

Web Programming

Web开发技术基础

第2章 Web页面开发技术(自学为主)

 计算机学院

 授课人：王尊亮

第2章 Web页面相关技术（自学为主）

2.1 HTML基础

- 2.1.1 HTML简介
- 2.1.2 HTML调试
- 2.1.3 HTML表单
- 2.1.4 HTML5

2.2 CSS基础（简要了解）

- 2.2.1 CSS简介
- 2.2.2 CSS调试
- 2.2.3 CSS选择器
- 2.2.4 响应式设计

2.3 JavaScript基础

- 2.3.1 JavaScript简介
- 2.3.2 JavaScript操作DOM
- 2.3.3 JavaScript操作BOM
- 2.3.4 JQuery

2.4 常用Web页面设计开发与框架

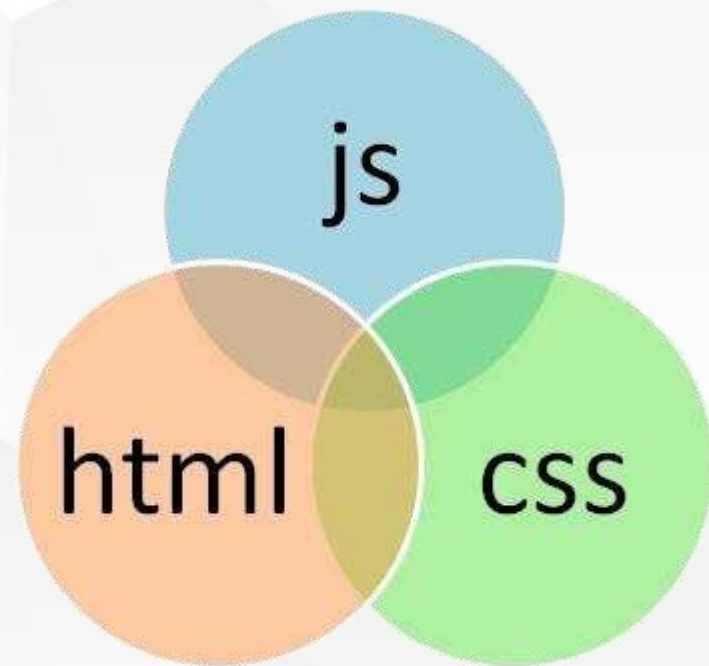
第2章 Web页面相关技术（自学为主）

Web页面主要有3部分组成：结构（structure）、表现（presentation）和行为（behavior）。

HTML：网页元素的结构

CSS：网页元素的表现形式，如：位置、颜色、大小

JS：网页元素的交互行为，如：按钮点击



内容与表现分离的示例与优势，[示例](#)

- ▶ 易于维护和改版
- ▶ 提高页面浏览速度
- ▶ 便于搜索引擎搜索
- ▶ 便于实现网页信息无障碍

2.1.1 HTML简介

- ▶ **HTML** (HyperText Markup Language): **超文本标记语言**
 - ▶ 超文本: 不仅可以加入文本, 还可以加入链接、图片、声音、动画、影视等内容的文本文件。
 - ▶ 标记语言: 不需要编译, 每个标签都是一条命令, 告诉浏览器如何显示文本, 通过标记标签来**描述**网页
 - ▶ 后缀: .html或.htm, 推荐使用html后缀

2.1.1 HTML简介

▶ HTML 标准的发展



Welcome to the WHATWG community
Maintaining and evolving HTML since 2004

- ▶ 超文本标记语言（第一版）——在1993年6月作为IETF工作草案发布
- ▶ HTML 2.0——1995年11月作为RFC 1866发布
- ▶ HTML 3.2——1997年1月14日，W3C推荐标准
- ▶ HTML 4.0——1997年12月18日，W3C推荐标准
- ▶ HTML 4.01——1999年12月24日，W3C推荐标准，常用标准
- ▶ XHTML 1.0（可扩展超文本标记语言）——2000年底，W3C标准，一个过渡技术，结合了部分XML的强大功能及大多数HTML的简单特性，此后W3C致力于XHTML2.0
- ▶ 在2004年，由Opera、Mozilla Foundation和Apple等浏览器厂商组成WHATWG: Web Hypertext Application Technology Working Group，Web 超文本应用技术工作组，
- ▶ HTML 5——2014年10月28日，<https://html.spec.whatwg.org/>，<https://whatwg-cn.github.io/html/>（中文版）

2.1.1 HTML简介

Table of contents

- 1 Introduction
- 2 Common infrastructure
- 3 Semantics, structure, and APIs of HTML documents
- 4 The elements of HTML
- 5 Microdata
- 6 User interaction
- 7 Loading web pages
- 8 Web application APIs
- 9 Communication
- 10 Web workers
- 11 Worklets
- 12 Web storage
- 13 The HTML syntax
- 14 The XML syntax
- 15 Rendering
- 16 Obsolete features
- 17 IANA considerations
- Index
- References
- Acknowledgments
- Intellectual property rights

目录

- 1 概述
- 2 公共基础设施
- 3 HTML 文档的语义, 结构, 与API
- 4 HTML 元素
- 5 微数据
- 6 用户交互
- 7 加载 Web 页面
- 8 Web 应用 API
- 9 通信
- 10 Web 工作线程
- 11 Worklets
- 12 Web 存储
- 13 HTML 语法
- 14 XML 语法
- 15 渲染
- 16 废弃的特性
- 17 IANA 注意事项
- 索引
- References
- Acknowledgments
- Intellectual property rights

2.1.1 HTML简介

The WHATWG was based on **several core principles**:

- technologies need to be **backwards compatible** (向后兼容)
- specifications and implementations need to match even if this means **changing the specification rather than the implementations**
- specifications need to be **detailed enough** that implementations can achieve complete **interoperability** (互操作) without reverse-engineering each other.

2.1.1 HTML简介

Web领域专家 Jeremy Keith曾在Fronteers 2010发表过一个关于HTML发展及设计原则的主题演讲，内容链接如下：

1. 演讲原文链接 [The Design of HTML5](#)
2. 演讲PPT <http://adactio.com/extras/slides/designofhtml5.pdf>
3. 演讲中文翻译版， [Web的真谛：HTML5到底是什么意思.](#)

设计原则是一种信念、一种想法、是行动的支柱。不管是制定规范，还是制造一种有形的物品，或者编写软件，甚至发明编程语言，都能找到背后的一个或者多个设计原则。

- | | |
|-------------------------------------|--|
| 1. avoid needless complexity | 4. pave the cowpaths |
| 2. support existing content | 5. degrade gracefully, 优雅降级 |
| 3. solve real problems | 6. priority of constituencies, 最终用户优先 |

2.1.1 HTML简介

▶ HTML 语法

- ▶ HTML文件由标题、段落、列表、表格、文本以及嵌入的各种对象组成，这些逻辑上统一的对象称为**元素**。
- ▶ HTML使用标签来描述这些元素
 - ▶ **<标签名>内容</标签名>** -----一个HTML元素
 - ▶ 元素以**开始标签**起始，以**结束标签**终止，元素的**内容**是开始标签与结束标签之间的内容
 - ▶ 某些 HTML 元素具有**空内容**，**空元素在开始标签中进行关闭**，如

 - ▶ 大多数 HTML 元素可拥有**属性**，为元素提供附件信息，形式为“**名称=值**”例如：

2.1.1 HTML简介

▶ HTML 语法

适用于大多数 HTML 元素的属性

属性	描述
class	为html元素定义一个或多个类名（classname）(类名从样式文件引入)
id	定义元素的唯一id
style	规定元素的行内样式（inline style）
title	描述了元素的额外信息 (作为工具条使用)

更多标准属性说明: [HTML 标准属性参考手册](#).

2.1.1 HTML简介

▶ HTML 基本文档

▶ <!DOCTYPE html>

<html>

<head>

<title>文档标题</title>

</head>

▶ <body>

可见文本...

</body>

▶ </html>

2.1.1 HTML简介

▶ **<!DOCTYPE> 声明**

- ▶ 向浏览器声明HTML的版本，浏览器就能正确显示网页内容。
- ▶ Doctype几个字符不区分大小写。声明不是一个 HTML 标签
- ▶ 常见声明：

- ▶ HTML5: `<!DOCTYPE html>`

- ▶ HTML 4.01:

- ▶ `<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN"`
`"http://www.w3.org/TR/html4/loose.dtd">`

- ▶ XHTML 1.0

- ▶ `<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"`
`http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd>`

2.1.2 HTML调试



input#kw.s_ipt 548 x 44

百度一下

关于百度 About Baidu 使用百度前必读 帮助中心 京公网安备11000002000001号 京ICP证030173号 ©2021 Baidu 互联网药品信息服务资格证书(京)-经营性-2017-0020 信

DevTools is now available in Chinese! Always match Chrome's language Switch DevTools to Chinese Don't show again

Elements Console Sources Network Performance Memory Application Lighthouse

```
<div id="head_wrapper" class="s-is-index-wrap head_wrapper s-title-img">
  <link rel="stylesheet" href="https://ss1.bdstatic.com/5eN1bjq8AAUym2zgoY3K/r/www/cache/static/protocol/https/soutu/css/soutu_new2_8c3ab22.css"
  type="text/css" data-for="result">
  <div id="s_fm" class="s_form s_form_login">
    <div class="s_form_wrapper soutu-env-mac soutu-env-newindex" id="s_form_wrapper">
      <div id="lg" class="s-p-top">...</div>
      <a href="/" id="result_logo" onmousedown="return c({'fm':'tab','tab':'logo'})">...</a>
      <form name="f" id="form" action="/s" class="fm" onsubmit="javascript:F.call('ps/sug','pssubmit');">
        <span id="s_kw_wrap" class="bg s_ipt_wr new-pmd quickdelete-wrap">
          <span class="ipt_rec" style="display: block;"></span>
          <span class="soutu-btn"></span>
          <input type="text" class="s_ipt" name="wd" id="kw" maxlength="100" autocomplete="off"> == $0
          <span class="soutu-hover-tip" style="display: none;">按图片搜索</span>
          <span class="voice-hover" style="display: none;">按语音搜索</span>
        </span>
        <input type="hidden" name="rsv_spt" value="1">
      </form>
    </div>
  </div>
</div>
```

Styles Computed La

Filter

```
element.style {
}
```

```
#head_wrapper .soutu
width: 450px;
padding-right: 78p
}
```

```
#head_wrapper #kw {
width: 512px;
height: 16px;
padding: 12px 16p
font-size: 16px;
margin: 0;
}
```

2.1.3 HTML表单

- ▶ 表单用于搜集不同类型的用户输入，并将数据提交给服务器
- ▶ 表单用<form></form>标签来创建，在开始和结束标签之间的属于表单的内容，单击提交按钮时，提交的也是表单范围内的内容。
- ▶ 重要属性
 - ▶ name: 表单名称
 - ▶ method: 提交数据的方式，可取值为GET和POST中的一个，GET是将表单内容附加在URL地址后面，长度受限，POST不会再浏览器地址栏中显示提交信息，数据长度理论上没有限制。
 - ▶ action: 表单内容提交到的处理程序url
 - ▶ enctype: 将表单数据发送到服务器之前如何对其进行编码，值有" application/x-www-form-urlencoded"、" multipart/form-data"，" text/plain"

2.1.3 HTML表单

▶ HTML表单标签

标签		描述
<u><form></u>		定义供用户输入的表单
<u><input></u>		定义输入域
<u><textarea></u>		定义文本域 (一个多行的输入控件)
<u><label></u>		定义了 <input> 元素的标签, 一般为输入标题
<u><fieldset></u>		定义了一组相关的表单元素, 并使用外框包含起来
<u><legend></u>		定义了 <fieldset> 元素的标题
<u><select></u>		定义了下拉选项列表
<u><optgroup></u>		定义选项组
<u><option></u>		定义下拉列表中的选项
<u><button></u>		定义一个点击按钮
<u><datalist></u>	New	指定一个预先定义的输入控件选项列表
<u><keygen></u>	New	定义了表单的密钥对生成器字段
<u><output></u>	New	定义一个计算结果

2.1.3 HTML表单

▶ Input标签

▶ 重要属性

▶ type

值		描述
button		定义可点击的按钮（通常与 JavaScript 一起使用来启动脚本）。
checkbox		定义复选框。
color	New	定义拾色器。
date	New	定义 date 控件（包括年、月、日，不包括时间）。
datetime	New	定义 date 和 time 控件（包括年、月、日、时、分、秒、几分之一秒，基于 UTC 时区）。
datetime-local	New	定义 date 和 time 控件（包括年、月、日、时、分、秒、几分之一秒，不带时区）。
email	New	定义用于 e-mail 地址的字段。
file		定义文件选择字段和“浏览...”按钮，供文件上传。
hidden		定义隐藏输入字段。

2.1.3 HTML表单

image		定义图像作为提交按钮。
month	New	定义 month 和 year 控件（不带时区）。
number	New	定义用于输入数字的字段。
password		定义密码字段（字段中的字符会被遮蔽）。
radio		定义单选按钮。
range	New	定义用于精确值不重要的输入数字的控件（比如 slider 控件）。
reset		定义重置按钮（重置所有的表单值为默认值）。
search	New	定义用于输入搜索字符串的文本字段。
submit		定义提交按钮。
tel	New	定义用于输入电话号码的字段。
text		默认。定义一个单行的文本字段（默认宽度为 20 个字符）。
time	New	定义用于输入时间的控件（不带时区）。
url	New	定义用于输入 URL 的字段。
week	New	定义 week 和 year 控件（不带时区）。

2.1.3 HTML表单

学习实例：

创建文本字段 (Text field)

创建密码字段

单选按钮(Radio buttons)

复选框(Checkboxes)

简单的下拉列表

预选下拉列表

文本域(Textarea)

创建按钮

2.1.4 HTML5

- ▶ HTML5, 新一代的HTML标准
 - ▶ 新元素
 - ▶ 新属性
 - ▶ 完全支持 CSS3
 - ▶ Video 和 Audio
 - ▶ 2D/3D 制图
 - ▶ 本地存储
 - ▶ 本地 SQL 数据
 - ▶ WEB应用 连接特性, 服务器数据推送

2.1.4 HTML5

▶ Canvas

- ▶ 支持脚本绘制图形



▶ 内联SVG

- ▶ SVG 指可伸缩矢量图形 (Scalable Vector Graphics)
- ▶ SVG 用于定义用于网络的基于矢量的图形
- ▶ SVG 使用 XML 格式定义图形
- ▶ SVG 图像在放大或改变尺寸的情况下其图形质量不会有损失

2.1.4 HTML5

▶ SVG 与 Canvas两者间的区别

SVG 是一种使用 XML 描述 2D 图形的语言。

Canvas 通过 JavaScript 来绘制 2D 图形。

SVG 基于 XML，这意味着 SVG DOM 中的每个元素都是可用的。您可以为某个元素附加 JavaScript 事件处理器。

在 SVG 中，每个被绘制的图形均被视为对象。如果 SVG 对象的属性发生变化，那么浏览器能够自动重现图形。

Canvas 是逐像素进行渲染的。在 canvas 中，一旦图形被绘制完成，它就不会继续得到浏览器的关注。如果其位置发生变化，那么整个场景也需要重新绘制，包括任何或许已被图形覆盖的对象。

2.1.4 HTML5

- ▶ HTML5 MathML
- ▶ 拖放 (Drag 和 Drop)
- ▶ Geolocation (地理定位)
- ▶ 视频
- ▶ 音频
- ▶ 新的 Input 类型
- ▶ 新的表单元素
- ▶ 新的表单属性
- ▶ 语义元素
- ▶ Web存储
- ▶ Web SQL
- ▶ 应用程序缓存
- ▶ Web Workers
- ▶ 服务器发送事件(Server-Sent Events)
- ▶ WebSocket

2.1.4 HTML5

▸ 语义元素

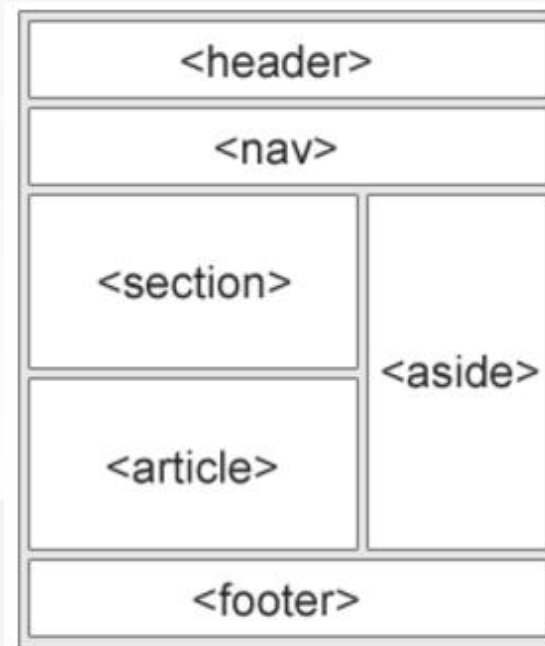
一个语义元素能够清楚的描述其意义给浏览器和开发者。

无语义 元素实例: `<div>` 和 `` - 无需考虑内容.

语义 元素实例: `<form>`, `<table>`, and `` - 清楚的定义了它的内容.

HTML5提供了新的语义元素来明确

一个Web页面的不同部分



2.1.4 HTML5

- ▶ Web存储
- ▶ HTML5 web 存储,一个比cookie更好的本地存储方式。

客户端存储数据的两个对象为:

localStorage - 没有时间限制的数据存储

sessionStorage - 针对一个 session 的数据存储, 当用户关闭浏览器窗口后, 数据会被删除。

```
<script>
if(typeof(Storage)!="undefined")
{
    localStorage.sitename="某某网站";
    document.getElementById("result").innerHTML="网站名: "
}

```

示例

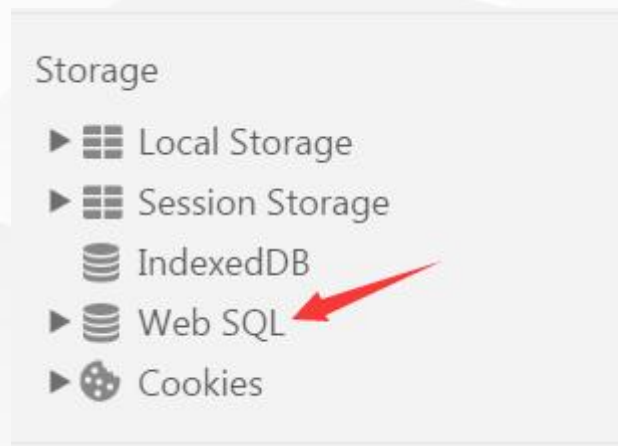
2.1.4 HTML5

▶ Web SQL

- ▶ Web SQL 数据库 API 并不是 HTML5 规范的一部分，但是它是一个独立的规范，引入了一组使用 SQL 操作客户端数据库的 APIs。

▶ Indexed Database

- ▶ 同样是一个Web客户端存储结构化数据的规范，是一个完全内置于浏览器中的一个沙盒环境中的文档数据库（NoSQL）



2.2.1 CSS简介

- ▶ 层叠样式表 (Cascading Style Sheets)
 - ▶ 样式表，样式表是文档的表现，html是文档的内容，内容与表现分离的思路，[示例](#)
 - ▶ 层叠：一组样式在一起层叠使用，控制一个或多个HTML元素。
 - ▶ 当前版本为CSS3
- ▶ CSS优点：
 - ▶ 内容和样式分离，在HTML文件中只存放内容信息，设计部分存放在独立的样式文件中，既可以实现精确的样式控制，又保持了html简单明了
 - ▶ 易于维护和改版
 - ▶ 提高页面浏览速度
 - ▶ 便于搜索引擎搜索
 - ▶ 便于实现网页信息无障碍

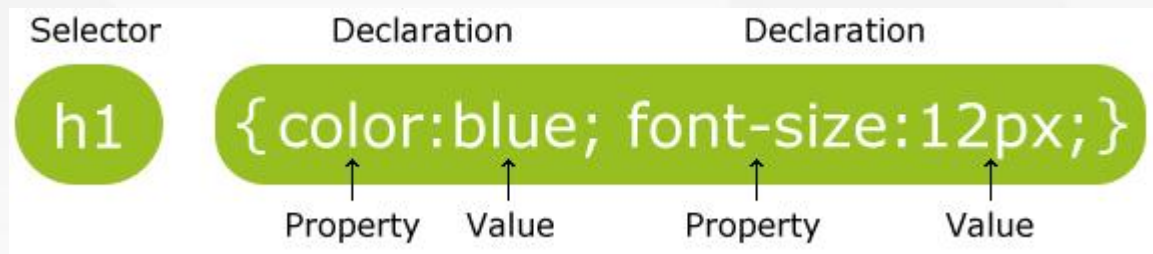
2.2.1 CSS简介



2.2.1 CSS简介

▶ CSS规则

- ▶ CSS 规则由两个主要的部分构成：选择器，以及一条或多条声明：



- ▶ 选择器通常确定需要改变样式的 HTML 元素。
- ▶ 每条声明由一个属性和一个值组成，属性和值被冒号分开，多个声明直接用分号分开。
- ▶ [实例1](#)，[实例2](#)

2.2.1 CSS简介

▶ 放置CSS的几种方式:

- ▶ **内联样式表**: 针对任意body元素

- ▶ `<p style="color:red;font-family:serif">内联样式</p>`

- ▶ **嵌入样式表**: 放到html的文档的head部分。

- ▶ `<head>`

- ▶ `<style type="text/css">`

- ▶ `</style>`

- ▶ `</head>`

- ▶ **外部样式表**: 在head部分通过link链接

- ▶ `<link rel="StyleSheet" href="style.css">`

- ▶ 当同一个 HTML 元素被不止一个样式定义时, 优先级为: **浏览器缺省设置<外部样式表<嵌入样式表<内联样式表**

2.2.1 CSS简介

- ▶ 背景: CSS没有变量、函数、SCOPE, 需要书写大量看似没有逻辑的代码, 不方便维护及扩展, 不利于复用, 为了方便前端开发的工作量, 出现了sass和less。
- ▶ SASS: **S**yntactically **A**wesome **S**tylesheets, 基于Ruby语言开发, 对css的一种扩展提升, 增加了规则、变量、混入、选择器、继承等等特性。 <https://www.sass.hk/guide/>

- ▶ LESS: 基于JavaScript, 扩展了 CSS 语言, 增加了变量、Mixin、函数等特性, 使 CSS 更易维护和扩展。

```
@nice-blue: #5B83AD;  
@light-blue: @nice-blue + #111;  
  
#header {  
  color: @light-blue;  
}
```

```
$highlight-color: #F90;  
.selected {  
  border: 1px solid $highlight-color;  
}
```

//编译后

```
.selected {  
  border: 1px solid #F90;  
}
```

SASS中的变量引用

2.2.2 CSS调试

北京邮电大学建校66周年系列活动引起广泛关注

发布日期：2021-10-19 来源：宣传部、校友会

今年是中国共产党成立100周年，也是新中国成立72周年，在全校上下深入开展党史学习教育之际，10月18日，学校迎来了建校66周年校庆日。北京邮电大学六十六载征途，是在中国共产党的坚强领导下，始终与党和国家发展同向同行的奋进之路，是全体北邮人秉持“传邮万里、国脉所系”的家国情怀，勇挑重担、开拓创新之路。

is now available in Chinese! [Always match Chrome's language](#) [Switch DevTools to Chinese](#) [Don't show again](#)

Elements Console Sources Network Performance Memory Application Lighthouse

html>

/head>

```
tyle="position: absolute; z-index: 19700; top: -1970px; left: -1970px;">...</div>
d="vivo-head">...</div>
lass="nybanner" style="background:url(../images/20210617.jpg) top center no-repeat;"></div>
lass="nybox">...</div>
lass="ny">
  class="ny_left1">
    ript language="javascript" src="../../dwr/interface/NewsvoteDWR.js"></script>
    ript language="javascript" src="../../dwr/engine.js"></script>
    ript language="javascript" src="/system/resource/js/news/newscontent.js"></script>
    nk href="/system/resource/js/photoswipe/3.0.5.1/photoswipe.css" type="text/css" rel="stylesheet">
    ript language="javascript" src="/system/resource/js/photoswipe/3.0.5.1/klass.min.js"></script>
    ript language="javascript" src="/system/resource/js/ajax.js"></script>
    rm name="_newscontent_fromname">
    div class="content">
      <div class="content-title">
```

Styles Computed Layout Event List

Filter

```
element.style {
}
```

```
.content-title h3 {
  font-size: 24px;
  font-weight: bold;
  line-height: 50px;
  width: 100%;
  text-align: center;
  color: #0c4a98;
  padding-bottom: 5px;
}
```

```
body, div, dl, dt, dd, ul, ol, li,
input, textarea, blockquote, p {
  padding: 0px;
  margin: 0px;
  font-weight: normal;
}
```

2.2.3 CSS选择器

- ▶ HTML选择器，也叫标签选择器
 - ▶ 通过标签名来选择，如：p{ color:red}
- ▶ class类选择器
 - ▶ 用于描述一组元素的样式，class 选择器在HTML中以class属性表示，在 CSS 中，类选择器以一个点"."号显示：
 - ▶ .center {text-align:center;}
- ▶ Id选择器
 - ▶ 为标有特定 id 的 HTML 元素指定特定的样式，唯一
 - ▶ HTML元素以id属性来设置id选择器,CSS 中 id 选择器以“#”来定义。如#para {color:red;} **实例**

2.2.3 CSS选择器

▶ 组合选择器

▶ 后代选取器(以空格分隔)

▶ div p

{

background-color:yellow;

}

▶ 子元素选择器(以大于号分隔)

▶ 相邻兄弟选择器 (以加号分隔)

▶ 后续兄弟选择器 (以破折号分隔)

2.2.3 CSS选择器

- ▶ 伪类选择器
- ▶ 伪元素选择器

选择器	示例	示例说明
<u>:link</u>	a:link	选择所有未访问链接
<u>:visited</u>	a:visited	选择所有访问过的链接
<u>:active</u>	a:active	选择正在活动链接
<u>:hover</u>	a:hover	把鼠标放在链接上的状态
<u>:focus</u>	input:focus	选择元素输入后具有焦点
<u>:first-letter</u>	p:first-letter	选择每个<p> 元素的第一个字母
<u>:first-line</u>	p:first-line	选择每个<p> 元素的第一行
<u>:first-child</u>	p:first-child	选择器匹配属于任意元素的第一个子元素的 <p> 元素
<u>:before</u>	p:before	在每个<p>元素之前插入内容
<u>:after</u>	p:after	在每个<p>元素之后插入内容
<u>:lang(<i>language</i>)</u>	p:lang(it)	为<p>元素的lang属性选择一个开始值

2.2.4 响应式设计

- ▶ 响应式设计：(responsive web design, RWD)
 - ▶ 早年设计Web时，页面是**以适配特定的屏幕大小为考量**创建的。如果用户正在使用比设计者考虑到的更小或者更大的屏幕，那么结果是出现滚动条、过长行或者空白区域。
 - ▶ 随着移动Web的发展，有些网站为了适配移动端开发了多套页面，采用专用的域名如m.xxx.com或者根据请求中的user-agent判断设备类型而返回对应的页面。
 - ▶ 响应式设计RWD是**允许Web页面适应不同屏幕宽度因素等**，进行**布局和外观的调整**的一系列实践。
 - ▶ 响应式Web设计**不是单独的技术**，它是**描述Web设计的一种方式**、或者是一组最佳实践的一个词，它是用来建立可以响应查看内容的设备的样式的一个词。



2.3.1 JavaScript简介

- ▶ 什么是JavaScript?
 - ▶ JavaScript是一门轻量级的、动态的、弱类型的脚本语言，非常适合面向对象和函数式的编程风格
 - ▶ JavaScript被设计用来向HTML页面**添加交互行为**
 - ▶ html定义网页的内容
 - ▶ css定义网页的布局外观
 - ▶ js定义网页的行为
 - ▶ JS是可插入 HTML 页面的编程代码，但现在**一般写成单独的js文件**，有利于结构和行为的分离
 - ▶ 跨平台特性，在绝大多数浏览器的支持下，可以在多种平台下运行

2.3.1 JavaScript简介

- ▶ 什么是JavaScript?
 - ▶ JavaScript与Java无关
 - ▶ ECMAScript是JavaScript的标准规范 (ES6)
 - ▶ 2015年6月17日, ECMAScript 6发布正式版本, 即ECMAScript 2015。
 - ▶ 2016年 ECMAScript 7
 - ▶ 2017年 ECMAScript 8
 - ▶ <https://www.ecma-international.org/publications/standards/Ecma-262.htm>
 - ▶ TypeScript 是 JavaScript 的一个超集, 它可以编译成纯 JavaScript, 设计目标是开发大型应用

2.3.1 JavaScript简介

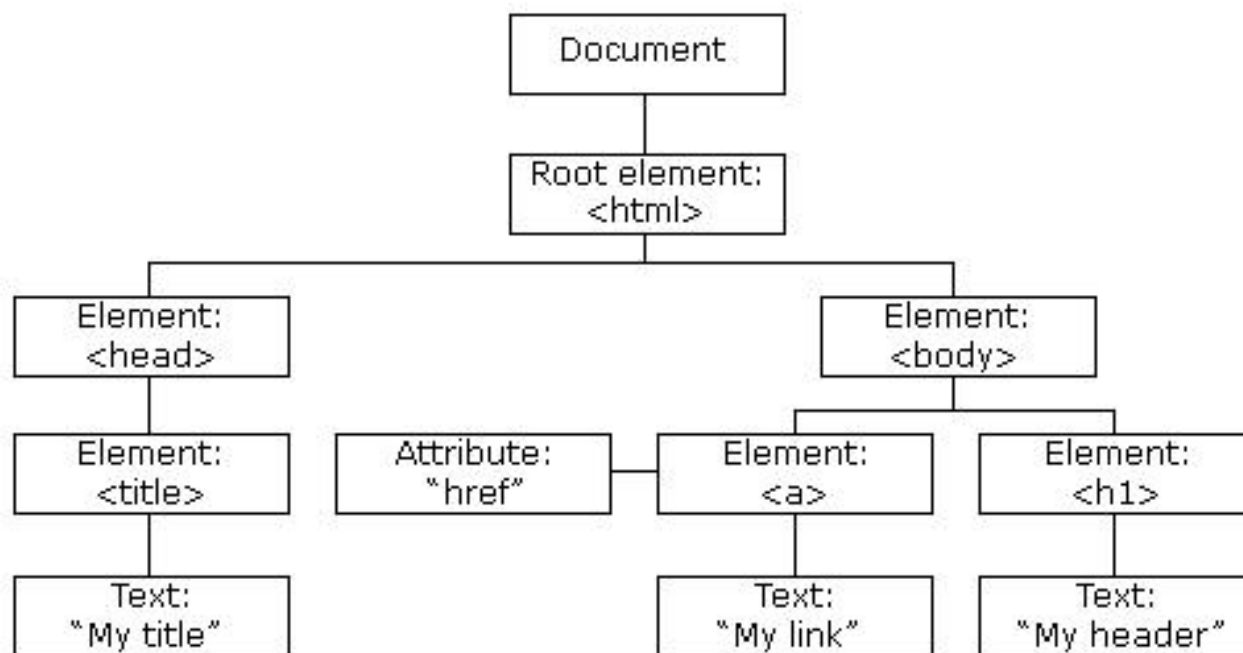
▶ JavaScript能干什么?

- ▶ 可以读写html元素
- ▶ 可以修改样式
- ▶ 可以对浏览器事件作出响应
- ▶ 可以在数据被提交到服务器之前验证数据
- ▶ 可以检测用户的浏览器信息
- ▶ 可以控制cookies
- ▶ 可以用于开发webapp
- ▶ 还可以基于Node.js技术进行后端编程
- ▶

2.3.2 JavaScript操作DOM

▶ DOM (Document Object Model)

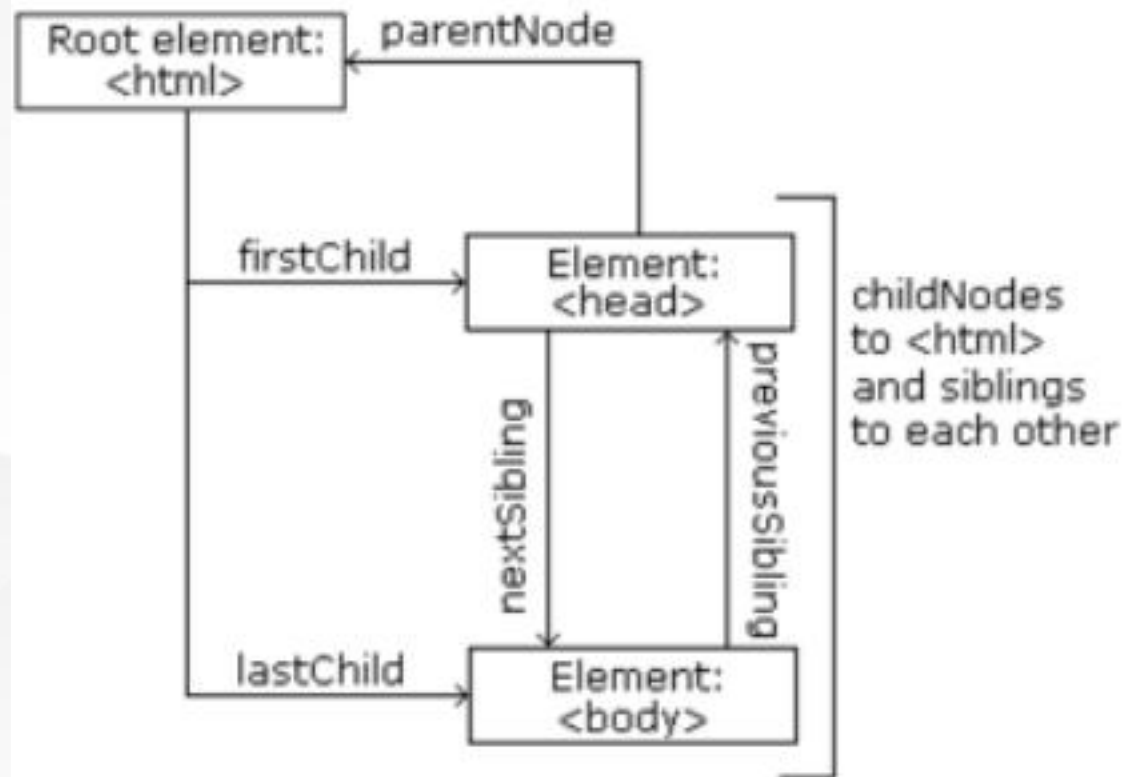
- ▶ 当网页被加载时，浏览器会创建页面的文档对象模型DOM
- ▶ DOM 是 W3C (万维网联盟) 的标准
- ▶ 文档对象模型 (DOM) 是中立于平台和语言的接口，它允许程序和脚本**动态地访问和更新文档的内容、结构和样式**
- ▶ 所有的HTML元素都是元素节点
- ▶ 所有 HTML 属性都是属性节点。
- ▶ 文本插入到 HTML 元素是文本节点。
- ▶ 注释是注释节点。



2.3.2 JavaScript操作DOM

节点树中的节点彼此拥有层级关系：父（parent）、子（child）和同胞（sibling）

```
<html>
  <head>
    <title>DOM Tutorial</title>
  </head>
  <body>
    <h1>DOM Lesson one</h1>
    <p>Hello world!</p>
  </body>
</html>
```



2.3.2 JavaScript操作DOM

▶ document 对象

- ▶ 当浏览器载入 HTML 文档, 它就会成为 document 对象。
- ▶ document 对象是HTML文档的根节点与所有其他节点（元素节点, 文本节点, 属性节点, 注释节点）。
- ▶ document 对象使我们可以从脚本中对 HTML 页面中的所有元素进行访问。
- ▶ document 对象属性和方法示例:
 - ▶ document.title
 - ▶ document.getElementsByClassName()
 - ▶ document.getElementById()
 - ▶ document.getElementsByName()

2.3.2 JavaScript操作DOM

▶ 访问 HTML 元素（节点）

- 通过使用 getElementById() 方法
- 通过使用 getElementsByTagName() 方法
- 通过使用 getElementsByClassName() 方法

▶ HTML DOM - 修改

- 改变 HTML 内容
- 改变 CSS 样式
- 改变 HTML 属性
- 创建新的 HTML 元素
- 删除已有的 HTML 元素
- 改变事件（处理程序）

2.3.2 JavaScript操作DOM

▶ 事件对象

- ▶ HTML DOM 事件允许Javascript在HTML文档元素中注册不同事件处理程序。
- ▶ 鼠标事件、键盘事件、框架/对象事件
- ▶ 表单事件、剪贴板事件、打印事件
- ▶ 拖动事件、多媒体（Media）事件
- ▶ 动画事件、过渡事件、其它事件

2.3.2 JavaScript操作DOM

- ▶ DOM
 - ▶ JavaScript 能够改变页面中的所有 HTML 元素
 - ▶ 示例1
 - ▶ JavaScript 能够改变页面中的所有 HTML 属性
 - ▶ 示例1
 - ▶ JavaScript 能够改变页面中的所有 CSS 样式
 - ▶ 示例
 - ▶ JavaScript 能够对页面中的所有事件做出反应
 - ▶ 示例

2.3.3 JavaScript操作BOM

- ▶ BOM (**B**rowser **O**bject **M**odel (BOM))
- ▶ Window对象
 - ▶ 所有浏览器都支持 window 对象。它表示浏览器窗口。
 - ▶ 所有 JavaScript 全局对象、函数以及变量均自动成为 window 对象的成员。
 - ▶ 全局变量是 window 对象的属性。
 - ▶ 全局函数是 window 对象的方法。
 - ▶ 甚至 HTML DOM 的 document 也是 window 对象的属性之一：

2.3.3 JavaScript操作BOM

▶ Window Screen

- ▶ window.screen 对象包含有关用户屏幕的信息。
- ▶ 宽度、高度

▶ Window Location

- ▶ 获得当前页面的地址 (URL), 并把浏览器重定向到新的页面

▶ Window History

- ▶ 包含浏览器的历史

▶ Window Navigator

- ▶ 浏览器信息

2.3.3 JavaScript操作BOM

▶ 弹窗

- ▶ 创建三种消息框：警告框、确认框、提示框
- ▶ `alert()`、`confirm()`、`prompt()`

▶ 计时事件

- ▶ 一个设定的时间间隔之后来执行代码
- ▶ `setInterval()` 和 `setTimeout()`

▶ Cookies

- ▶ 可以使用 `document.cookie` 属性来创建、读取、及删除 cookies

2.3.4 jQuery

▶ jQuery

- ▶ jQuery是一个JavaScript函数库。
- ▶ jQuery是一个轻量级的"写的少，做的多"的JavaScript库。
- ▶ jQuery库包含以下功能：
 - ▶ HTML 元素选取与操作
 - ▶ CSS 操作
 - ▶ HTML 事件函数
 - ▶ JavaScript 特效和动画
 - ▶ HTML DOM 遍历和修改
 - ▶ AJAX 以及其它 Utilities
 - ▶ 除此之外，Jquery还提供了大量的插件。

2.3.4 jQuery

▶ jQuery

▶ 开发版和发布版

- ▶ 发布版是被精简和压缩的,

▶ 部署方式

▶ 放在自己的Web服务器上

- ▶ `<script src="jquery-1.10.2.min.js"></script>`

▶ 从CDN中载入jQuery

- ▶ `<script src=http://libs.baidu.com/jquery/1.10.2/jquery.min.js>`
- ▶ `</script>`

2.3.4 jQuery

▶ jQuery 语法

- ▶ jQuery 语法是通过选取 HTML 元素，并对选取的元素执行某些操作。
- ▶ 基础语法：\$(selector).action()
 - ▶ 美元符号定义 jQuery
 - ▶ 选择符 (selector) "查询"和"查找" HTML 元素
 - ▶ jQuery 的 action() 执行对元素的操作

2.3.4 jQuery

▶ jQuery 语法

- ▶ 基础语法: `$(selector).action()`
- ▶ 实例:
- ▶ `$(this).hide()` - 隐藏当前元素
- ▶ `$("p").hide()` - 隐藏所有 `<p>` 元素
- ▶ `$("p.test").hide()` - 隐藏所有 `class="test"` 的 `<p>` 元素
- ▶ `$("#test").hide()` - 隐藏所有 `id="test"` 的元素

2.4 常用Web页面设计开发与框架

▶ 前端组件库:

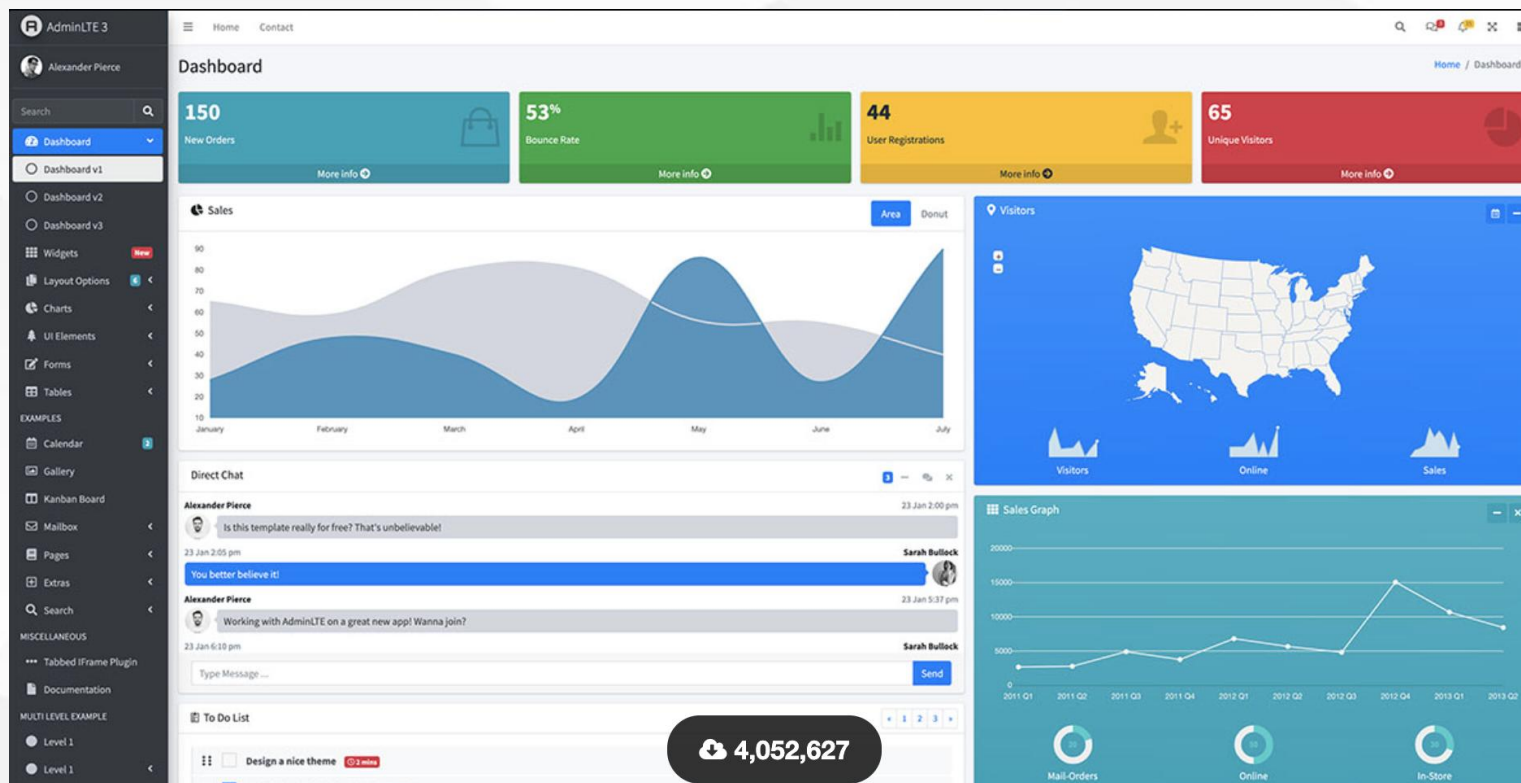
▶ Bootstrap

▶ 入门教程

▶ 在线编辑器

▶ 基于 bootstrap 的轻量级后台模板

▶ AdminLTE实例



2.4 常用Web页面设计开发与框架

▶ 可视化库: highchart、[echart](#)

▶ 前端开发框架:

▶ [React](#)

▶ [Vue](#)

▶ [AngularJS](#)

▶ 企业级开源UI组件库

▶ [ant.design](#)

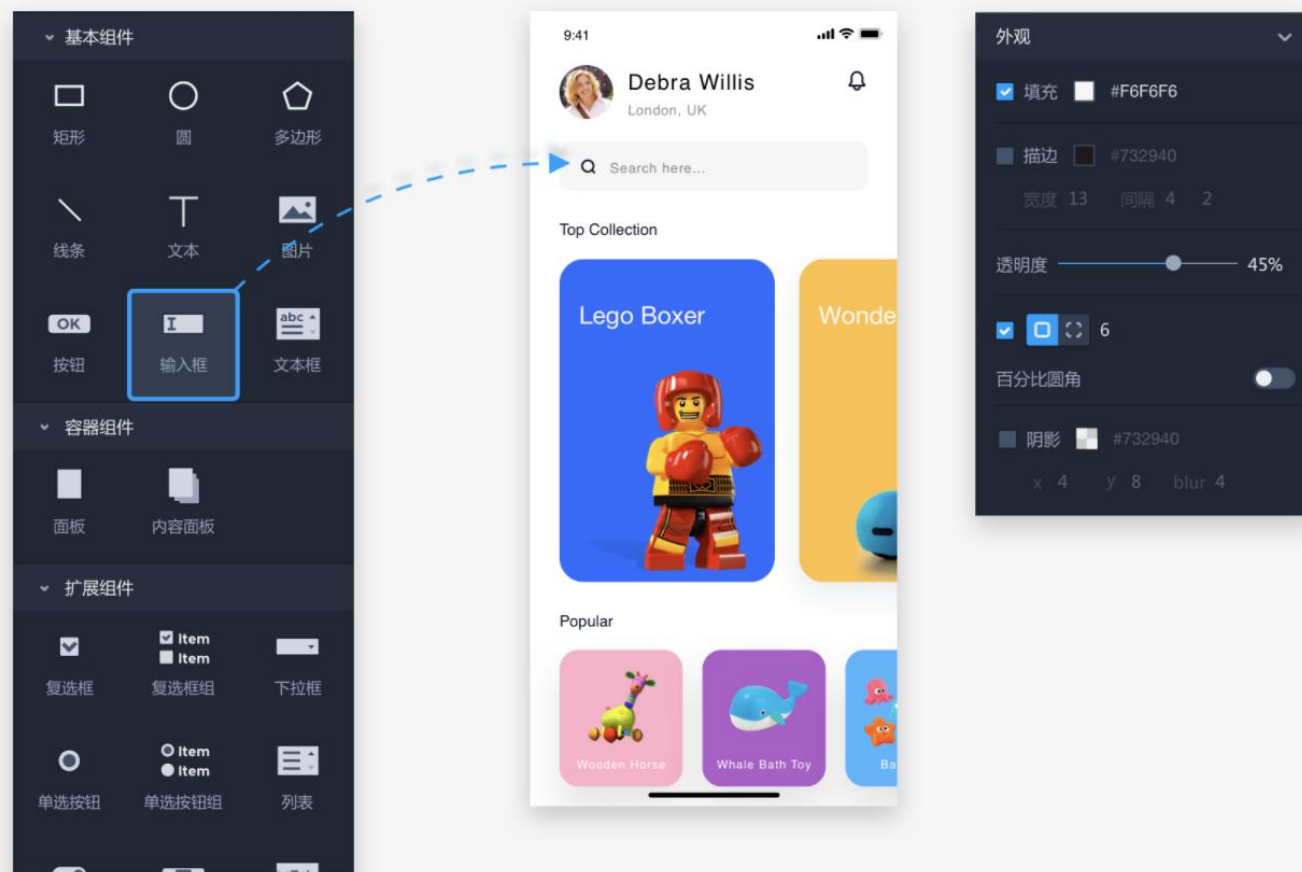
- [Ant Design of React](#) (官方实现)
- [NG-ZORRO - Ant Design of Angular](#)
- [NG-ZORRO-MOBILE - Ant Design Mobile of Angular](#)
- [Ant Design of Vue](#)
- [Ant Design Blazor](#)
- [San UI Toolkit for Ant Design](#)
- [antizer \(ClojureScript\)](#)



2.4 常用Web页面设计开发与框架

▶ UI/原型设计工具

- ▶ Sketch
- ▶ Axure RP
- ▶ [Mockplus](#)
- ▶ [墨刀](#)
- ▶ [即时设计](#)



2.4 常用Web页面设计开发与框架

▶ UI/原型设计工具

<https://ant.design/docs/resources-cn>

这里提供 Ant Design 相关设计资源和设计工具的下载，更多设计资源正在整理和完善中。你可以在这个[地址](#)中反馈对新版本 Sketch Symbols 组件的意见。

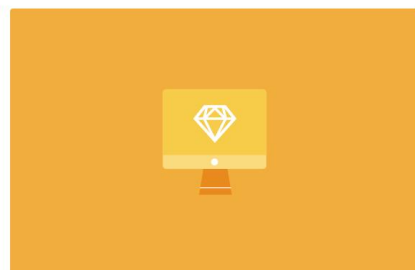
▶ Sketch

▶ Axure RP

▶ [Mockplus](#)

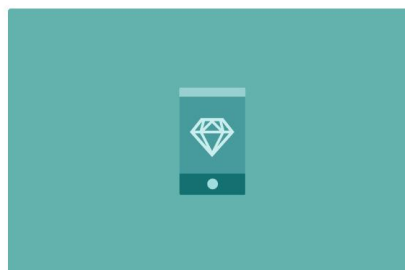
▶ [墨刀](#)

▶ [即时设计](#)



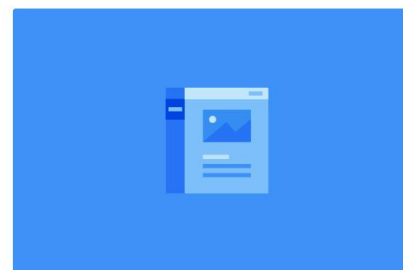
全新 Sketch 组件包

桌面组件 Sketch 模板包



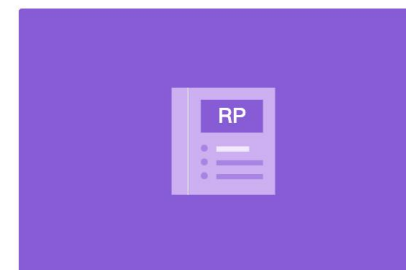
Mobile Components

移动组件 Sketch 模板



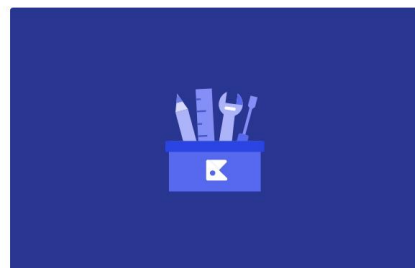
Ant Design Pro

典型页面 + 通用业务模板



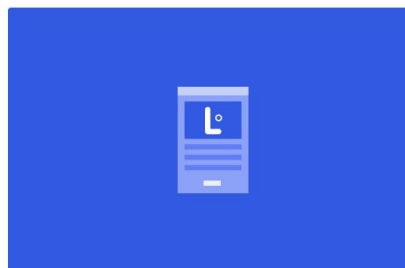
Ant Design Library

一套强大的 Axure 部件库



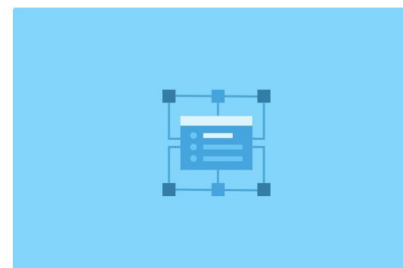
Kitchen

Sketch 工具集



Ant Design Landing

首页模板集



Ant UX

一套页面逻辑原型库



Ant Design 原型 (xiaopiu)

可在线编辑的组件库和交互原型