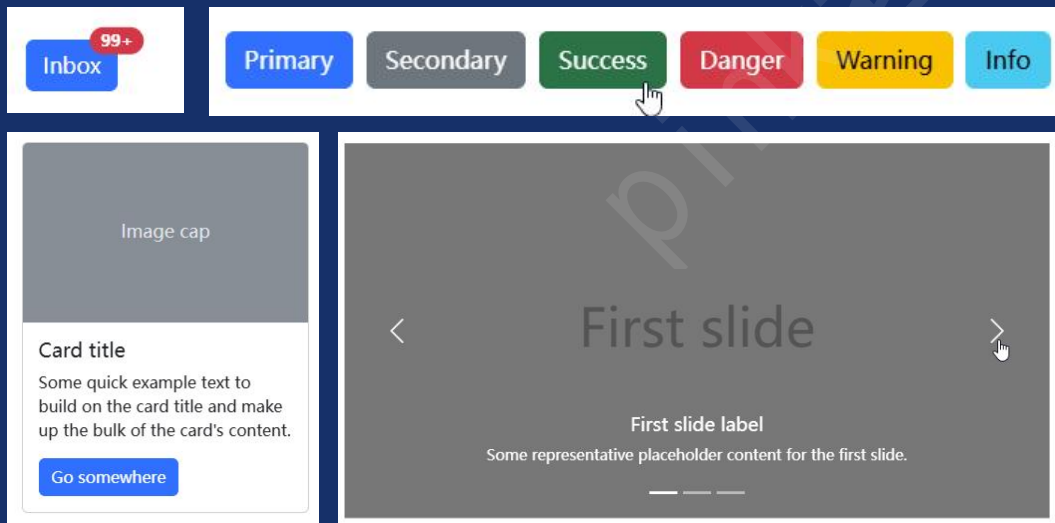
可见性

Bootstrap框架

3. 组件 (Components)

Bootstrap组件是预定义的网页元素，具有各种不同的功能，从呈现内容到用户交互都有涵盖。

常见的Bootstrap组件比如徽章、按钮、卡片、轮播图、导航栏、选项卡等



B 中文文档 示例

Components

- Accordion
- 警告框 (Alerts)
- 徽章 (Badge)
- 面包屑导航 (Breadcrumb)
- Buttons
- Button group
- Card
- Carousel
- Close button
- Collapse
- Dropdowns
- List group
- Modal
- Navbar
- Navs & tabs
- Offcanvas

Bootstrap框架

3. 组件 (Components)

Bootstrap组件是预定义的网页元素，具有各种不同的功能，从呈现内容到用户交互都有涵盖。

常见的Bootstrap组件比如徽章、按钮、卡片、轮播图、导航栏、选项卡等

使用方法：

1. 需要引入Bootstrap的 [css文件](#) 和 [js文件](#)。
2. 找到文档对应内容，选择样例。
3. 复制文档提供的样例代码，根据需求调整。



中文文档 示例

Components

Accordion

警告框 (Alerts)

徽章 (Badge)

面包屑导航
(Breadcrumb)

Buttons

Button group

Card

Carousel

Close button

Collapse

Dropdowns

List group

Modal

Navbar

Navs & tabs

Offcanvas

Bootstrap框架

4. 布局 (Layout)

Bootstrap布局是我们学习的重点，重点学习里面的网格系统（栅格系统）需要学习断点、容器、栅格系统、列等相关内容。

4.1 断点：

断点（Breakpoints）是响应式设计的组成部分。通过断点可以控制何时可以在特定视口（viewport）或设备尺寸下调整布局。

Breakpoint	Class infix	Dimensions
Extra small	<i>None</i>	<576px
Small	<i>sm</i>	≥576px
Medium	<i>md</i>	≥768px
Large	<i>lg</i>	≥992px
Extra large	<i>xl</i>	≥1200px
Extra extra large	<i>xxl</i>	≥1400px



中文文档 示例

Layout

Breakpoints

Containers

Grid

Columns

Gutters

Utilities

Z-index

CSS Grid

Bootstrap框架

4. 布局 (Layout)

4.2 容器：

容器是 Bootstrap 中最基本的布局元素，在使用我们的默认网格系统时是必需的。容器用于将其中的内容居中。

包含：默认容器（container）、响应式容器（Responsive containers）、流式容器（Fluid containers）三种

Bootstrap附带三种不同的容器：

- `.container`，它在每个响应断点处设置 `max-width`
- `.container-fluid`，所有断点 `width: 100%`
- `.container-{breakpoint}`，直到指定的断点 `width: 100%`为止

	Extra small <576px	Small ≥576px	Medium ≥768px	Large ≥992px	X-Large ≥1200px	XX-Large ≥1400px
<code>.container</code>	100%	540px	720px	960px	1140px	1320px
<code>.container-sm</code>	100%	540px	720px	960px	1140px	1320px
<code>.container-md</code>	100%	100%	720px	960px	1140px	1320px
<code>.container-lg</code>	100%	100%	100%	960px	1140px	1320px
<code>.container-xl</code>	100%	100%	100%	100%	1140px	1320px
<code>.container-xxl</code>	100%	100%	100%	100%	100%	1320px
<code>.container-fluid</code>	100%	100%	100%	100%	100%	100%

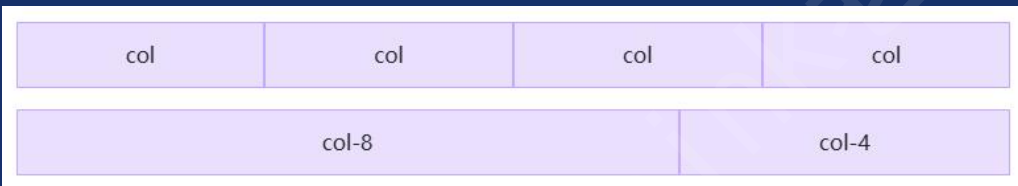
Bootstrap框架

4. 布局 (Layout)

4.3 网格系统 (grid) :

Bootstrap 的网格系统（栅格系统）使用一系列容器、行和列来布局和对齐内容。它是用 flexbox 构建的，并且是完全响应的。

注意：Bootstrap 的网格系统每行划分为12份。



	xs <576px	sm ≥576px	md ≥768px	lg ≥992px	xl ≥1200px	xxl ≥1400px
Container max-width	None (auto)	540px	720px	960px	1140px	1320px
Class prefix	.col-	.col-sm-	.col-md-	.col-lg-	.col-xl-	.col-xxl-
# of columns	12					

```
<div class="container">
  <div class="row">
    <div class="col-md-3">1</div>
    <div class="col-md-3">2</div>
    <div class="col-md-3">3</div>
    <div class="col-md-3">4</div>
  </div>
</div>
```

注意：

1. row 代表一行
2. col-xx-份数 比如 col-md-3
3. 可以写多个类名，决定不同宽度下排列个数

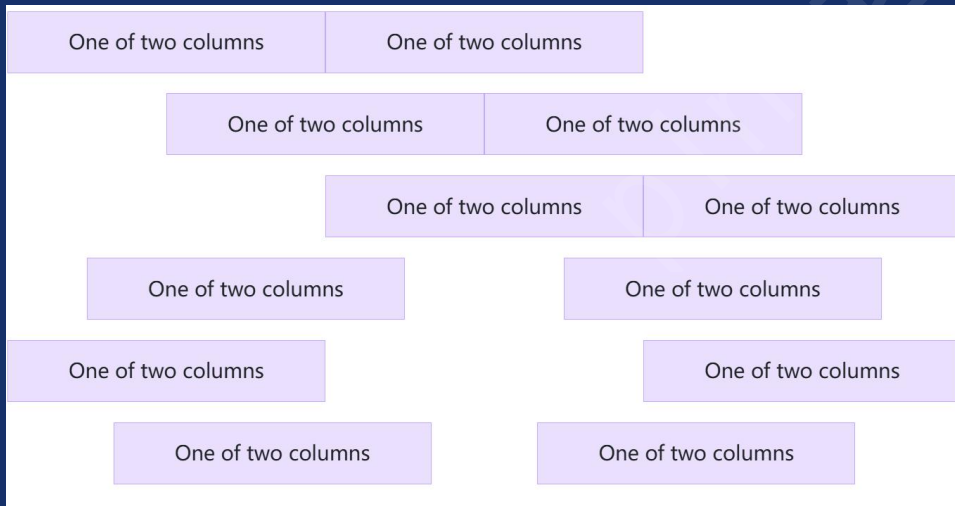
Bootstrap框架

4. 布局 (Layout)

4.4 列对齐:

可以参考官网: <https://v5.bootcss.com/docs/layout/columns/>

比如水平对齐可以: `justify-content-*`



Bootstrap框架

5. 字体图标 (Layout)

 中文文档 代码实例 图标库 模板/主题 官方博客

安装

Bootstrap 图标库 已被发布到了 npm，但是仍然可以手动下载并安装。

通过软件包管理工具来安装

通过 npm 或 Composer 来安装 [Bootstrap 图标库](#)，其中包括 SVG 格式的图标、sprite 图标和图标字体。然后，请参考 [使用指南](#) 来选择并使用你所需要的图标。

```
$ npm i bootstrap-icons
```

```
$ composer require twbs/bootstrap-icons
```

下载安装包

[GitHub 上所有已发布的版本](#)，并且在压缩包内包含了所有 SVG 格式的图标、许可证和 readme 文件。`package.json` 文件也被包含在内，不过这些 npm 脚本主要用于我们的开发流程。

[下载最新版本的 ZIP 压缩包](#)

通过公共 CDN 加载

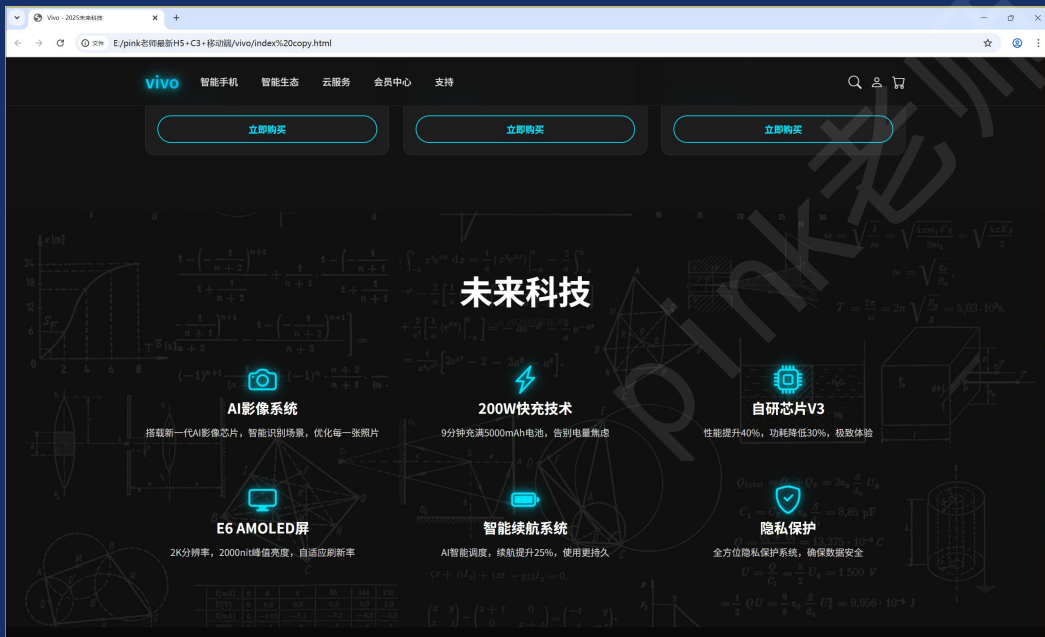
利用我们提供的公共 CDN 服务并将图标字体的样式表添加到网站的 `<head>` 标签内（或通过 CSS 的 `@import` 指令加载）就能立即使用 Bootstrap 图标库了。参见[图标字体相关文档](#)以查看示例用法。

```
<link rel="stylesheet" href="https://cdn.jsdelivr.net/npm/bootstrap-icons@1.5.0/font/bootstrap-icons.css">
```

```
@import url("https://cdn.jsdelivr.net/npm/bootstrap-icons@1.5.0/font/bootstrap-icons.css");
```


Bootstrap框架

综合案例：



案例目标：

1. 实战方式更深刻理解bootstrap使用。
2. 根据实际需求修改bootstrap样式。
3. 学习到新的知识点。

Bootstrap框架

1. TDK 网站优化标签

TDK是Title、Description和Keywords三个元标签的缩写，它们是网站SEO优化的核心元素。直接影响搜索引擎对网页的理解和排名。

Title: 决定搜索结果的标题显示，是搜索引擎判断页面主题的首要依据。

Description: 生成搜索结果的摘要描述，影响用户点击率（CTR）。

Keywords: 提供页面主题关键词（尽管现代搜索引擎已弱化其权重，仍部分参考）

三者共同构成网页的“第一印象”，优化不当可能导致排名下降或点击流失

```
<title>vivo智能手机官方网站 - X200s 影像旗舰</title>
```

```
<meta name="keywords"
```

```
content="vivo, vivo官网, vivo智能手机大全, vivo手机最新报价, vivo手机图片, X200系列、S20系列、X Fold3、vivo Pad5 Pro、vivo WATCH 5、iQOO 13">
```

```
<meta name="description" content="手机领域的国际化品牌, vivo官网提供手机资讯、售前咨询、在线购买、售后服务等功能; vivo当季明星...">
```

Bootstrap框架

2. 图片遮罩 mask-image

`mask-image` 的核心作用是通过定义一张图片或一个渐变来充当遮罩，从而控制一个元素的可视区域。。

语法：

`mask-image: url(..xx.jpg)` 或者 `mask-image: 线性渐变 / 径向渐变;`

```
mask-image: linear-gradient(□rgb(0, 0, 0) 80%, □rgba(0, 0, 0, 0) 100%);
```

Bootstrap框架

3. 自定义浏览器滚动条



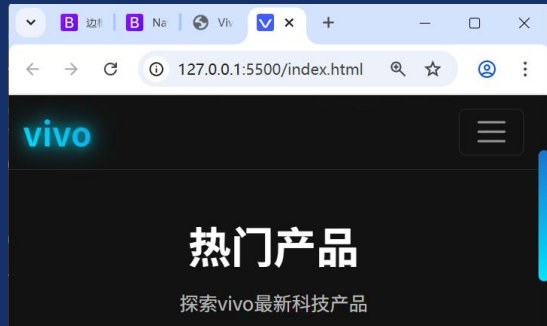
轨道

滑块

Bootstrap框架

3. 自定义浏览器滚动条

```
/* 滚动条设置宽度和高度 */  
::-webkit-scrollbar {  
  width: 10px;  
}  
/* 设置轨道背景色, 圆角等 */  
::-webkit-scrollbar-track {  
  background: var(--vivo-dark);  
}  
/* 设置滑块样式 */  
::-webkit-scrollbar-thumb {  
  background: linear-gradient(var(--vivo-blue), var(--vivo-cyan));  
  border-radius: 10px;  
}  
/* 鼠标经过滑块 */  
::-webkit-scrollbar-thumb:hover {  
  background: linear-gradient(var(--vivo-cyan), var(--vivo-blue));  
}
```



T H A N K Y O U

行无止境