

学科实践一

第一章 前端开发技术入门

厦门大学信息学院实验教学中心 苏淑文

本章纲要

1. HBuildX与Uni-app
2. HTML
3. CSS
4. JavaScript
5. HTML DOM
6. AJAX
7. 本章作业

一、HBuildX与uni-app

- 数字天堂（北京）网络技术有限公司

www.dcloud.io

- HBuildX, HX, 免费, 20+M, 免安装, 轻如编辑器, 强如IDE的合体版本的软件

- 课堂下载, 解压, 打开HX



- UniApp: 一个使用 **Vue.js** 开发所有前端应用的**框架**, 开发者编写一套代码, 可发布到iOS、Android、Web (响应式)、以及各种小程序 (微信/支付宝/百度/头条/飞书/QQ/快手/钉钉/淘宝)、快应用等**多个平台**。uni-app代码编写, 基本语言包括js、vue、css以及ts、scss等css预编译器。

■ 磨刀不误砍柴工之HX入门教程

- <https://hx.dcloud.net.cn/Tutorial/StartedTutorial>

■ <https://uniapp.dcloud.net.cn/>

- 看视频，初次认识 Uni-app
- 扫二维码，安装移动端不同的应用，体会Hello Uni-app

■ 打开HX，新建“项目”，体会“普通项目”与“uni-app”项目

■ 新建一个“普通项目”中的“基本HTML项目”，观察其目录架构。

■ 新建一个“uni-app”项目中的“默认模板”项目与“Hello uni-app”项目。

■ Uniapp快速入门学习 (**内容有点多**)

- https://blog.csdn.net/weixin_43602502/article/details/121629274

普通项目

uni-app

Wap2App

5+App

IDE插件

新建普通项目

任意文件夹项目，包括非uni-app的web项目、文档目录等。

项目名称

D:/Family/Su/HBuildX

浏览

选择模板



空项目

仅包含一个工程目录



基本HTML项目

包含index.html和css、js、img目录



vue项目(普通模式)

在html中引用vue.js (v3.2.8)



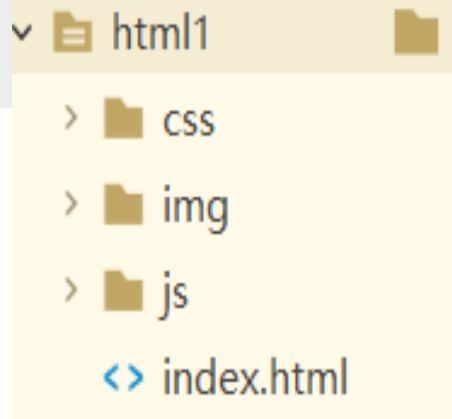
vue3项目(3.2.8)

vue3-cli默认项目 (包含 vite)



vue项目(2.6.10)

vue-cli默认项目 (仅包含 babel)



普通项目

uni-app

Wap2App

5+App

IDE插件

新建uni-app项目

开发一次，同时发布为Android、iOS原生App、小程序、H5等多个平台。[了解更多](#)

项目名称

D:/Family/Su/HBuildX

浏览

选择模板



默认模板
默认模板



Hello uni-app
演示uni-app框架的组件、接口、模板。[详情](#)



Hello uniCloud
演示uniCloud的基本操作。[详情](#)



uni-starter
云端一体应用快速开发基本项目模板。[详情](#)



uni-admin
admin管理系统基础框架。[详情](#)



uni-ui项目
基于uni-app的 UI 组件库。[详情](#)



Hello i18n
uni-app 国际化演示。[详情](#)



新闻/资讯模板
支持PC和Vue的新闻模板。[详情](#)



底部选项卡模板
中间带+号的底部选项卡。[详情](#)



网赚游戏-合成猫
集成uniAD的合成类网赚游戏。[详情](#)

查找模板

去搜索...

U3

- > pages
- > static
- App.vue
- <> index.html
- main.js
- manifest.json
- pages.json
- uni.scss

U1

- > common
- > components
- > hybrid
- > pages
- > platforms
- > static
- > store
- > uni_modules
- > windows
- > wxcomponents
- App.vue
- changelog.md
- <> index.html
- LICENSE
- main.js
- manifest.json
- package.json
- pages.json
- README.md
- <> template.h5.h...
- uni.scss

本章纲要

1. HBuildX与Uni-app
2. HTML
3. CSS
4. JavaScript
5. HTML DOM
6. AJAX
7. 本章作业

二、HTML

- HTML: Hyper Text Markup Language 超文本标记语言
- HTML包括一系列标签。通过这些标签,可以将网络上的文档格式统一,使分散的Internet资源连接为一个逻辑整体。HTML文本是由HTML命令组成的描述性文本, HTML命令可以说明文字, 图形、动画、声音、表格、链接等。
- 1990年创立, 独立于各种操作系统平台,通过专用的浏览器来识别, 并将这些HTML文件“翻译”成可以识别的信息, 即我们所见到的网页。
- 超文本标记语言是标准通用标记语言下的一个应用, 也是一种规范, 一种标准, 它通过标记符号来标记要显示的网页中的各个部分。网页文件本身是一种文本文件

请课后自学http(https)、ftp、smtp、telnet协议极其默认端口相关知识

HTML历史版本

- ①HTML 1.0: 在1993年6月作为互联网工程工作小组(IETF)工作草案发布。
- ②HTML 2.0: 1995年1 1月作为RFC 1866发布, 于2000年6月发布之后被宣布已经过时。
- ③HTML 3.2: 1997年1月14日, W3C推荐标准。 [5]
- ④HTML 4.0: 1997年12月18日, W3C推荐标准。 [5]
- ⑤HTML 4.01 (微小改进): 1999年12月24日, W3C推荐标准。
- ⑥HTML 5: HTML5是公认的下一代Web语言, 极大地提升了Web在富媒体、富内容和富应用等方面的能力, 被喻为终将改变移动互联网的重要推手。Internet Explorer 8及以前的版本不支持。

HTML(不限于HTML)的学习网站

RUNOOB.COM



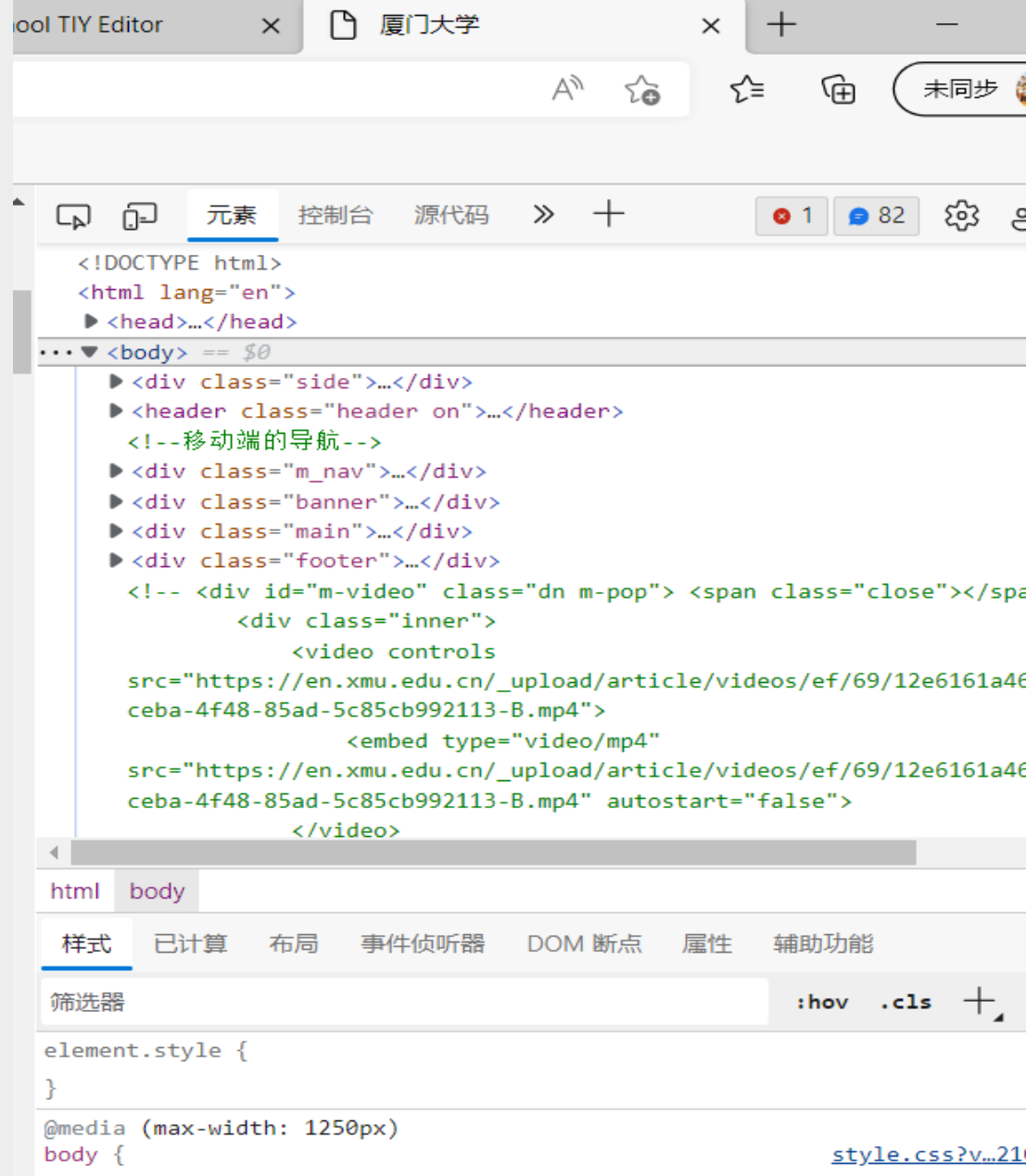
W3school

廖雪峰的官方网站



浏览器端的学习

- 浏览器——》打开一个网页——》右上角…——更多工具——》开发者工具
- 或F12快捷打开



HTML 教程

HTML

HTML5

XHTML

CSS

CSS3

Bootstrap 5

TCP/IP

浏览器脚本

JavaScript

HTML DOM

jQuery

jQuery Mobile

AJAX

JSON

DHTML

E4X

WMLScript

服务器脚本

PHP

SQL

以W3school为例



- 要求至少掌握：HTML基础教程、HTML表单两部分内容

- [HTML 标签参考手册 - 功能排序](http://w3school.com.cn)
(w3school.com.cn)

HTML 元素 (ELEMENT) 语法

- HTML 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间的内容。标签**推荐**使用**小写**。 `<p>This is a paragraph </p>`

- 大多数 HTML 元素可拥有**属性**，名称/值对的形式，在 **开始标签**中定义。推荐小写的属性/属性值。

`<body bgcolor="yellow">`

`<table border="1">`

- 某些 HTML 元素没有内容，被称为**空元素**。空元素在开始标签中关闭（以开始标签的结束而结束）

`
`（`
` 标签定义换行）

`<p>To break
lines
in a
paragraph,
use the br tag.</p>`

HTML 文档由嵌套的 HTML 元素构成

- `<p>` HTML文档段落
- `<body>` HTML文档主体
- `<html>` 元素定义了整个HTML 文档。
- 注释: `<!--`这是一段注释。注释不会在浏览器中显示。`-->`

```
<html>

<body>
<p>This is my first paragraph.</p>
</body>

</html>
```

HTML超链接与锚

- **超文本 (Hypertext)** 的基本特征：可以超链接文档；可以指向其它位置，该位置可以在当前的文档中、局域网中的其他文档，也可以在因特网上的任何位置的文档中。这些文档组成了一个杂乱的信息网。目标文档通常与其来源有某些关联，并且丰富了来源；来源中的链接元素则将这种关系传递给浏览者。
- **超链接 (Hyperlink)**，或者按照标准叫法，称为**锚 (anchor)**，是使用 `<a>` 标签标记的。有两种类型的应用：
 - 文档中创建一个**热点**：用户激活或选中某热点，浏览器会进行链接并自动加载、显示相关内容，或触发某些与因特网服务相关的操作，如发送电子邮件或下载特殊文件等；
 - 在文档中创建一个**标记**，该标记可以被超链接引用。

HTML超链接与锚(2)

<a>超链接可以是一个字，一个词，一组词，或是一幅图像，点击超链接，跳转到新文档或者当前文档中的某位置。

- 通过使用 href 属性 - 创建指向另一个文档的链接

```
<a href="http://www.w3school.com.cn/" target="_blank">Visit  
W3School!</a>
```

- 通过使用 name 属性 - 创建文档内的书签

```
<a name="tips">基本的注意事项 - 有用的提示</a>
```

```
<a href="#tips">有用的提示</a>
```

```
<a href="http://www.w3school.com.cn/html/html_links.asp#tips">有用的提示</a>
```

分区

- `<div>` 定义文档中的分区或节 (division/section)，它把文档分割为独立的、不同的部分。块级元素，div 元素前后自动放置一个换行符。
- div 的属性：`class` 用于元素组（某一类元素）；`id` 用于标识单独的唯一元素。
- 使用 `class` 或 `id` 对 div 进行标识，为 div 添加了合适的语义，且便于进一步使用 `style` 对 div 进行格式化。

```
<html>
<body>

<h3>This is a header</h3>
<p>This is a paragraph.</p>

<div style="color:#00FF00">
  <h3>This is a header</h3>
  <p>This is a paragraph.</p>
</div>

</body>
</html>
```

This is a header

This is a paragraph.

This is a header

This is a paragraph.

```
<!DOCTYPE html>
<html>
<head>
<style>
.cities {
  background-color:black;
  color:white;
  margin:20px;
  padding:20px;
}
</style>
</head>
<body>
<div class="cities">
<h2>London</h2>
<p>London is the capital city of England. It is the
most populous city in the United Kingdom, with a
metropolitan area of over 13 million inhabitants.</p>
</div>
<div class="cities">
<h2>Paris</h2>
<p>Paris is the capital and most populous city of
France.</p>
<p>Within its metropolitan area is one of the largest
population centers in Europe, with over 12 million
inhabitants.</p>
</div>
</body>
</html>
```

London

London is the capital city of England. It is the most populous city in the United Kingdom, with a metropolitan area of over 13 million inhabitants.

Paris

Paris is the capital and most populous city of France.

Within its metropolitan area is one of the largest population centers in Europe, with over 12 million inhabitants.

CLASS

- HTML中的class: 定义元素的类名; 也可以利用它通过 JavaScript 来改变带有指定 class 的 HTML 元素。
- 可以给 HTML 的一个元素赋予多个 class, 中间以空格分开, 从而把若干个 CSS 类合并到一个 HTML 元素。
类名区分大小写。

```
<html>
<head>
<style type="text/css">
h1.intro
{
color:blue;
text-align:center;
}
.important {background-color:yellow;}
</style>
</head>

<body>
<h1 class="intro important">Header 1</h1>
</body>

</html>
```

Header 1

ID

- HTML id 属性用于为HTML 元素指定**唯一的 ID**。一个HTML文档中不能存在多个有相同 ID的元素。id 属性用于指向样式表中的特定样式声明。JavaScript 也可使用它来访问和操作拥有特定 ID 的元素。
- id 的语法是：**#号引出**，后跟一个 id 名（至少一个字符，且**不能包含空白字符**（空格、制表符等，id 名称**对大小写敏感**），并在花括号 {} 中定义 CSS 属性。

CLASS与ID的例子

Class 与 ID 的差异

一个类名可以多个HTML元素使用，而一个id名称只能由页面中的一个 HTML元素使用

我的城市

北京

中国的首都。

上海

中国最大的经济城市。

```
<head>
<style>
/* 设置 id 为 "myHeader" 的元素的样式 */
#myHeader {
    background-color: lightblue;
    color: black;
    padding: 40px;
    text-align: center;
}

/* 设置类名为 "city" 的所有元素的样式 */
.city {
    background-color: tomato;
    color: white;
    padding: 10px;
}
</style>
</head>
<body>
<h1>Class 与 ID 的差异</h1>
<p>一个类名可以多个HTML元素使用，而一个id名称只能由页面中的一个 HTML元素使用</p>
<!-- 拥有唯一 id 的元素 -->
<h1 id="myHeader">我的城市</h1>
<!-- 拥有相同类名的多个元素 -->
<h2 class="city">北京</h2>
<p>中国的首都。</p>
<h2 class="city">上海</h2>
<p>中国最大的经济城市。</p>
</body>
</html>
```

LIST

- 无序列表 unordered list: **ul** (下一级是**li**: list item)
- 有序列表 ordered list : **ol** (下一级是li)
- 自定义列表 definition list: **dl**
 - 下一级是**dt**, 表示自定义列表项
 - 下一级是**dd**, 表示上一级“自定义列表项”的子项

<h4>嵌套列表（含有序表与无序表，注意type定义）：</h4>

<li type="square">咖啡

<li type="circle">茶

红茶

绿茶

<ol type="A">

中国茶

非洲茶

牛奶

全脂牛奶

脱脂牛奶

<h2>一个定义列表：</h2>

<dl>

<dt>计算机</dt>

<dd>用来计算的仪器</dd>

<dt>显示器</dt>

<dd>以视觉方式显示信息的装置</dd>

</dl>

嵌套列表（含有序表与无序表，注意type定义）

■ 咖啡

○ 茶

○ 红茶

○ 绿茶

A. 中国茶

B. 非洲茶

● 牛奶

1. 全脂牛奶

2. 脱脂牛奶

一个定义列表：

计算机

用来计算的仪器

显示器

以视觉方式显示信息的装置

表格 TABLE

- 表格由 `<table>` 标签来定义。
 - 每个表格均有若干行（由 `<tr>` 标签定义; r:row）
 - 每行被分割为若干单元格（由 `<td>` 标签定义;d :data），即数据单元格的内容。数据单元格可以包含文本、图片、列表、段落、表单、水平线、表格等。
 - 表头数据用`<th>`标签定义（加粗效果， h:header）

| 消费项目.... | 一月 | 二月 |
|----------|----------|---------|
| 衣服 | \$241.10 | \$50.20 |
| 化妆品 | \$30.00 | \$44.45 |
| 总计 | \$271.10 | \$94.65 |

```
<table width="400" border="1">
  <tr>
    <th align="left">消费项目....</th>
    <th align="right">一月</th>
    <th align="right">二月</th>
  </tr>
  <tr>
    <td align="left">衣服</td>
    <td align="right">$241.10</td>
    <td align="right">$50.20</td>
  </tr>
  <tr>
    <td align="left">化妆品</td>
    <td align="right">$30.00</td>
    <td align="right">$44.45</td>
  </tr>
  <tr>
    <th align="left">总计</th>
    <th align="right">$271.10</th>
    <th align="right">$94.65</th>
  </tr>
</table>
```

图像IMG

- 在 HTML 中，图像由 `` 标签定义。`` 是**空标签**，它只包含属性，并且**没有闭合标签**。
- **源属性src**：指source，其值表示图像 URL 地址。
- **alt 属性**：用户为图像定义一串预备的可替换的文本。当浏览器无法载入图像时，浏览器将显示替换文本属性，以告诉读者他们失去的信息。

```
<!DOCTYPE HTML>
<html>

<body background="/i/eg_background.jpg">

<p>
一幅图像:

</p>

<p>
一幅动画图像:

</p>

</body>
</html>
```

一幅图像:



一幅动画图像:



表单form

- HTML 表单<form>用于收集不同类型的用户输入。
- 表单元素指的是不同类型的 input 元素、复选框、单选按钮、提交按钮等。 <input> 元素是最重要的表单元素。
- 未设置name的input将不会被提交 (submit)
- <input type="text">
- <input type="password">
- <input type="checkbox">
- <input type="radio">
- <input type="submit">
- <input type="button" onclick="alert('Hello World!')" value="Click Me!">

HTML5 增加了多个新的输入类型:

- color
- date
- datetime
- datetime-local
- email
- month
- number
- range
- search
- tel
- time
- url
- week

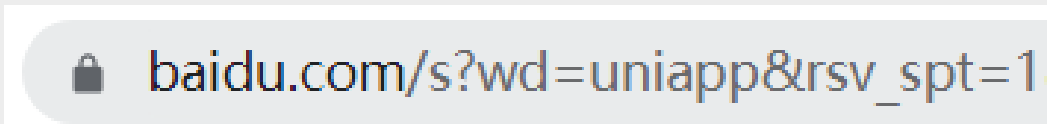
form的method属性

■ Method属性: get与post

- `<form action="/action_page.php" method="post">`

■ 关于 GET 的注意事项:

- 以名称/值对的形式将表单数据追加到 URL
- 对于用户希望将结果添加为书签的表单提交很有用; 适用于非安全数据, 例如 Google 中的查询字符串; 不要使用 GET 发送敏感数据!
- URL 的长度受到限制 (2048 个字符)



■ 关于 POST 的注意事项:

- 将表单数据附加在 HTTP 请求的正文中 (不在 URL 中显示提交的表单数据)
- POST 没有大小限制, 可用于发送大量数据。
- 带有 POST 的表单提交无法添加书签

■ 提示: 如果表单数据包含敏感信息或个人信息, 请务必使用 POST!

form的target属性

```
<form target="value">
```

属性值

| 值 | 描述 |
|------------------|---------------|
| _blank | 在新窗口中打开。 |
| _self | 默认。在相同的框架中打开。 |
| _parent | 在父框架集中打开。 |
| _top | 在整个窗口中打开。 |
| <i>framename</i> | 在指定的框架中打开。 |

```
<!DOCTYPE html>
<html>
<body>

<form action="/demo/demo_form.asp, method="post">
<fieldset>
<legend>Personal information:</legend>
Name:<br>
<input type="text" name="firstname" value="Mickey">
<br>
Gender:<br>
<input type="radio" name="sex" value="male"
checked>Male
<br>
<input type="radio" name="sex" value="female">Female
<br>
<input type="submit" value="Submit">
</fieldset>
</form>
<p>**点击提交, 表单数据将发送到demo_form.asp 页面。**</p>

</body>
</html>
```

Personal information:

Name:

Gender:

☒ Male

☐ Female

点击提交, 表单数据将发送到demo_form.asp 页面.

HTML 头部元素

- `<head>` 元素：描述了文档的各种属性和信息。包括文档的标题；引用脚本、指示浏览器在哪里找到样式表、提供元信息等。绝大多数头部里的数据都不会真正作为内容显示给读者。常见标签有 `<title>` `<base>`, `<link>`, `<meta>`, `<script>`, `<style>`。
- `<title>` 定义网页标题，它是 head 部分中唯一必需的元素。
- `<base target="_blank" />`
- `<link rel="stylesheet" type="text/css" href="/html/csstest1.css" >` rel: relationship; stylesheet: 样式表; href: Hypertext Reference,超文本引用，用来建立当前元素和文档之间的链接。
- `<meta>` 元素可提供有关页面的元信息（meta-information），如针对搜索引擎和更新频度的描述和关键词。
- `<script>` 标签用于定义客户端脚本，如 JavaScript；它既可以包含脚本语句，也可以通过 src 属性指向外部脚本文件。

本章纲要

1. HBuildX与Uni-app
2. HTML
3. CSS
4. JavaScript
5. HTML DOM
6. AJAX
7. 本章作业

三、CSS

- 层叠样式表(**C**ascading **S**tyle **S**heets),又称: 级联^样式^表。(网页是个多层结构, css可以分别为网页的每一层设置样式, 我们最终能看见的是最上面一层, 那么, 什么是CSS的层次?)
- 用于定义html文档的样式 (外观), 是页面表现的基础, 可以控制布局, 控制元素的渲染

HTML 与 CSS 区别

- HTML是网页的结构， CSS是网页的样式。
- HTML由围绕内容的标签组成， CSS由选择器和声明块组成
- HTML文件可以包含CSS代码， 而CSS样式表不能包含HTML代码。
- 注释格式不同； （css注释以 /* 开始， 以 */ 结束：
可以在代码中的任何位置添加注释： 可横跨多行。

样式的使用

- 外部样式表：样式定义保存成CSS文件，可以应用到多个页面，通过link引入
- 内部样式表：单个文件使用。
- 内联样式：应用在个别元素上

```
<head>
```

```
<style type="text/css">  
body {background-color: red}  
p {margin-left: 20px}  
</style>
```

```
</head>
```

```
<head>
```

```
<link rel="stylesheet" type="text/css" href="mystyle.css">
```

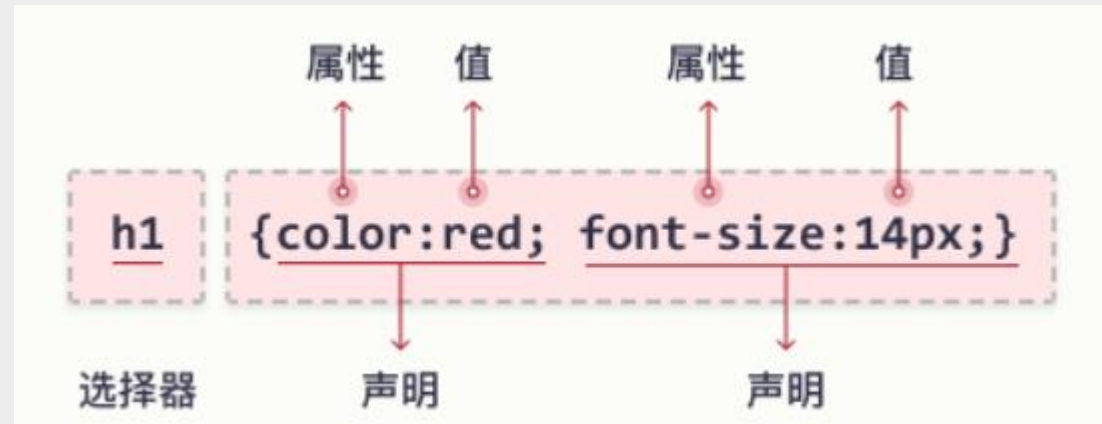
```
</head>
```

```
<p style="color: red; margin-left: 20px">
```

```
This is a paragraph
```

```
</p>
```

CSS语法



- CSS 规则集（rule-set）由选择器和声明块组成
- 选择器：“查找”（或选取）设置样式的 HTML 元素。
- 声明块（用花括号括起来）包含一条或多条用分号分隔的声明。每条声明都包含一个 CSS 属性名称和一个值，以冒号分隔。

CSS 选择器分类

- 简单选择器（根据name、id、class来选取元素）
- 组合器选择器（根据它们之间的特定关系来选取元素）
- 伪类选择器（根据特定状态选取元素）
- 伪元素选择器（选取元素的一部分并设置其样式）
- 属性选择器（根据属性或属性值来选取元素）

CSS简单选择器

- https://www.w3school.com.cn/css/css_selectors.asp

| 选择器 | 例子 | 例子描述 |
|---------------------------|------------|--------------------------|
| <u>.class</u> | .intro | 选取所有 class="intro" 的元素。 |
| <u>#id</u> | #firstname | 选取 id="firstname" 的那个元素。 |
| <u>*</u> — | * | 选取所有元素。 |
| <u>element</u> | p | 选取所有 <p> 元素。 |
| <u>element,element,..</u> | div, p | 选取所有 <div> 元素和所有 <p> 元素。 |

组合器选择器

- CSS 组合器：解释选择器之间关系的某种机制。CSS 中有四种不同的组合器：
 1. 后代选择器 (空格)：匹配属于指定元素后代的**所有**元素。（管”子”也管”孙”）
 2. 子选择器 (>)：匹配属于指定元素子元素的所有元素。（管”子”不管”孙”）
 3. 相邻兄弟选择器 (+)：匹配所有作为指定元素的**相邻同级**的元素。（唯一！只能是”弟”不能是”兄”）
 4. 通用兄弟选择器 (~)：匹配属于指定元素的同级所有元素。（可能多项，只能是”弟”不能是”兄”）

`div p {……}` `div > p {……}` `div +p {……}` `div ~p {……}`

- 伪类选择器： 与 伪元素选择器 ::

https://www.w3school.com.cn/css/css_pseudo_classes.asp

- 属性选择器：

https://www.w3school.com.cn/css/css_attribute_selectors.asp

HTML与CSS案例1

■ 某电子商城login界面

- login.html login.css



登录商城 [新用户注册](#)

请输入您的用户名 

用户名不能为空!

请输入您的密码 

密码不能为空!

☐ 自动登录 [忘记密码?](#)

登录

HTML与CSS案例2

■ CSS 单选按钮

- <https://codepen.io/AngelaVelasquez/pen/DGeErL>

Hey! Read Me...

This is an example of what you can do with pure css to style radio buttons or checkboxes. If you are looking a more flexible & morden way to style radio buttons on forms, take a look a about it.

Your favorite thing in the world:

☒ Pizza

☐ Bacon

☐ Cats

CODEPEN.IO

- CodePen is a **social development environment** for front-end designers and developers. Build and deploy a website, show off your work, build test cases to learn and debug, and find inspiration.

Reveal Text
alphardex + Follow

Settings Sign Up Log In

HTML

```
1 <div class="reveal">sword art online</div>
```

CSS (SCSS)

```
28 &::before,  
29 &::after {  
30   position: absolute;  
31   content: "";  
32   top: 0;  
33   bottom: 0;  
34   width: 2px;  
35   height: 100%;  
36   background: white;  
37   opacity: 0;  
38   transform: scale(0);  
39 }  
40  
41 &::before {  
42   left: 50%;
```

JS (TypeScript)

```
1 let duration = 0.8;  
2 let delay = 0.3;  
3 let revealText = document.querySelector(".reveal");  
4 let letters = revealText.textContent.split("");  
5 revealText.textContent = "";  
6 let middle = letters.filter(e => e !== " ").length /  
7 2;  
8 letters.forEach((letter, i) => {  
9   let span = document.createElement("span");  
10  span.textContent = letter;  
11  span.style.animationDelay = `${delay + Math.abs(i -  
12  middle) * 0.1}s`;  
13  revealText.append(span);  
14  });
```

| SWORD ART ONLINE |

Console Clear X

本章纲要

1. HBuildX与Uni-app
2. HTML
3. CSS
4. JavaScript
5. HTML DOM
6. AJAX
7. 本章作业

JAVASCRIPT

- JavaScript 是 web 开发者必学的三种语言之一：
 - **HTML** 定义网页的**内容**
 - **CSS** 规定网页的**布局**
 - **JavaScript** 对网页的**行为**进行编程
- JavaScript (简称JS)和 Java 是完全不同的语言，不论是概念还是设计。
- JavaScript 在 1995 年由 Brendan Eich 发明，并于 1997 年成为一部 ECMA 标准。

- ECMA (European Computer Manufacturers Association) 欧洲计算机制造商协会，这个组织目标是评估，开发和认可电信和计算机标准。后改名为ECMA国际。
- ECMAScript(**ES**): 由ECMA国际通过ECMA-262标准化的脚本程序设计语言。
- **ES**: 脚本语言的规范，JS是ES的一种实现，ES新特性其实指的就是JavaScript的新特性。
- 不同浏览器支持的ES版本不同
- JavaScript 是 HTML 中的默认脚本语言。

- JS 程序：一系列的JS语句构成，在 HTML 中， JS 程序由 web 浏览器执行。
- JS 语句构成：值、运算符、表达式、关键词和注释。
- **分号**分隔 JS 语句，允许一行多句，允许一句多行
- JS 语句可以用**花括号**组合在**代码块**中，代码块指一同执行的语句。
- **注释**：单行注释双斜杠 // 多行注释/* 与 */
- JS 标识符**对大小写敏感**；JS程序员倾向于使用以小写字母开头的驼峰大小写， JS 中不能使用连字符，连字符为减法预留。

关键词

break

continue

debugger

do ... while

for

function

if ... else

return

switch

try ... catch

var

JS 数据类型:数值、字符串值、数组、对象

- 数据类型包含数值、字符串值、数组、对象等。JS 拥有动态类型。这意味着相同变量可用作不同类型
- **数值**只有一种类型； 数值和字符串运算时， 数值+字符串1=字符串2
- **字符串**被包围在双引号或单引号中。数值不用引号。
- **布尔**值两个值： true 或 false
- **数组**用**方括号**， 数据由逗号分隔， 数组索引基于零。在 JavaScript 中， 数组即对象
 - `var cars = ["Porsche", "Volvo", "BMW"];`

JS 数据类型:数值、字符串值、数组、对象(2)

- **对象**用**花括号**，对象属性是 name:value 对，由逗号分隔。
 - `var person = {firstName:"Bill", lastName:"Gates", age:62, eyeColor:"blue"};`
- 有两种不同的方法来访问对象属性。
 - 使用 `person.property` 或 `person["property"]`。
- 访问对象的方法： `objectName.methodName()`
- **null**是空对象，可以设置值为null来清空对象。 `var person = null;`

```
var person = {
  firstName: "Bill",
  lastName : "Gates",
  id       : 678,
  fullName : function() {
    return this.firstName + " " + this.lastName;
  }
};
```

JS的类class (例子)

```
<!DOCTYPE html>
<html>
<body>
<h1>JavaScript 类方法</h1>
<p id="demo"></p>
<script>
```

```
class Car {
  constructor(name, year) {
    this.name = name;
    this.year = year;
  }
  age(x) {
    return x - this.year;
  }
}
```

```
let date = new Date();
let year = date.getFullYear();

let myCar = new Car("Citroen ", 2007);
document.getElementById("demo").innerHTML=
"My car is " + myCar.age(year) + " years old.";

</script>
</body>
</html>
```

变量

- 变量声明 var。可以在一条语句中声明许多变量，以 var 作为语句的开头，并以逗号分隔变量：
 - var person = "Bill Gates", carName = "porsche", price = 15000;
- 通过 var 关键词声明的变量没有“块作用域”。在块 {} 内声明的变量可以从块之外进行访问。（但 let 有）

JavaScript 赋值运算符

赋值运算符向 JavaScript 变量

| 运算符 |
|-----|
| = |
| += |
| -= |
| *= |
| /= |
| %= |

JavaScript 算数运算符

算数运算符用于对数字执行算数

| 运算符 |
|-----|
| + |
| - |
| * |
| / |
| % |
| ++ |
| -- |

函数

- function, 很常规
- 有些函数仅包含关键字function(), 不需要函数名
- 可以把函数当变量值直接使用
 - var text = "The temperature is " + toCelsius(77) + " Celsius";

```
var x = myFunction(7, 8);  
  
function myFunction(a, b) {  
    return a * b;  
}
```

事件EVENT

- HTML 事件是发生在 HTML 元素上的“事情”。
- 当在 HTML 页面中使用 JavaScript 时，JavaScript 能够“应对”这些事件。
 - `<button onclick="this.innerHTML=Date()">时间是? </button>`
 - `<button onclick="displayDate()">现在的时间是? </button>`

常见的 HTML 事件

下面是一些常见的 HTML 事件：

| 事件 | 描述 |
|-------------|-------------------|
| onchange | HTML 元素已被改变 |
| onclick | 用户点击了 HTML 元素 |
| onmouseover | 用户把鼠标移动到 HTML 元素上 |
| onmouseout | 用户把鼠标移开 HTML 元素 |
| onkeydown | 用户按下键盘按键 |
| onload | 浏览器已经完成页面加载 |

更完整的列表：[W3School JavaScript 参考手册 HTML DOM 事件](#)。

JS的四种输出

- 更改 HTML 元素的 innerHTML 属性是在 HTML 中显示数据的常用方法。
 - `document.getElementById(id).innerHTML = 10`
- 用于测试的 `document.write()`，HTML 文档完全加载后使用 `document.write()` 将删除所有已有的 HTML
 - `<button onclick="document.write(5 + 6)">试一试</button>`
- 使用警告框来显示数据 `window.alert(5 + 6);`
- `console.log()` 方法来显示数据，F12 激活浏览器控制台，可看到写入的日志数据

JS 使用（外部JS文件与内部JS代码）

- 脚本可被放置与 HTML 页面的 <body> 或 <head> 部分，在 <script> 标签内定义或引入。
- 把脚本置于 <body> 元素的底部，可改善显示速度，因为脚本编译会拖慢显示
- JavaScript 函数和事件常内嵌在html中
- JavaScript 文件的文件扩展名是 .js，外部脚本不能包含 <script> 标签。
 - <script src="myScript.js"></script>
 - 分离了 HTML 和代码
 - 使 HTML 和 JavaScript 更易于阅读和维护
 - 已缓存的 JavaScript 文件可加速页面加载

JSON

- JSON: JavaScript Object Notation; 是轻量级的数据交换格式;它独立于编程语言,“自描述的”且易于理解
- JSON 的语法是来自 JavaScript 对象符号的语法,但 JSON 格式是纯文本。读取和生成 JSON 数据的代码可以在任何编程语言编写的。

JSON (2)

■ JSON 语法规则

- 数据是名称/值对
- 数据由逗号分隔
- 花括号保存对象
- 方括号保存数组
- 数组能包含对象

```
<body>

<h1>用 JSON 字符串创建对象</h1>

<p id="demo"></p>

<script>
var text = '{"employees":[" +
'{"firstName":"Bill","lastName":"Gates" },' +
'{"firstName":"Steve","lastName":"Jobs" },' +
'{"firstName":"Elon","lastName":"Musk" }]]';

// JS 的内建函数 JSON.parse() 来把这个字符串转换为对象
obj = JSON.parse(text);

document.getElementById("demo").innerHTML =
obj.employees[1].firstName + " " + obj.employees[1].lastName;
</script>

</body>
```

本章纲要

1. HBuildX与Uni-app
2. HTML
3. CSS
4. JavaScript
5. HTML DOM
6. AJAX
7. 本章作业

HTML DOM (文档对象模型)

- 当网页被加载时，浏览器会创建页面的文档对象模型 (Document Object Model: DOM) 。
- DOM 是W3C 标准。“W3C 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、更新文档的内容、结构和样式。”
- **HTML DOM 是关于如何获取、更改、添加或删除 HTML 元素的标准。**
- DOM是 HTML 的标准**对象模型**和**编程接口**。它定义了：
 - 作为**对象**的 HTML 元素
 - 所有 HTML 元素的**属性**
 - 访问所有 HTML 元素的**方法**
 - 所有 HTML 元素的**事件**

DOM的应用

- innerHTML 属性可用于获取或替换 HTML 元素的内容。
`document.getElementById(name)`
- `document.getElementsByTagName(name)`
- `document.getElementsByClassName(name)`
- `element.attribute = new value`
- `element.style.property = new style`
- `document.appendChild(element)`
- `document.getElementById(id).onclick = function(){code}`
- `document.anchors`
- `document.images`

DOM例子

```
<!DOCTYPE html>
<html>
<body>

<h1>通过标签名查找 HTML 元素</h1>

<div id="main">
<p>DOM 很有用。</p>
<p>本例演示 <b>getElementsByName</b> 方法。</p>
</div>

<p id="demo"></p>

<script>
var x = document.getElementById("main");
var y = x.getElementsByTagName("p");
document.getElementById("demo").innerHTML =
'“main”中的第一段（索引 0）是：' + y[0].innerHTML;
</script>

</body>
</html>
```

通过标签名查找 HTML 元素

DOM 很有用。

本例演示 **getElementsByTagName** 方法。

“main” 中的第一段（索引 0）是：DOM 很有用。

一个JS综合应用的例子

- 源码见 `messagewall.html`

表白墙

输入后点击提交, 将会把消息显示在墙上

谁:

对谁:

说什么:

提交

韩梅梅对李雷说: 今晚月色真美!

李雷对韩梅梅说: 风也温柔

本章纲要

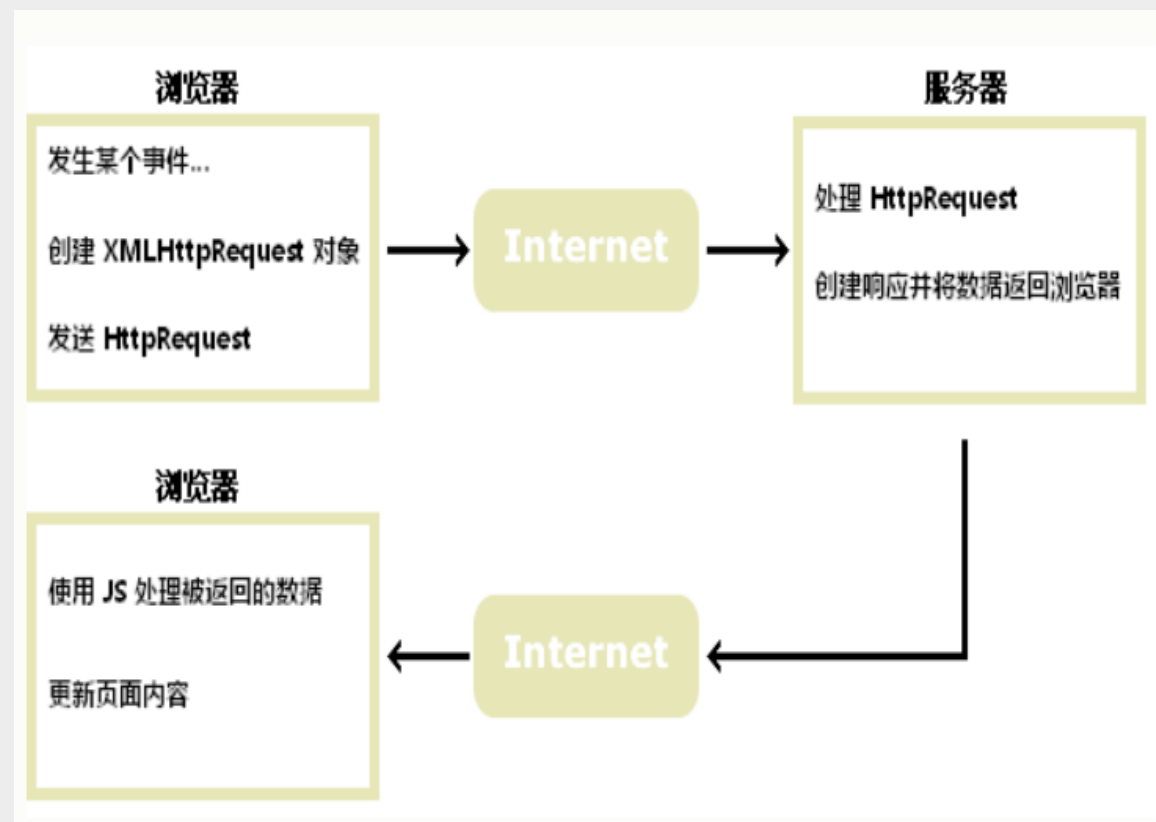
1. HBuildX与Uni-app
2. HTML
3. CSS
4. JavaScript
5. HTML DOM
6. AJAX
7. 本章作业

AJAX

- AJAX = **A**synchronous **J**avaScript **A**nd **X**ML; AJAX 组合了：
 - 浏览器内建的 XMLHttpRequest 对象（从 web 服务器请求数据）
 - JavaScript 和 HTML DOM
 - Ajax 应用程序可能使用 XML、纯文本或 JSON 文本来传输数据，
- Ajax 允许通过与场景后面的 Web 服务器交换数据来异步以局部更新网页，而不需要重新加载整个页面。

AJAX 如何工作

1. 网页中发生一个事件（页面加载、按钮点击等）
2. 由 JavaScript 创建 XMLHttpRequest 对象
3. XMLHttpRequest 对象向 web 服务器发送请求
4. 服务器处理该请求
5. 服务器将响应发送回网页
6. 由 JavaScript 读取响应
7. 由 JavaScript 执行正确的动作（比如更新页面）



AJAX 例子

保存 XMLHttpRequest 的状态。

readyState

- 0: 请求未初始化
- 1: 服务器连接已建立
- 2: 请求已收到
- 3: 正在处理请求
- 4: 请求已完成且响应已就绪

返回请求的状态号

status

- 200: "OK"
- 403: "Forbidden"
- 404: "Not Found"

```
<!DOCTYPE html>
<html>
<body>

<h1>XMLHttpRequest 对象</h1>

<button type="button" onclick="loadDoc()">请求数据</button>

<p>单击该按钮几次以查看时间是否已更改，或文件是否已缓存。</p>

<p id="demo"></p>

<script>
function loadDoc() {
    var xhttp = new XMLHttpRequest();
    xhttp.onreadystatechange=function() {
        if (this.readyState == 4 && this.status == 200) {
            document.getElementById("demo").innerHTML = this.responseText;
        }
    };
    xhttp.open("GET", "/demo/demo_get.asp?t=" + Math.random(), true);
    xhttp.send();
}
</script>

</body>
</html>
```

本章纲要

1. HBuildX与Uni-app
2. HTML
3. CSS
4. JavaScript
5. HTML DOM
6. AJAX
7. 本章作业

本章作业

- 作业一 基本掌握HX的使用方法（界面、菜单、快捷方式）
- 为未来大学本科毕业的自己提前撰写一份简历。要求：
 - 在HX上完成，
 - 需用到课程中涉及的大部分html与css内容（包含表格、链接、图片、音乐等元素）。
 - 不要贴个人真实照片；可以对未来的自己多点美好的心愿。
 - 请提交 html、css源文件以及展示的html截图。
 - 作业截至时间：2023.9.28