

静态网页—HTML5 静的ウェブページ

- Web开发概述 ウェブ開発概要
- HTML概述 HTML概要
- 创建HTML文件 HTMLファイルの作成
- HTML5基本语法 HTML5の基本文法



1

Web开发概述

ウェブ開発概要

2

HTML概述

HTML概要

3

创建HTML文件

HTMLファイルの作成

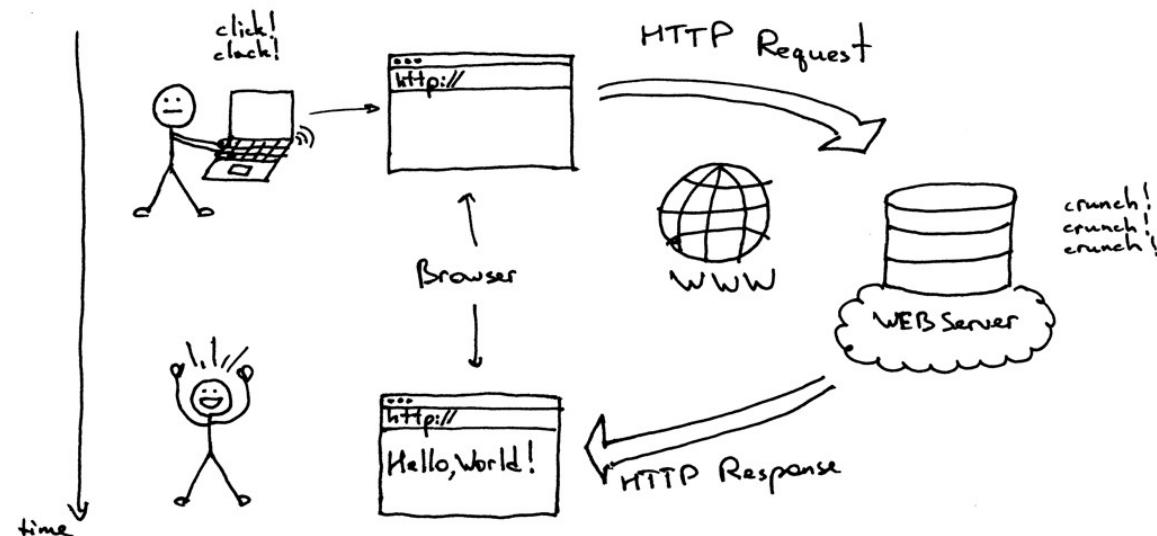
4

HTML5基本语法(重点)

HTML5の基本文法

网络通信原理 [ネットワークの仕組み]

- Web服务[Webサービス]通过超文本传输协议(HTTP)向用户提供多媒体信息[マルチメディア]。所提供的信息的基本单位是网页[ページ], 每个网页包含文字、图片、音频、影视等多种信息;另外, 网页上还可以加入超链接[ハイパーリンク], 使用户能够通过这些超链接[ハイパーリンク]直接访问另外一个页面。



Web服务器[ウェブサーバー]

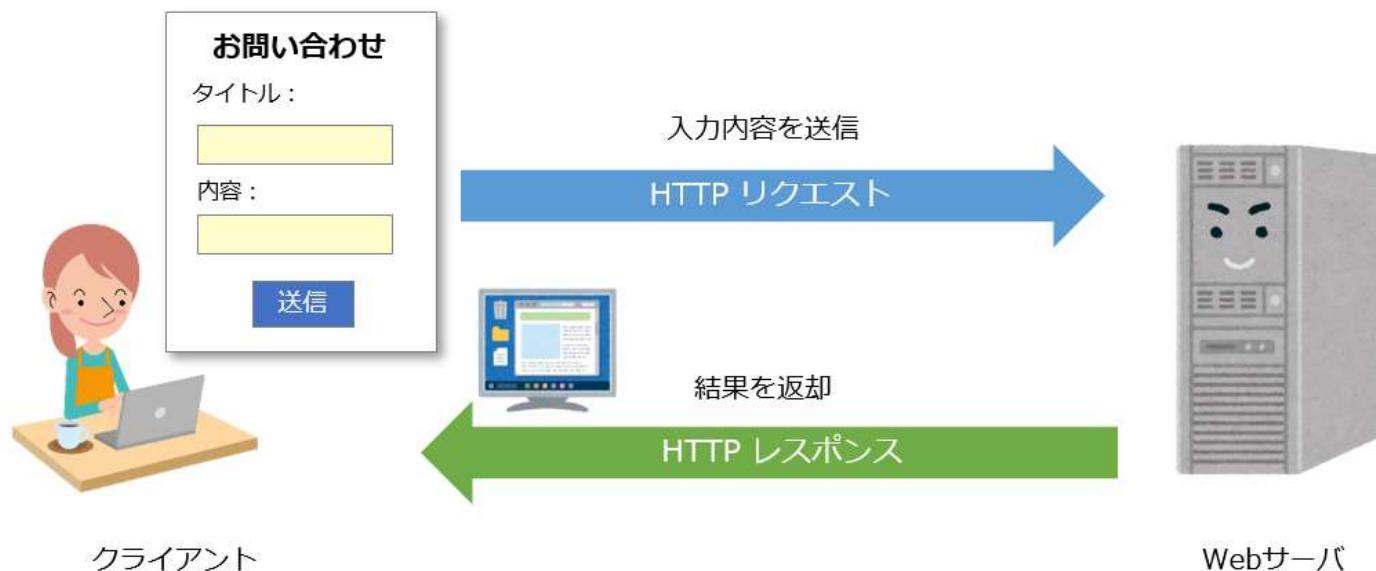
- 一台负责提供网页的电脑，主要是各种编程语言构建而成，通过超文本传输协议（HTTP）传给客户端[クライアント]（一般是指网页浏览器[ウェブブラウザー]）

以下是由W3Techs提供最新的网页服务器在网络上的市场占有率
统计数据：[1]

产品	供应商	占有比率
Apache	Apache	48.5%
nginx	NGINX, Inc.	35.4%
IIS	Microsoft	10.8%
LiteSpeed Web Server	LiteSpeed Technologies	2.9%
GWS	Google	1.1%

超文本传输协议 (HTTP)

- HTTP (HyperText Transfer Protocol) 是一个客户端[クライアント] (用户) 和服务端[サーバー] (网站) 之间请求[リクエスト]和应答[レスポンス]的标准。通过使用网页浏览器[ウェブブラウザー]、网络爬虫[クローラ]或者其它的工具，客户端发起一个HTTP请求到服务器上指定端口[ポート] (默认端口为80)。



静态网页 [静的ウェブページ] VS 动态网页 [動的ウェブページ]

- 静态网页：无用户交互，他主要指的是网页中没有程序代码，只有HTML。不管任何人任何时间打开的页面的内容都是不变的。**
- 动态网页：有用户交互，网页文件中除了HTML标签以外，还包括一些特定功能的程序代码，这些代码可以使得浏览器和服务器可以交互，所以服务器端根据客户的不同请求动态的生成网页内容。**

	静态网页	动态网页
内容	网页内容固定	网页内容动态生成
后缀	.htm;.html等	.asp .shtm .php .cgi等
优点	无需系统实时生成 网页风格灵活多样	日常维护简单 更改结构方便 交互性能强
缺点	交互性能较差 日常维护繁琐	需要大量的系统资源合成网页
数据库	不支持	支持

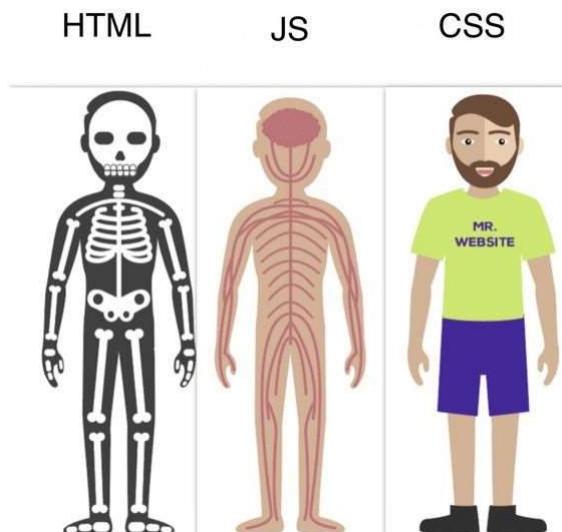


动态网页例子：商品加入购物车

静态网页结构—HTML, CSS, JS

- HTML是用于创建网页的标准标签语言[マークアップ言語]。 (并不属于编程语言[プログラミング言語])
- CSS是描述HTML文档样式的语言，CSS描述了应如何显示HTML元素。
- JavaScript是HTML和Web的编程语言。
- 三者关系：HTML定义网页内容；CSS指定网页的布局；使用JavaScript编写网页行为。

- 例如：
- HTML: 设置文字的内容...
- CSS: 设置文字的大小，颜色，字体...
- JavaScript: 点击事件响应，逻辑运算...

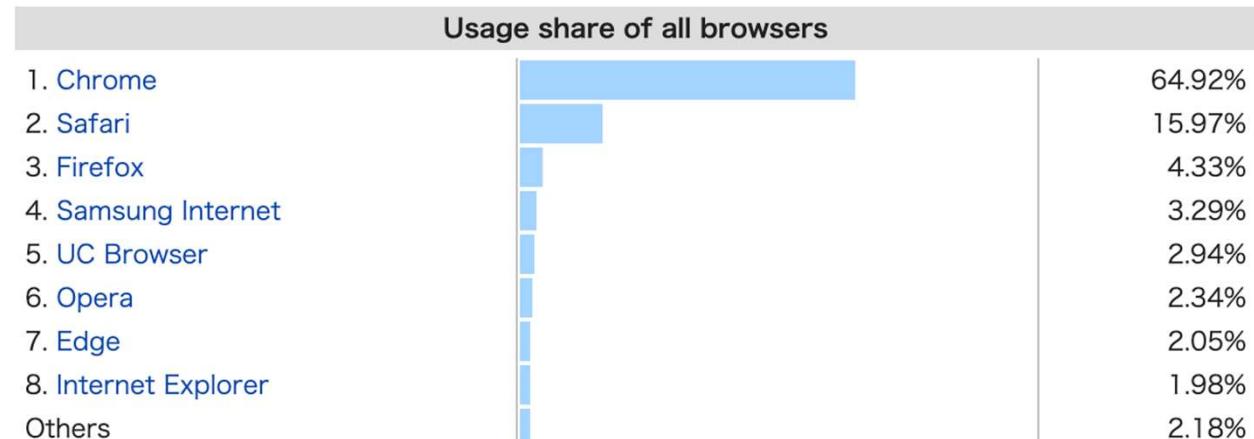


什么是网页浏览器[ウェブブラウザー]

网络浏览器（Chrome, Edge, Firefox, Safari）的目的是读取HTML文档并正确显示它们。

浏览器不显示HTML标签，而是使用它们来确定如何显示文档。

常用浏览器用户占比：



Q & A

You Have Questions
We Have Answers



1 Web开发概述
ウェブ開発概要

2 HTML概述
HTML概要

3 创建HTML文件
HTMLファイルの作成

4 HTML5基本语法(重点)
HTML5の基本文法

什么是HTML

- 超文本标签语言
- 描述了网页的结构
- 由一系列元素组成
- HTML元素决定了浏览器如何显示内容

```
<!DOCTYPE html>
<html>
<head>
<title>Page Title</title>
</head>
<body>

<h1>This is a Heading</h1>
<p>This is a paragraph.</p>

</body>
</html>
```

什么是HTML元素？

- HTML元素由开始标签[タグ]<>, 中间的一些内容和结束标签[タグ]</>定义:

<标签名>内容在这里... </标签名>

HTML 元素是从开始标记到结束标记的所有内容:

< h1 >我的第一个标题 < / h1 >

< p >我的第一段。< / p >

HTML代码范例

```
<!DOCTYPE html>          <!DOCTYPE html>声明定义，这个文件是一个HTML5文档  
<html>                  <html>元素是一个HTML页面的根元素[ルート要素]  
<head>                  <head>元素包含HTML页面的元信息[メタ情報]  
<title>Page Title</title>  <title>元素指定用于HTML页面的标题（它在浏览器的标  
题栏或在页面的选项卡显示）  
</head>  
<body>                  <body>元素定义了文档的身体，并且对于所有的可见内  
容，诸如标题，段落，图像，超链接[ハイパーリンク]，  
表格，列表等的容器  
<h1>My First Heading</h1>  <h1>元素定义了一个大的标题  
<p>My first paragraph.</p>    <p>元素定义了一个段落  
  
</body>  
</html>
```

HTML页面结构

- 以下是HTML页面结构的可视化：

```
<html>
```

```
  <head>
```

```
    <title>Page title</title>
```

```
  </head>
```

```
  <body>
```

```
    <h1>This is a heading</h1>
```

```
    <p>This is a paragraph.</p>
```

```
    <p>This is another paragraph.</p>
```

```
  </body>
```

```
</html>
```

HTML历史

Year	Version
1989	Tim Berners-Lee invented www
1991	Tim Berners-Lee invented HTML
1993	Dave Raggett drafted HTML+
1995	HTML Working Group defined HTML 2.0
1997	W3C Recommendation: HTML 3.2
1999	W3C Recommendation: HTML 4.01
2000	W3C Recommendation: XHTML 1.0
2008	WHATWG HTML5 First Public Draft
2012	<u>WHATWG HTML5 Living Standard</u>
2014	<u>W3C Recommendation: HTML5</u>
2016	W3C Candidate Recommendation: HTML 5.1
2017	<u>W3C Recommendation: HTML5.1 2nd Edition</u>
2017	<u>W3C Recommendation: HTML5.2</u>

我们学习最新版本HTML5

Q & A

You Have Questions
We Have Answers



1 Web开发概述
ウェブ開発概要

2 HTML概述
HTML概要

3 创建HTML文件
HTMLファイルの作成

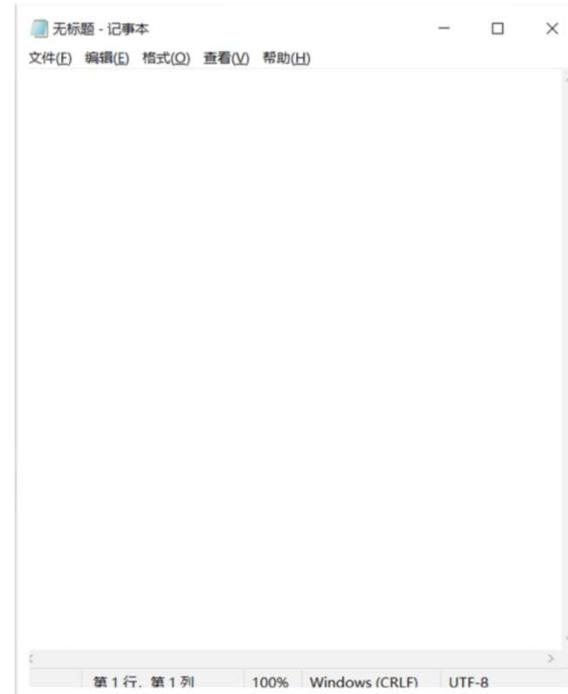
4 HTML5基本语法(重点)
HTML5の基本文法

HTML 编辑器

- 学习HTML只需一个简单的文本编辑器即可。
- 也可以使用专业的编辑器如VSCode、Sublime来创建和修改HTML网页。
- 请按照以下步骤使用记事本或TextEdit创建你的第一个网页。

步骤1：打开记事本（PC）

- Windows 8或更高版本：
打开开始屏幕（屏幕左下方的窗口符号）。键入记事本。
- Windows 7或更早版本：
打开开始 > 程序> 附件> 记事本



步骤1：打开TextEdit (Mac)

- 打开Finder(访达) > 应用程序 > TextEdit
- 同时更改一些首选项，以使应用程序正确保存文件。在偏好设置 (左上角) > 格式> 选择“纯文本” (英语版本：Preferences > Format > 选择 "Plain Text")
- 然后在“打开并保存” (Open and save) 下，选中“将HTML文件显示为HTML代码而不是格式化文本” (Display HTML files as HTML code instead of formatted text) 复选框。
- 然后创建一个新文件以放置代码。

步骤2：编写一些HTML

- 将以下HTML代码编写到记事本中：

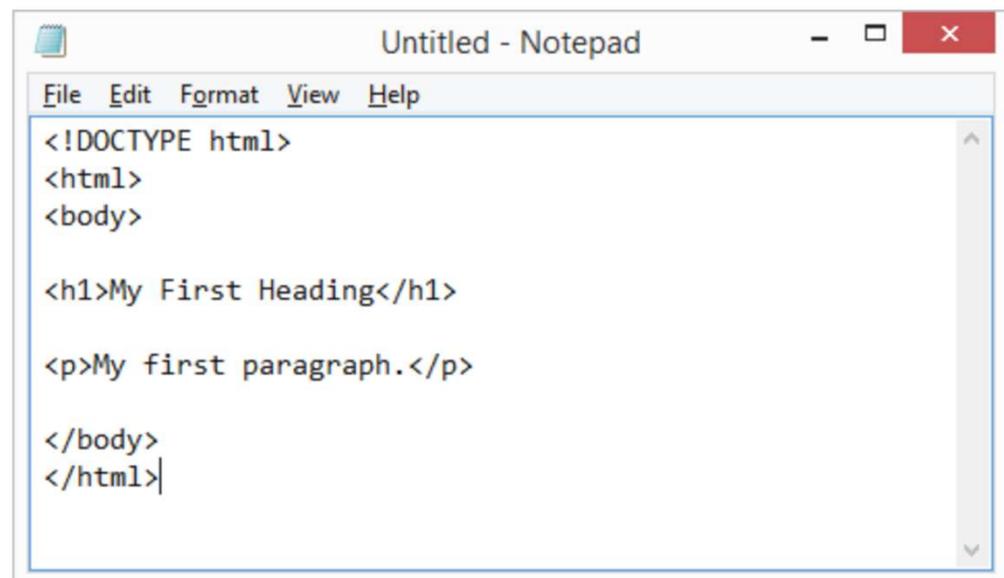
```
<!DOCTYPE html>
<html>
<body>

<h1>My First Heading</h1>

<p>My first paragraph.</p>

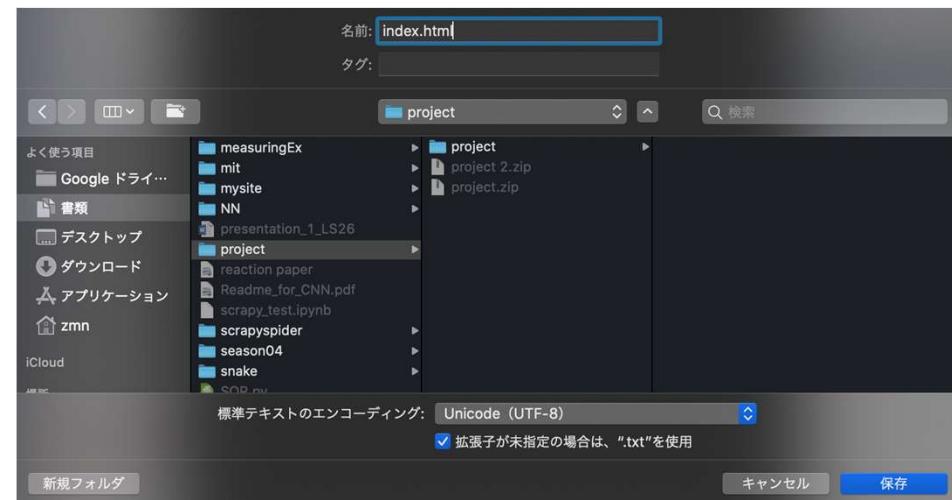
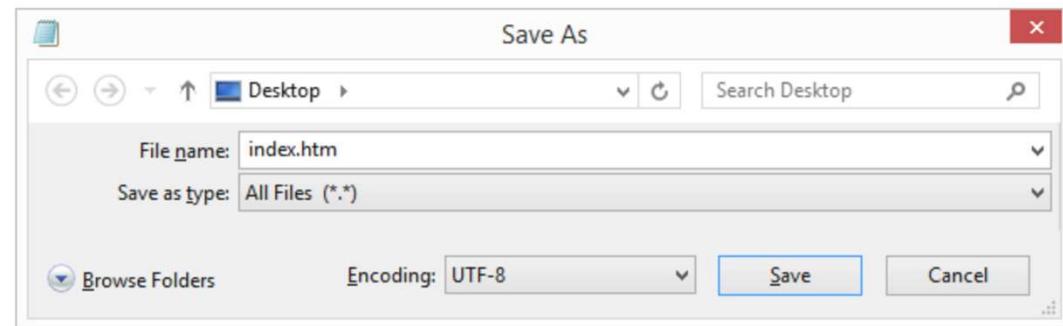
</body>
</html>
```

就像这样：



步骤3：保存HTML文件

- 将文件保存在您的计算机上。在记事本（textedit）菜单中选择文件 > 另存为。
- 将文件命名为“index.html”，并将编码设置为 UTF-8（这是 HTML 文件的首选编码）。



你可以使用.htm或.html作为文件扩展名，没有区别。

步骤4：在浏览器中查看HTML页面

- 在你喜欢的浏览器中打开保存的HTML文件（双击该文件，或右键单击-并选择“打开方式”）。
结果看起来像这样：



Q & A

You Have Questions
We Have Answers



1 Web开发概述
ウェブ開発概要

2 HTML概述
HTML概要

3 创建HTML文件
HTMLファイルの作成

4 HTML5基本语法(重点)
HTML5の基本文法

HTML文档[HTML ドキュメント] (Document)

- 所有HTML文档都必须以文档类型声明开头：<!DOCTYPE html>
- HTML文档本身以<html>结尾</html>
- HTML文档的可见部分在<body>和之间</body>

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

<h1>My First Heading</h1>
<p>My first paragraph.</p>

</body>
</html>
```

HTML元素 [HTML要素] (Element)

- HTML元素由**开始标签**[タグ], **一些内容**和**结束标签**[タグ]定义:
- HTML都有哪些元素: <https://www.w3schools.com/tags/default.asp>

<标签名>内容在这里... < /标签名>

HTML 元素是从开始标记到结束标记的所有内容:

< h1 >我的第一个标题 < / h1 >

< p >我的第一段。< / p >

注意: 某些HTML元素没有内容 (例如
元素)。这些元素称为空元素[空要素]。空元素[空要素]没有结束标签[タグ]!

嵌套[ネスト構造]的HTML元素

- HTML元素可以嵌套（这意味着元素可以包含其他元素）。
- 所有HTML文档都由嵌套的HTML元素组成。
- 下面的例子包含四个HTML元素（`<html>`, `<body>`, `<h1>` 和 `<p>`）：

```
<!DOCTYPE html>
<html>
  <body>
    <h1>My First Heading</h1>
    <p>My first paragraph.</p>
  </body>
</html>
```

HTML标签 Tips

- **不跳过结束标签**，即使你忘记了结束标签，某些HTML元素也将正确显示。但是，请不要依赖于此，如果忘记结束标签，可能会发生意外的结果和错误！
- 没有内容的HTML元素称为空元素[\[空要素\]](#)。比如
标签定义换行，并且是一个没有结束标签的空元素[\[空要素\]](#)。
- HTML标签**不区分大小写**：<P>与相同<p>。HTML标准不需要小写标签，但是**建议在HTML中使用小写**，对于更严格的文档类型（例如XHTML）则要求小写。
- 所有标签的基本用法表：<https://www.w3schools.com/tags/>

HTML属性 (Attribute)

- 所有HTML元素都可以具有属性，他提供HTML元素的其他信息
- 始终在开始标签<>中指定
- 通常以名称跟值成对的形式出现，例如： name = “ value”
- 格式：
`<标签名 属性=“属性值”> 内容 </标签名>`
- 例：
` 这是一个a标签， 跳转到本页面`

HTML属性范例

- **title属性**定义有关元素的补充信息。当你将鼠标悬停在元素上时，title属性的值将显示为提示：

```
<p title="I'm a tooltip">This is a paragraph.</p>  
<p title='John "ShotGun" Nelson'>
```

尝试代码： attribute.html

注意：HTML标准不需要在属性值两边加上引号。但是，W3C建议使用引号。

在HTML中，属性值周围的双引号是最常见的，但是也可以使用单引号。
在某些情况下，当属性值本身包含双引号时，有必要使用单引号：（反之亦然）

HTML属性范例

- **<a>标签的href属性:** [タグ] 定义超链接[ハイパーリンク]。 href属性指定超链接要转到的页面的URL：

```
<a href="https://www.w3schools.com">Visit W3Schools</a>
```

- **标签的src属性:** 指定要显示的图像的路径[パス]
- **width和height属性**指定图像的宽度和高度（以像素为单位）：

```

```

- **style属性**用于向元素添加样式，例如颜色，字体，大小等：

```
<p style="color:red;">This is a red paragraph.</p>
```

HTML 注释

- HTML注释不会显示在浏览器中，但是它们可以用来解释你的HTML源代码。

```
<!-- Write your comments here -->
```

```
<!-- This is a comment -->
```

```
<p>This is a paragraph.</p>
```

```
<!-- Remember to add more information here -->
```

HTML标签一览

- 共75个，黄色为**核心标签**（7个），红色为**重要标签**（32个）

标签	说明	标签	说明	标签	说明
<!--...-->	定义注释		定义强调文本	<p>	定义段落
<!DOCTYPE>	定义文档类型	<fieldset>	定义围绕表单中元素的边框	<param>	定义对象的参数
<a>	定义锚、连接	<form>	定义供用户输入的 HTML 表单	<pre>	定义预格式文本
<abbr>	定义缩写	<h1> to <h6>	定义HTML标题	<q>	定义短的引用
<address>	定义文档作者或拥有者的联系信息	<head>	定义关于文档的信息	<samp>	定义计算机代码样本
<area>	定义图像映射内部的区域	<hr>	定义水平线	<script>	定义客户端脚本
	定义粗体字	<html>	定义 HTML 文档	<select>	定义选择列表（下拉列表）
<base>	定义页面中所有链接的默认地址或默认目标	<i>	定义斜体字	<small>	定义小号文本
<bdo>	定义文字方向	<iframe>	定义内联框架		定义文档中的节
<big>	定义大号文本		定义图像		定义强调文本
<blockquote>	定义长的引用	<input>	定义输入控件	<style>	定义文档的样式信息
<body>	定义文档的主体	<ins>	定义被插入文本	<sub>	定义下标文本
 	定义简单的折行	<kbd>	定义键盘文本	<sup>	定义上标文本
<button>	定义按钮	<label>	定义input元素的标注	<table>	定义表格
<caption>	定义表格标题	<legend>	定义fieldset元素的标题	<tbody>	定义表格中的主体内容
<cite>	定义引用		定义列表的项目	<td>	定义表格中的单元
<code>	定义计算机代码文本	<link>	定义文档与外部资源的关系	<textarea>	定义多行的文本输入控件
<col>	定义表格中一个或多个列的属性值	<map>	定义图像映射	<tfoot>	定义表格中的表注内容（脚注）
<colgroup>	定义表格中供格式化的列组	<menu>	定义命令的列表或菜单	<th>	定义表格中的表头单元格
<dd>	定义定义列表中项目的描述	<menuitem>	定义用户可以从弹出菜单调用的命令/菜单项目	<thead>	定义表格中的表头内容
	定义被删除文本	<meta>	定义关于 HTML 文档的元信息	<title>	定义文档的标题
<div>	定义文档中的节	<object>	定义内嵌对象	<tr>	定义表格中的行
<dfn>	定义项目		定义有序列表	<tt>	定义打字机文本
<dl>	定义列表	<optgroup>	定义选择列表中相关选项的组合		定义无序列表
<dt>	定义列表中的项目	<option>	定义选择列表中的选项	<var>	定义文本的变量部分

HTML <h1> ~ <h6>标题

- HTML标题是使用<h1>to <h6>标签定义的。
- <h1>定义最大的标题。 <h6>定义最小的标题。

```
<h1>Heading 1</h1>
<h2>Heading 2</h2>
<h3>Heading 3</h3>
<h4>Heading 4</h4>
<h5>Heading 5</h5>
<h6>Heading 6</h6>
```

Heading 1

Heading 2

Heading 3

Heading 4

Heading 5

Heading 6

HTML 文本格式

- HTML包含几个用于定义具有特殊含义的文本的元素。

`` - 粗体文字

`` - 重要文字

`<i>` - 斜体文字

`` - 强调文字

`<mark>` - 标记文字

`<small>` - 较小的文字

`` - 删除的文字

`<ins>` - 插入文字

`<sub>` - 下标文字

`<sup>` - 上标文字

重点掌握

HTML <head>标签

- <head>标签是metadata[メタ情報]（元数据：记述数据的数据）的容器，被放置在<html>标签[タグ]和<body>标签之间。
- HTML metadata是有关HTML文档的数据。不会在浏览器上显示出来。
- metadata通常定义文档标题，字符集，样式（如CSS），脚本（如JS）和其他元信息[メタ情報]。

Metadata

Tag	Description
<head>	Defines information about the document
<title>	Defines the title of a document
<base>	Defines a default address or a default target for all links on a page
<link>	Defines the relationship between a document and an external resource
<meta>	Defines metadata about an HTML document
<script>	Defines a client-side script
<style>	Defines style information for a document

HTML <meta>标签

- <meta>标签用于指定使用哪个字符集，页面描述，关键字[キーワード]，作者和其他元数据。
- 浏览器（如何显示内容），搜索引擎（关键字）和其他Web服务都使用metadata。
- 定义使用的字符集：

```
<meta charset="UTF-8">
```

- 定义您的网页描述：

```
<meta name="description" content="Free Web tutorials">
```

- 定义搜索引擎的关键字：

```
<meta name="keywords" content="HTML, CSS, XML, JavaScript">
```

HTML <style>标签

- <style>标签用于定义一个HTML页面样式信息：

```
<!DOCTYPE html>
<html>
<head>
  <title>Page Title</title>
  <style>
    body {background-color: powderblue;}
    h1 {color: red;}
    p {color: blue;}
  </style>
</head>
<body>

  <h1>This is a Heading</h1>
  <p>This is a paragraph.</p>

</body>
</html>
```

This is a Heading

This is a paragraph.

HTML <script>标签

- <script>标签用于定义客户端[クライアント]JavaScript。

```
<script>
function myFunction() {
    document.getElementById("demo").innerHTML = "Hello JavaScript!";
}
</script>
```

- 第三节课会深入学习JS！

HTML <link>标签

- <link>标签用于链接到外部样式表[スタイルシート](如外部css文件):

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

- 下节课会深入学习CSS!

<div>标签

- <div>标签通常用作其他HTML元素的容器。
- <div>标签没有必需的属性，但是style, class和id经常被用到。
- 与CSS的一起使用时，<div>标签可用于设置内容块的样式：

```
<div style="background-color:black;color:white;padding:20px;">
  <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>
  <p>Standing on the River Thames, London has been a major settlement
for two millennia, its history going back to its founding by the Romans,
who named it Londinium.</p>
</div>
```

可以在<div> 标签内放置任何内容！

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.

Standing on the River Thames, London has been a major settlement for two millennia, its history going back to its founding by the Romans, who named it Londinium.

标签

- 标签通常用作某些文本的容器。
- 标签没有必需的属性，但是style, class和id经常被用到。
- 与CSS的style一起使用时，标签可用于设置文本的样式：

```
<h1>My <span style="color:red">Important</span> Heading</h1>
```

My Important Heading

HTML <a>超链接[ハイパーアリンク]

- 超链接是使用<a>标签[タグ]定义的：

```
<a href="url">link text</a>
```

- 例：

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

<h2>HTML Links</h2>
<p><a href="https://www.w3schools.com/html/">Visit our HTML
tutorial</a></p>

</body>
</html>
```

HTML Links

[Visit our HTML tutorial](https://www.w3schools.com/html/)

HTML <a>超链接[ハイパーリンク]—本地链接[ローカルリンク]

- 前面的示例使用了绝对URL（完整的网址）。
- 使用相对URL（不带https://www....）指定本地链接（指向相同网站的链接）。

HTML <a>超链接[ハイパーアリンク]—target属性

- target属性指定在何处打开链接的文档。
- target属性可以具有以下值之一：
 - _blank :在新窗口或标签[タグ]中打开链接的文档
 - _self :在与单击相同的窗口/选项卡中打开链接的文档（默认设置）
 - _parent :在父框架中打开链接的文档
 - _top :在整个窗口中打开链接的文档

尝试代码：target.html

HTML <a>超链接[ハイパーリンク]—图片作为链接

把img标签[タグ]嵌套在a标签里：

```
<a href="default.asp">  
    
</a>
```

尝试代码：img_link.html

HTML 图片 [イメージ]

- 在HTML中，图像是用标签[タグ]定义的空标签[空要素]。
- 该src属性指定图像的URL（网址）：

```

```

- alt属性：如果用户由于某种原因无法查看图像（由于连接速度慢，src属性错误或用户使用屏幕阅读器），则该属性会为图像提供替代文本：

```

```

尝试代码：[image.html](#)

HTML 图片[イメージ]—宽度和高度

- 可以使用style属性指定图像的宽度和高度。

```

```

- 也可以使用width和height属性：

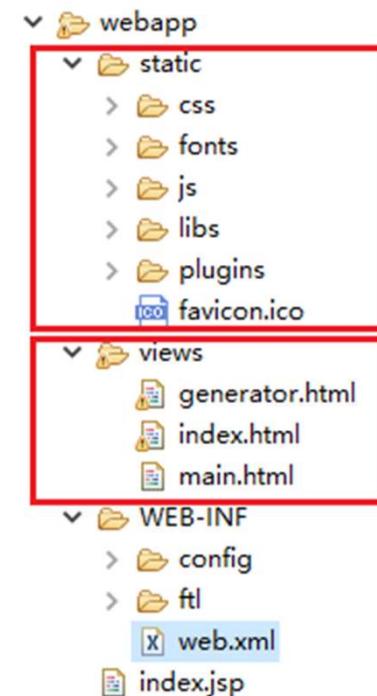
```

```

- 但是建议使用第一种方式

HTML文件路径[パス]

- 文件路径[パス]描述了网站文件夹结构中文件的位置。
- 使用文件路径链接到的外部文件例如：
 - Web pages
 - Images
 - CSSs
 - JavaScripts



HTML文件路径[パス]

- 绝对文件路径[絶対パス]是文件的完整URL:

```

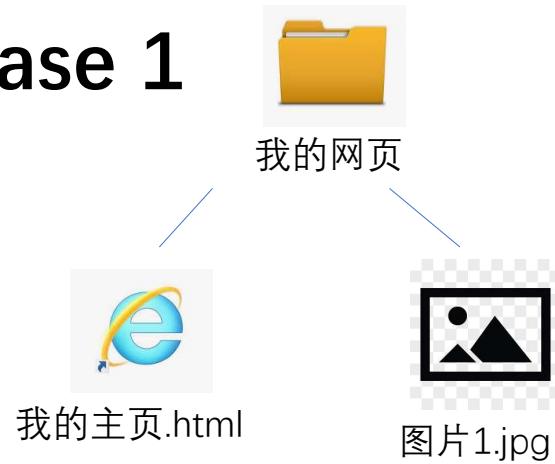
```

- 相对文件路径[相対パス]指向相对于当前页面的文件:

路径	描述
	picture.jpg与当前页面位于同一文件夹中
	picture.jpg位于当前文件夹的images文件夹中
	picture.jpg位于当前站点根目录的images文件夹中
	picture.jpg位于当前文件夹上一层的文件夹中

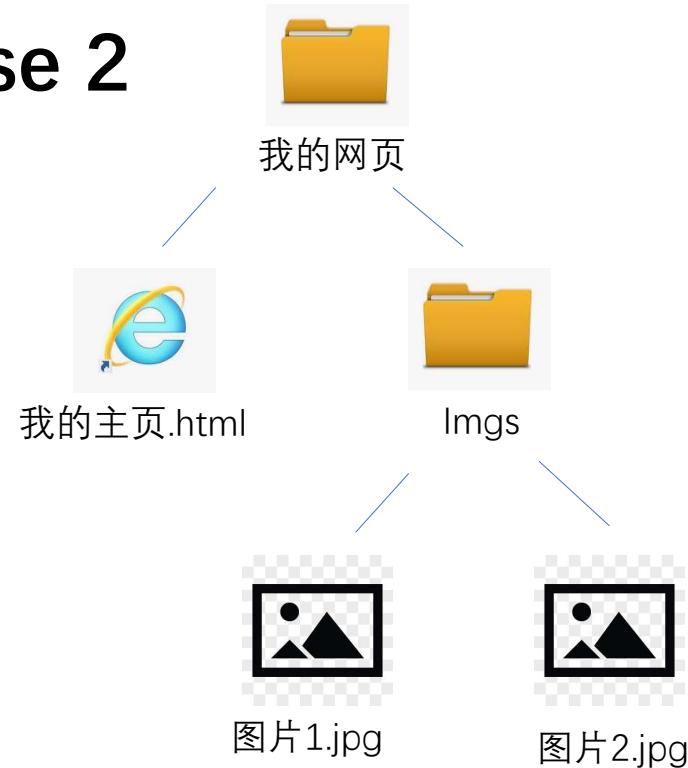
相对路径[相対パス]

Case 1



HTML文件里引入图片1：
`src="图片1.jpg"`

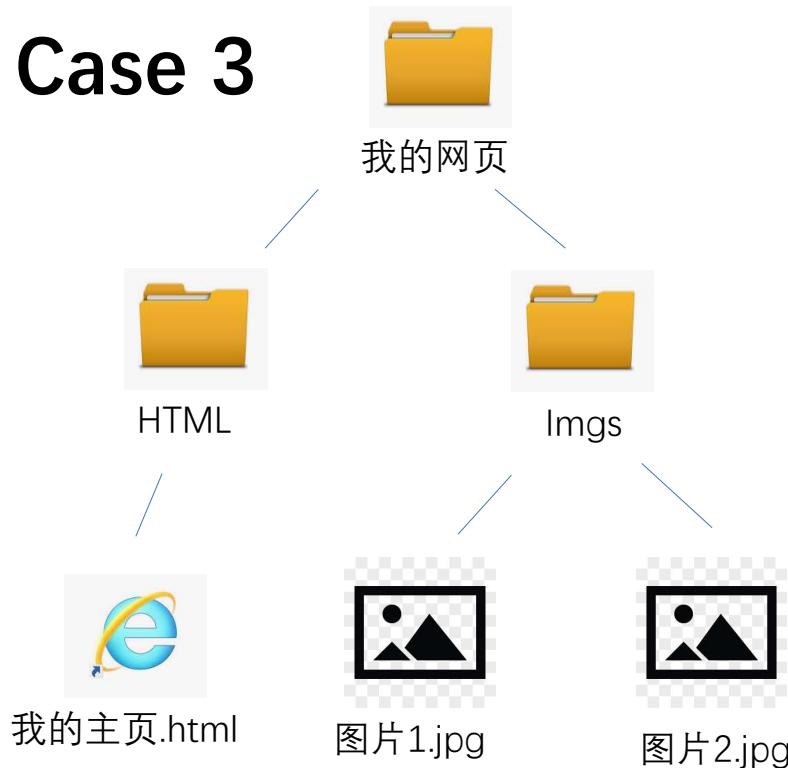
Case 2



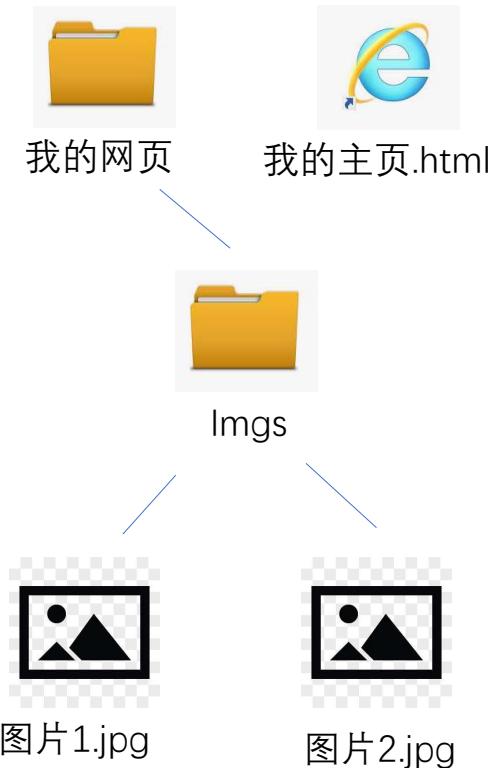
HTML文件里引入图片1：
`src="imgs/图片1.jpg"`

相对路径[相対パス]

Case 3



Case 4



HTML文件里引入图片1：
`src="..../imgs/图片1.jpg"`

HTML文件里引入图片1：
`src= ?`

HTML ,,列表[リスト] (Lists)

- 无序列表[順序なしリスト] (unordered list) : 无序列表以标签[タグ]开头。每个列表项均以标签[タグ]开头 :

```
<ul>
  <li>Coffee</li>
  <li>Tea</li>
  <li>Milk</li>
</ul>
```

- Coffee
- Tea
- Milk

- 有序列表[順序ありリスト] (ordered list): 有序列表以标签开头。每个列表项均以标签开头。 :

```
<ol>
  <li>Coffee</li>
  <li>Tea</li>
  <li>Milk</li>
</ol>
```

1. Coffee
2. Tea
3. Milk

HTML <table>表格[テーブル]

- HTML表是用<table>标签[タグ]定义的。
- 每个表行(table row)都用<tr>标签定义。表标题(table head)是用<th>标签定义的。表数据/单元格(table data)用<td>标签定义。

```
<table border="1">
  <tr>
    <th>Month</th>
    <th>Savings</th>
  </tr>
  <tr>
    <td>January</td>
    <td>$100</td>
  </tr>
</table>
```



Month	Savings
January	\$100

尝试代码：table.html

- 注：<td>是表的数据容器。
它们可以包含各种HTML标签：文字，图片，列表，其他表格等

HTML <table>表格[テーブル]—添加边框[ボーダーライン]

- 如果你没有为表格指定边框，它就不显示边框。
- 使用CSS border属性设置边框：

```
table, th, td {  
    border: 1px solid black;  
}
```

- 塌陷的边框：

```
table, th, td {  
    border: 1px solid black;  
    border-collapse: collapse;  
}
```

尝试代码：table.html

HTML <table>表格[テーブル]

- 添加单元格填充：单元格填充指定单元格内容与其边界之间的空间

◦

```
th, td {  
    padding: 15px;  
}
```

- 对齐：要使表格标题左对齐，请使用css text-align属性：(默认是center)

```
th {  
    text-align: left;  
}
```

尝试代码：table.html

HTML <table>表格[テーブル]—跨许多列的单元格

- 要使一个单元格跨越一列以上，请使用colspan属性：

```
<table style="width:100%">
  <tr>
    <th>Name</th>
    <th colspan="2">Telephone</th>
  </tr>
  <tr>
    <td>Bill Gates</td>
    <td>55577854</td>
    <td>55577855</td>
  </tr>
</table>
```

Name	Telephone	
Bill Gates	55577854	55577855

HTML <table>表格[テーブル]—跨许多行的单元格

- 要使一个单元格跨度超过一行，请使用rowspan属性：

```
<table style="width:100%">
  <tr>
    <th>Name:</th>
    <td>Bill Gates</td>
  </tr>
  <tr>
    <th rowspan="2">Telephone:</th>
    <td>55577854</td>
  </tr>
  <tr>
    <td>55577855</td>
  </tr>
</table>
```

Name:	Bill Gates
Telephone:	55577854
	55577855

HTML <table>表格[テーブル]—添加标题

- 要给表格添加标题，请使用<caption>标签[タグ]:

```
<table style="width:100%">
  <caption>Monthly savings</caption>
  <tr>
    <th>Month</th>
    <th>Savings</th>
  </tr>
  <tr>
    <td>January</td>
    <td>$100</td>
  </tr>
  <tr>
    <td>February</td>
    <td>$50</td>
  </tr>
</table>
```

Monthly savings	
Month	Savings
January	\$100
February	\$50

HTML <form>表单 [フォーム] (Forms)

- HTML表单用于收集用户输入。然后可以将用户输入发送到服务器 [サーバー] 进行处理。

First name:

Last name:

尝试代码: form.html

HTML <form>表单 [フォーム] (Forms)

- <form>标签定义了一种用于收集用户输入的表单：

```
<form>
  .
  form elements
  .
</form>
```

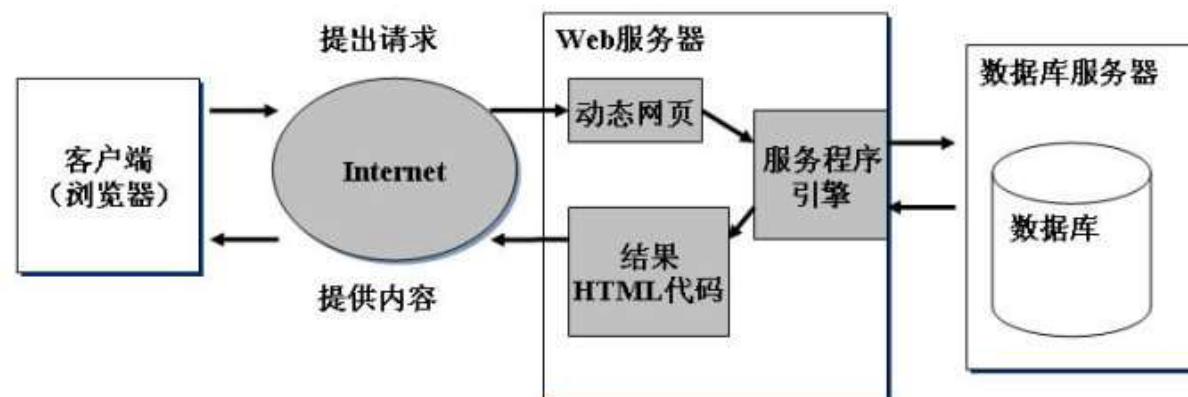
- <input>标签是最重要的表单标签。<input>根据type 属性，以几种方式显示。

类型	描述
<input type = " text">	定义单行文本输入字段
<input type = " radio">	定义一个单选按钮（用于选择多个选项之一）
<input type = " submit">	定义一个提交按钮（用于提交表单）

HTML <form>表单[フォーム]—action属性

- action属性定义提交表单时要执行的操作。
- 通常当用户单击“提交”按钮时，表单数据将发送到服务器[サーバー]上的页面。
- 下面表单数据被发送到服务器上名为“/action_page.php”的页面。此页面包含用于处理表单数据的服务器端脚本：

```
<form action="/action_page.php">
```



动态内容技术运行原理

HTML <form>表单[フォーム]—method属性

- method属性指定提交表单数据时要使用的HTTP方法（GET或POST）。

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

- 提交表单数据时，默认的HTTP方法是GET。表单数据将在页面的地址字段中可见。
- 如果表单数据包含敏感信息或个人信息，请使用POST。POST方法不会在页面地址字段中显示表单数据。

HTML <form>表单常见标签

Tag	Description
<u><form></u>	Defines an HTML form for user input
<u><input></u>	Defines an input control
<u><textarea></u>	Defines a multiline input control (text area)
<u><label></u>	Defines a label for an <input> element
<u><fieldset></u>	Groups related elements in a form
<u><legend></u>	Defines a caption for a <fieldset> element
<u><select></u>	Defines a drop-down list
<u><optgroup></u>	Defines a group of related options in a drop-down list
<u><option></u>	Defines an option in a drop-down list
<u><button></u>	Defines a clickable button
<u><datalist></u>	Specifies a list of pre-defined options for input controls
<u><output></u>	Defines the result of a calculation

HTML input 类型

重点掌握

- `<input type="button">`
- `<input type="checkbox">`
- `<input type="color">`
- `<input type="date">`
- `<input type="datetime-local">`
- `<input type="email">`
- `<input type="file">`
- `<input type="hidden">`
- `<input type="image">`
- `<input type="month">`
- `<input type="number">`
- `<input type="password">`
- `<input type="radio">`
- `<input type="range">`
- `<input type="reset">`
- `<input type="search">`
- `<input type="submit">`
- `<input type="tel">`
- `<input type="text">`
- `<input type="time">`
- `<input type="url">`
- `<input type="week">`

- 右侧是可以在HTML中使用的输入类型：

这是一些常见输入限制的列表：

Attribute	Description
checked	Specifies that an input field should be pre-selected when the page loads (for type="checkbox" or type="radio")
disabled	Specifies that an input field should be disabled
max	Specifies the maximum value for an input field
maxlength	Specifies the maximum number of character for an input field
min	Specifies the minimum value for an input field
pattern	Specifies a regular expression to check the input value against
readonly	Specifies that an input field is read only (cannot be changed)
required	Specifies that an input field is required (must be filled out)
size	Specifies the width (in characters) of an input field
step	Specifies the legal number intervals for an input field
value	Specifies the default value for an input field

HTML Layout标签 (布局)

- HTML提供了几种语义标签，它们定义了网页的不同部分：



- `<header>` - 定义文档或节的标题
- `<nav>` - 定义了一组导航链接
- `<section>` - 在文档中定义一个部分
- `<article>` - 定义独立，独立的内容
- `<aside>` - 定义除内容之外的内容（如边栏）
- `<footer>` - 为文档或节定义页脚
- `<details>` - 定义用户可以按需打开和关闭的其他详细信息
- `<summary>` - 定义 `<details>` 元素的标题

尝试代码：layout.html

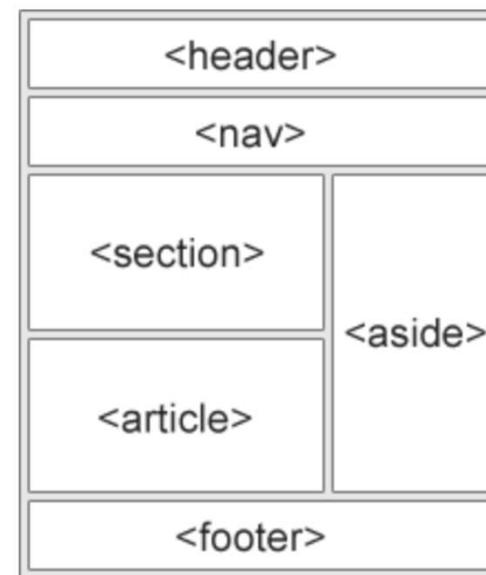
HTML 语义元素 [セマンティック要素]

- 语义元素=具有含义的元素。
- 语义元素对于浏览器和开发人员都清楚地描述了其含义。
- 非语义元素的示例：<div>和-不知道其内容。
- 语义元素的示例：<form>, <table>和<article>-明确定义其内容。

在HTML中，有一些语义元素可用于定义网页的不同部分：

查看更多语义元素：

https://www.w3schools.com/html/html5_semantic_elements.asp



HTML <section>元素

- <section>元素定义了一个文档中的一个部分。
- 部分是内容的主题分组，通常带有标题。
- 通常可以将主页分为介绍，内容和联系信息等部分。

```
<section>
  <h1>WWF</h1>
  <p>The World Wide Fund for Nature (WWF) is....</p>
</section>
```

HTML <article>元素

- <article>元素指定独立的，自包含的内容。
- 文章应该是有意义的，并且应该可以独立于网站的其余部分来阅读
 -
- <article>使用示例：
 - 论坛帖子
 - 博客文章
 - 报纸文章

```
<article>
  <h1>What Does WWF Do?</h1>
  <p>WWF's mission is to stop the degradation of our planet's natural
environment,
  and build a future in which humans live in harmony with nature.</p>
</article>
```

HTML <header>元素

- <header>元素指定一个文件或部分（section）的标题。
- <header>元素通常被用作介绍性内容的容器。
- 一个<header>中可以包含多个元素。

```
<article>
  <header>
    <h1>What Does WWF Do?</h1>
    <p>WWF's mission:</p>
  </header>
  <p>WWF's mission is to stop the degradation of our planet's natural
environment,
and build a future in which humans live in harmony with nature.</p>
</article>
```

What Does WWF Do?

WWF's mission:

WWF's mission is to stop the degradation of our planet's natural environment, and build a future in which humans live in harmony with nature.

HTML <nav>元素

- <nav>元素定义了一组导航链接。
- 请注意，并非文档的所有链接都应在<nav>元素内。<nav>元素仅用于主要的导航链接块。

```
<nav>
  <a href="/html/">HTML</a> |
  <a href="/css/">CSS</a> |
  <a href="/js/">JavaScript</a> |
  <a href="/jquery/">jQuery</a>
</nav>
```

[HTML](#) | [CSS](#) | [JavaScript](#) | [jQuery](#).

HTML <aside>元素

- <aside>元素定义了边栏放置的内容。 (sidebar)
- <aside>的内容应与周围的内容相关联。

```
<p>My family and I visited The Epcot center this summer.</p>
```

```
<aside>
  <h4>Epcot Center</h4>
  <p>The Epcot Center is a theme park in Disney World, Florida.</p>
</aside>
```

My family and I visited The Epcot center this summer.

Epcot Center

The Epcot Center is a theme park in Disney World, Florida.

HTML <footer>元素

- <footer>元素指定为一个文件或部分（section）页脚。
- <footer>元素需要包含有关它的包含单元的信息。
- 页脚通常包含文档的作者，版权信息，使用条款的链接，联系信息等
 -
- 一个<footer>中可以包含多个元素。

```
<footer>
  <p>Posted by: Hege Refsnes</p>
  <p>Contact information: <a href="mailto:someone@example.com">
    someone@example.com</a>.</p>
</footer>
```

Posted by: Hege Refsnes

Contact information: someone@example.com.

HTML 块级元素 [ブロックレベル要素]

- 块级元素（块元素）总是从新行开始，并占据可用的全部宽度（尽可能向左和向右伸展）。



- HTML中的常见块级元素：
 - <div>, <p>, <h1>-<h6>, <hr>,
 - <form>, <table>, , , ,

HTML 内联元素[インライン要素]

- 内联元素（行元素）不从新行开始，仅占用所需的宽度。



- HTML中的常见内联元素：
 - <a>, ,

 - <button>, , <input>, <label>, <select>
- 如果有需要，行元素和块元素直接可以相互转化

HTML 一些建议

- 不要忘记关闭所有非空的HTML元素。
- 建议使用小写的属性名称。
- 建议属性值带引号。
- 为了提高可读性，请添加空白行以分隔大型或逻辑代码块。
- 为了便于阅读，请添加两个空格作为缩进[インデント]。不要使用Tab键。
- 设置视口。
- .htm和.html文件扩展名之间没有区别！

如何查阅HTML相关信息

- 可以到[W3School中文网站](#)查阅
- [这个网站](#)有所有HTML标签的解说
- 也可以直接谷歌“HTML +‘你想实现的功能’”



HTML 颜色

- 在HTML中，可以通过使用颜色名称或颜色代码来指定颜色。
- HTML支持140种标准颜色名称。
- 设置背景颜色:(style属性的值的内容属于CSS的知识，下节课会具体讲到)

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

<h1 style="background-color:DodgerBlue;">Hello World</h1>

<p style="background-color:Tomato;">
Lorem ipsum dolor sit amet, consectetuer adipiscing elit, sed diam
nonummy nibh euismod tincidunt ut laoreet dolore magna aliquam erat
volutpat. Ut wisi enim ad minim veniam, quis nostrud exerci
tation ullamcorper suscipit lobortis nisl ut aliquip ex ea commodo
consequat.

</p>

</body>
</html>
```

Hello World

Lorem ipsum dolor sit amet, consectetuer adipiscing elit, sed diam nonummy nibh euismod tincidunt ut laoreet dolore magna aliquam erat volutpat. Ut wisi enim ad minim veniam, quis nostrud exercitatio ullamcorper suscipit lobortis nisl ut aliquip ex ea commodo consequat.

HTML 颜色

- 设置文字颜色：

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

<h3 style="color:Tomato;">Hello World</h3>

<p style="color:DodgerBlue;">Lorem ipsum dolor sit amet, consectetuer
adipiscing elit, sed diam nonummy nibh euismod tincidunt ut laoreet
dolore magna aliquam erat volutpat.</p>

<p style="color:MediumSeaGreen;">Ut wisi enim ad minim veniam, quis
nostrud exerci tation ullamcorper suscipit lobortis nisl ut aliquip ex ea
commodo consequat.</p>

</body>
</html>
```

Hello World

Lorem ipsum dolor sit amet, consectetuer adipiscing elit, sed diam nonummy nibh euismod tincidunt ut laoreet dolore magna aliquam erat volutpat.

Ut wisi enim ad minim veniam, quis nostrud exerci tation ullamcorper suscipit lobortis nisl ut aliquip ex ea commodo consequat.

HTML 颜色

- 设置边框[ボーダーライン]颜色: (border:2px solid Tomato表示番茄色的2像素单位粗的边框[ボーダーライン])

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

<h1 style="border: 2px solid Tomato;">Hello World</h1>
<h1 style="border: 2px solid DodgerBlue;">Hello World</h1>
<h1 style="border: 2px solid Violet;">Hello World</h1>

</body>
</html>
```

Hello World

Hello World

Hello World

Q & A

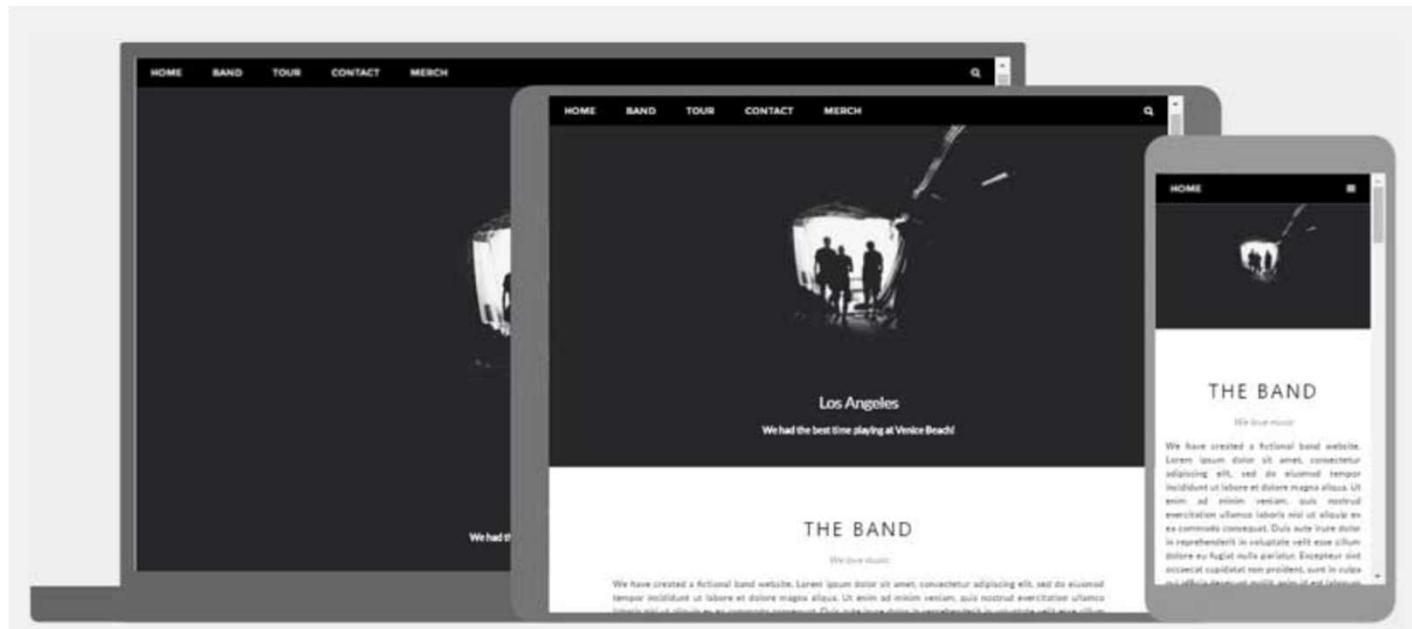
You Have Questions
We Have Answers

Appendix——HTML布局技术

- 有四种不同的技术可以创建多列布局。每种方式都有其优点和缺点：
 - CSS框架
 - CSS float属性
 - CSS Flexbox
 - CSS网格
- 下节课会深入讲解CSS！

Appendix—— HTML 响应式网页设计(Responsive Web Design)

- 响应式网页设计[レスポンシブWebデザイン]是指创建在所有设备上看起来都正常的网页
- 响应式网页设计会自动针对不同的屏幕尺寸和视口[ビューポート] (viewport) 进行自我调整。



Appendix—— HTML 响应式网页设计—设置视口(viewport)

- 想要创建响应式网站，请在你的所有网页上添加以下标签：

```
<meta name="viewport" content="width=device-width, initial-scale=1.0">
```

- 这样会设置页面的视口，向浏览器提供有关如何控制页面的尺寸和缩放比例的说明。



更多关于响应式网页设计的内容：

https://www.w3schools.com/html/html_responsive.asp

Appendix—— HTML 符号实体

- 许多数学，技术和货币符号在普通键盘上没有。
- 要将此类符号添加到HTML页面的话，可以使用HTML实体名称。
- 如果不存在实体名称，则可以使用实体编号，十进制或十六进制引用。
- 空格： &nbsp

```
<p>I will display &euro;</p>
<p>I will display &#8364;</p>
<p>I will display &#x20AC;</p>
```

I will display €

I will display €

I will display €

查看更多数学符号：

https://www.w3schools.com/charsets/ref_utf_math.asp

Appendix—— HTML Uniform Resource Locators (URL)

- URL即网址。
- URL可以由单词 (w3schools.com) 或Internet协议 (IP) 地址 (192.68.20.50) 组成。
- URL遵循以下语法规则：
scheme://prefix.domain:port/path/filename
- 说明：
 - scheme-定义Internet服务的类型 (最常见的是http或https)
 - prefix-定义域前缀 (http的默认值为www)
 - domain -定义Internet 域名 (如w3schools.com)
 - port-定义主机上的端口号 [ポート] (http的默认值为80)
 - path-定义服务器[サーバー]上的路径 [パス] (如果省略：站点的根目录)
 - filename -定义文档或资源的名称

THANK YOU