

4.1 HTML 基础

- 网页开发基本概念
- HTML 简介
- HTML 基本语法
- HTML 常见元素

目录

1

网页开发基本概念

2

HTML 简介

3

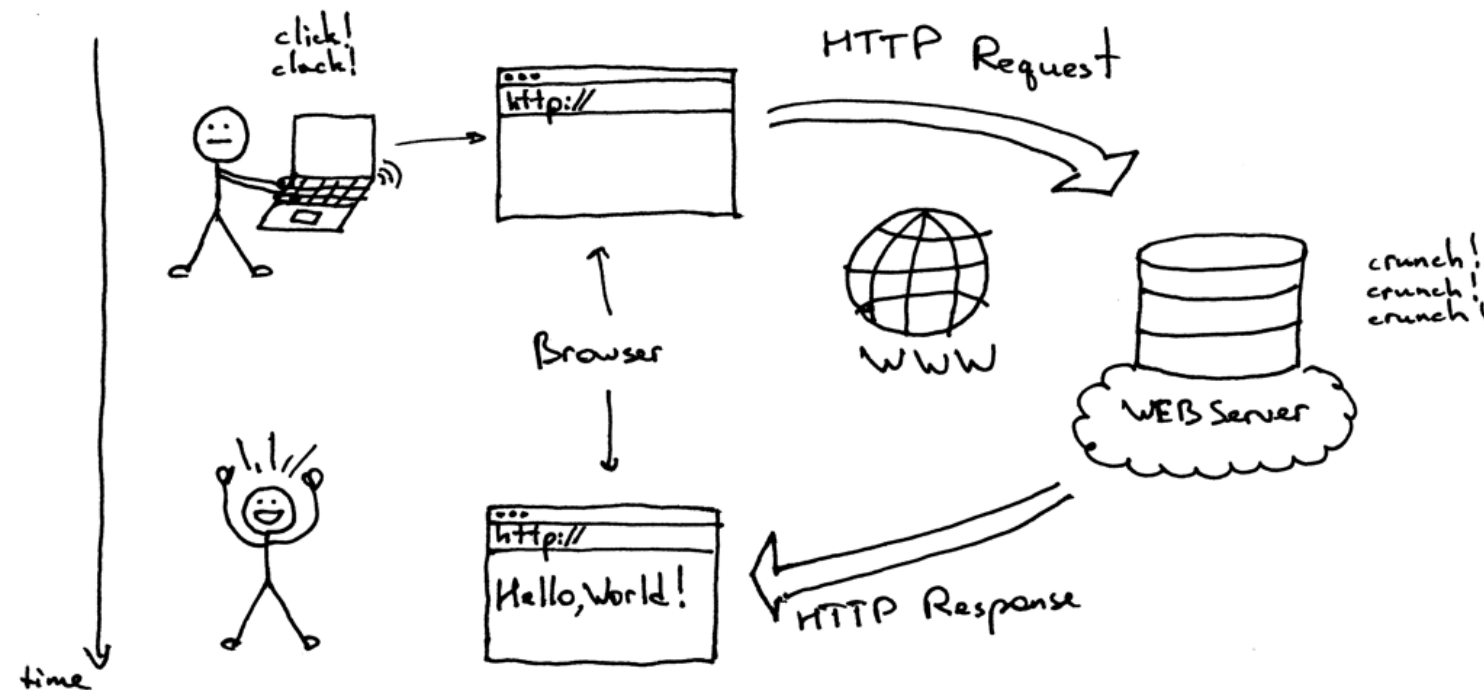
HTML 基本语法

4

HTML 常见元素

Web 服务

- **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]。
- Web 服务所提供信息的基本单位是**网页** [ウェブページ]，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]，使用户能够直接访问另外一个页面。



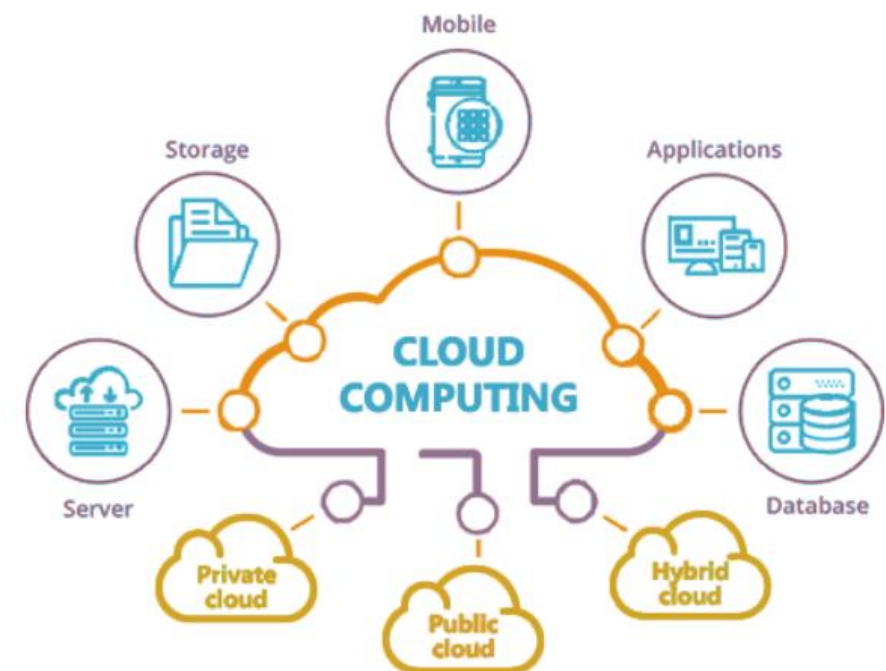
网页服务器

- **网页服务器**[Web サーバー]是一台负责提供网页的计算机，它通过超文本传输协议（HTTP）将网页数据传输给**客户端**[クライアント]。
- 网络服务器上通常会运行一个网络服务器程序。以下是 Netcraft 提供的截至 2021 年的网络服务器程序市场份额：

服务	供应商	市场占比
nginx	Nginx, Inc.	34.95%
Apache	Apache Software Foundation	24.63%
OpenResty	OpenResty Software Foundation	6.45%
Cloudflare	Cloudflare, Inc.	4.87%
IIS	Microsoft	4.00%
GWS	Google	4.00%

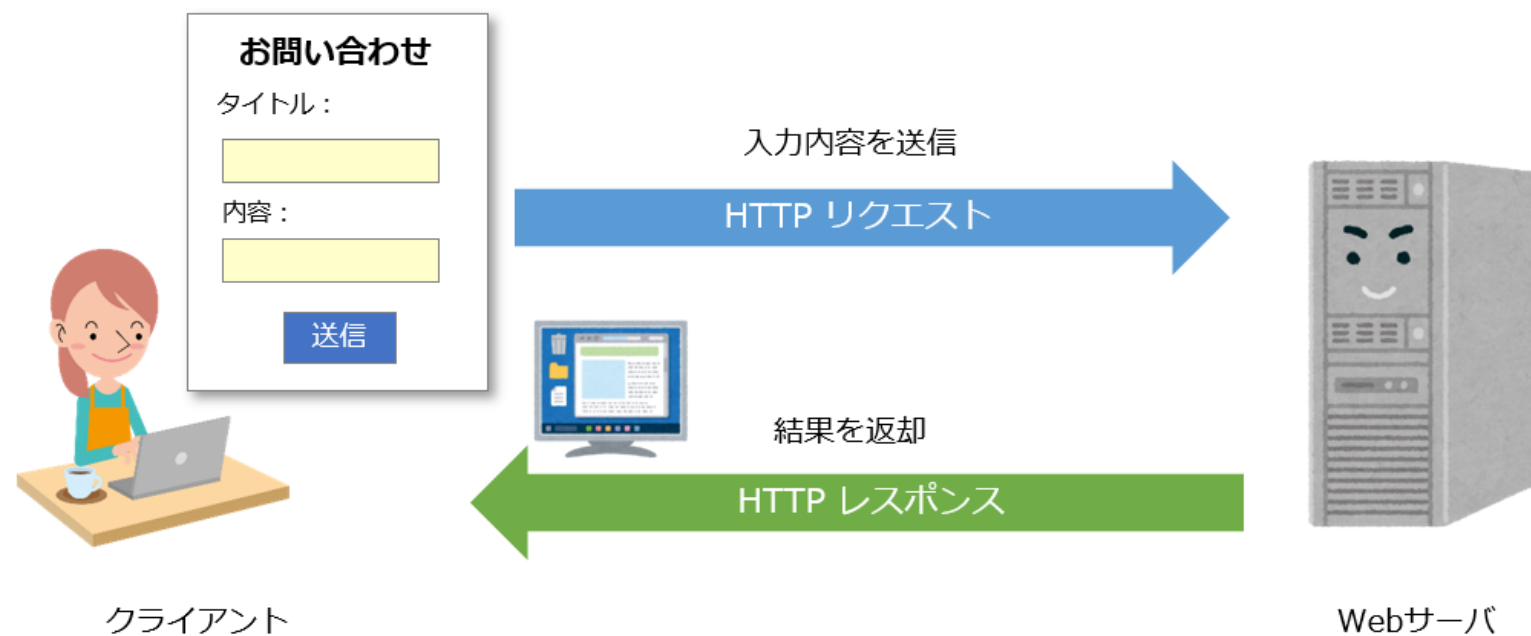
云计算服务

- **云计算** [Web サーバー] 是一种区别于传统客户端 - 服务端形式的新的互联网服务形式。
- 云计算通过共享的硬件和软件资源向用户提供服务，服务端的持有者不再需要了解基础服务的细节。这样做有几个好处：
 - 服务端部署的学习成本更低。
 - 服务端的调整更灵活快速。
 - 资源使用量的调整更灵活。
 - 全局部署更简单。
- 常见的云计算服务包括 Amazon Web Services (**AWS**)、Google Compute Platform、Microsoft Azure 等。



超文本传输协议

- **超文本传输协议** (HyperText Transfer Protocol, **HTTP**) 是一个客户端和服务端之间传输**请求**[リクエスト]和**响应**[レスポンス]的标准。
- 当客户端想要获得一个网页时，可以通过浏览器、爬虫等程序发送一个 **HTTP** 请求至服务器。服务器会接受该请求并通过 **HTTP** 应答的形式返回网页的数据。



网页浏览器

- **网络浏览器**[\[ウェブブラウザ\]](#)是一种可以读取网页并正确显示它们的软件。
- 网络浏览器也可以帮助我们向服务器发起 HTTP 请求从而获得想要访问的网页。
- 主要的网络浏览器包括 Google Chrome, Microsoft Edge, Mozilla Firefox, Safari 等。
- 常用浏览器市场占比:



截至 2021 年 10 月, [台式机浏览器的市场份额](#) [22]

Google Chrome	72.96%
Microsoft Edge	12.61%
Firefox 浏览器	5.54%
Safari	2.72%
Internet Explorer	2.45%

Q & A

Question and answer

目录

1

网页开发基本概念

2

HTML 简介

3

HTML 基本语法

4

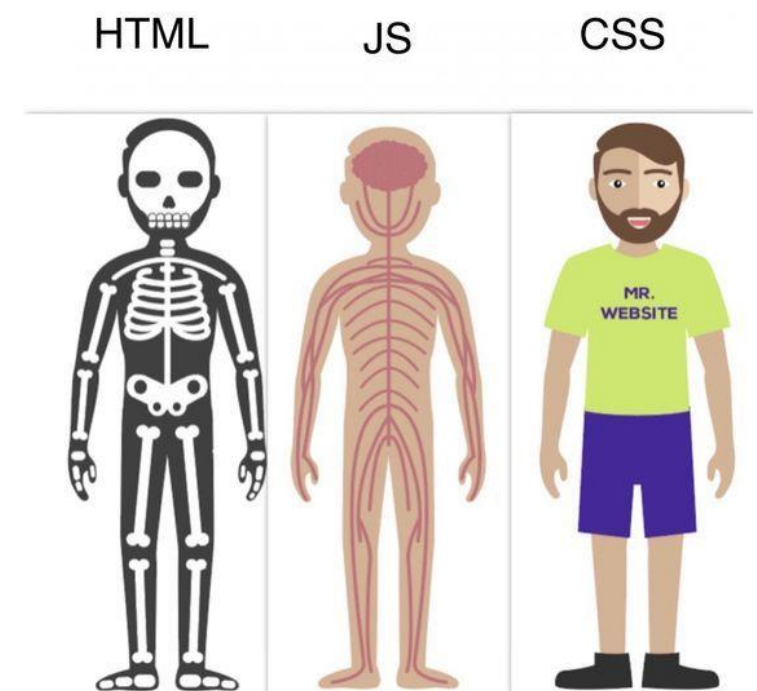
HTML 常见元素

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]。
- **CSS** 是描述 HTML 文档**样式**的样式表语言[スタイルシート言語]，描述了 HTML 中的元素具体应该如何显示。
- **JavaScript** 是编写网页**行为**的编程语言。

- 例如：

- HTML：设置文字、图片的内容等。
- CSS：设置文字的大小，颜色，字体等。
- JavaScript：设置点击按钮、的反应等。



静态网页和动态网页

- **静态网页**^[静的ページ]：每当用户向服务器发送同样的请求，客户端总会收到同样的网页。不管任何人任何时间获得的页面的内容都是不变的。
- **动态网页**^[動的ページ]：服务器根据客户的不同请求动态的生成网页内容。根据用户请求的种类和附加的信息，服务器会生成各种各样的网页。

	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 .shbm .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

HTML

- **HTML** 是一种超文本标签语言，描述了网页内容的基本结构。
- HTML 文档由一系列元素组成，每一个元素都描述了网页上显示的一个内容。
- 当浏览器打开 HTML 文档，它们将读取每个元素，并把它们转换成我们最终看到的网页上的内容。

```
1 <!DOCTYPE html>
2 <html>
3   <head>
4     <title>Hello</title>
5   </head>
6
7   <body>
8     <h1>This Is A Heading</h1>
9     <p>This is a paragraph.</p>
10    <button>This is a button</button>
11  </body>
12 </html>
```

This Is A Heading

This is a paragraph.

This is a button

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 HTML 5。

HTML 文件的编写

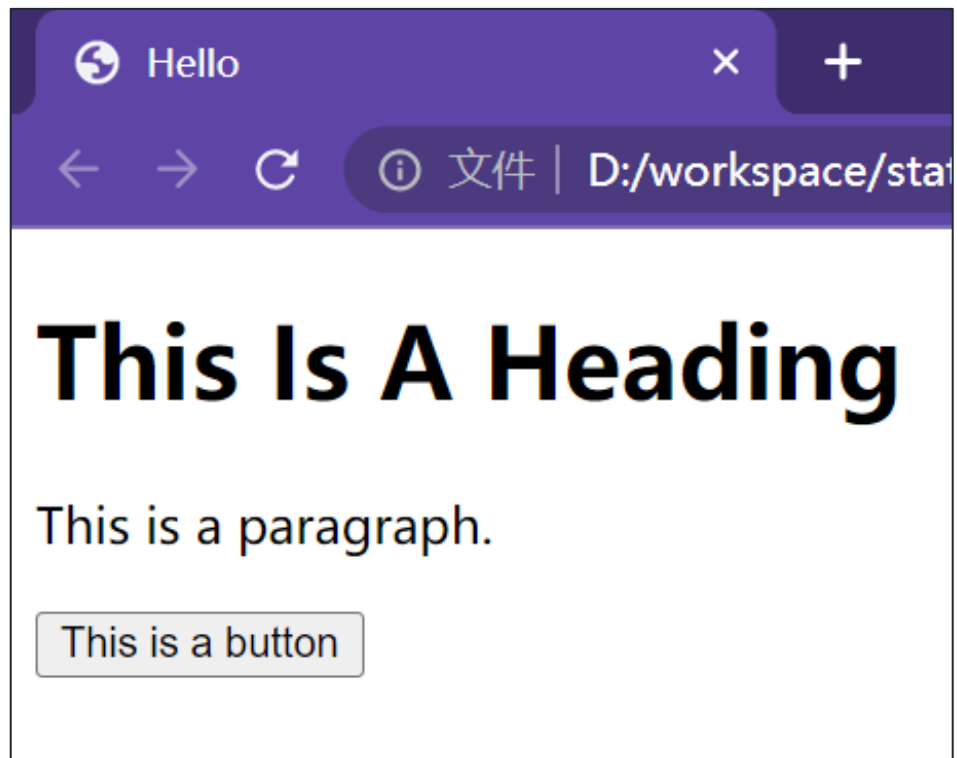
- HTML 文档和 Java 代码一样都可以用普通的文本编辑器书写。
- 打开你的文本编辑器，输入以下 HTML 代码（注意缩进 **2** 个空格）：

```
1 <!DOCTYPE html>
2 <html>
3 <head>
4   <title>Hello</title>
5 </head>
6
7 <body>
8   <h1>This Is A Heading</h1>
9   <p>This is a paragraph.</p>
10  <button>This is a button</button>
11 </body>
12 </html>
```

- 将该文件保存为 hello.html。

HTML 文件的运行

- 我们说过，浏览器可以用来显示网页。网页的本质就是 HTML 文档中的元素，我们只需要将写好的 HTML 文件放入到自己平常使用的浏览器中即可运行：



- 由于最新版 HTML 的某些功能在一些旧浏览器中可能无法实现，建议大家暂时在主流浏览器（如 Chrome）的最新版本上进行测试。

HTML 文件的开发环境

- 只用系统自带的文本编辑器开发仍然会有很多问题，比如无法高亮语法、Debug 困难等。
- 但总体来说静态网页开发不像其他语言一样需要很复杂的开发环境，各位根据自己的喜好选择一个文本编辑器即可。
- 常见的可以用于 HTML 开发的文本编辑器包括 **Notepad++**、**Sublime Text**、**Atom**、**Visual Studio Code** (VS Code) 等。
- 实际上，**Eclipse** 本身也可以直接用于书写 HTML 文件。

Q & A

Question and answer

目录

1

网页开发基本概念

2

HTML 简介

3

HTML 基本语法

4

HTML 常见元素

HTML 元素和标签

- **元素**[要素]是 HTML 最基本的组成部分。
- 每一个元素都通过 1 个**标签**[タグ]来定义。标签通常由 1 个**开始标签**“<>”、内容和**结束标签**“</>”构成：

```
<h1>This Is A Heading</h1>  
  
<p>This is a paragraph.</p>  
  
<button>This is a button</button>
```

- 也有一些元素没有内容和结束标签，只有 1 个开始标签：

```
<hr>
```


元素的嵌套

- 一个 HTML 元素中可以放入另一些 HTML 元素，只要将后者的标签写在前者的开始和结束标签之间即可：

```
<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 文档结构

声明该文件是 HTML 5 文档

```
1 <!DOCTYPE html>
```

html 元素里包含了网页的所有内容

```
2 <html>
```

```
3 <head>
```

head 元素包含页面的元信息

```
4 <title>Hello</title>
```

title 元素指定页面的标题

```
5 </head>
```

body 元素包含了网页上所有的可见内容

```
6 <body>
```

```
7 <h1>This Is A Heading</h1>
```

h1 元素定义了一个大的标题

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

p 元素定义了一个段落

```
9 <button>This is a button</button>
```

button 元素定义了一个按钮

```
10 </body>
```

```
11 </html>
```

属性

- 每一个 HTML 元素都可以具有一些属性 (Attribute) 。属性提供元素的一些具体信息，如名字、类型等。
- 属性始终在开始标签中指定：

```
<p 属性名="属性值"></p>
```

- 如果有多个属性，可以用空格隔开。
- 比如，在刚刚代码中的 <h1> 标签中加入 title 属性：

```
<h1 title="Hi!">This Is A Heading</h1>
```



This Is A Heading

Hi!

全局属性

- 一般来说，每种元素都会有一些它们特有的属性。但也有一些属性可以应用于任意元素，它们被称为全局属性[グローバル属性]。
- 常见的全局属性包括：
 - **id**：定义元素的 **id**；
 - **class**：定义元素的类；
 - **style**：定义元素的样式；
 - **lang**：定义元素的语言；
 - **title**：定义元素的标题。
- 我们会在讲 CSS 时详细介绍 id、class 和 style。

注释

- HTML 的**注释**通过一个特别的标签来实现：

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

- 该标签会被浏览器无视。由于标签内可以换行，多行注释也可以用同样方式实现。
- 注意 “<!--” 或 “-->” 各自所有字符必须连在一起，不能插入空格。

Q & A

Question and answer

目录

1

网页开发基本概念

2

HTML 简介

3

HTML 基本语法

4

HTML 常见元素

HTML 元素一览

- 所有 HTML 元素和它们所对应的标签都可以在这里查看：
<https://developer.mozilla.org/en-US/docs/Web/HTML/Element>
- 我们今天只介绍一些最常见的元素。

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

<head> 标签

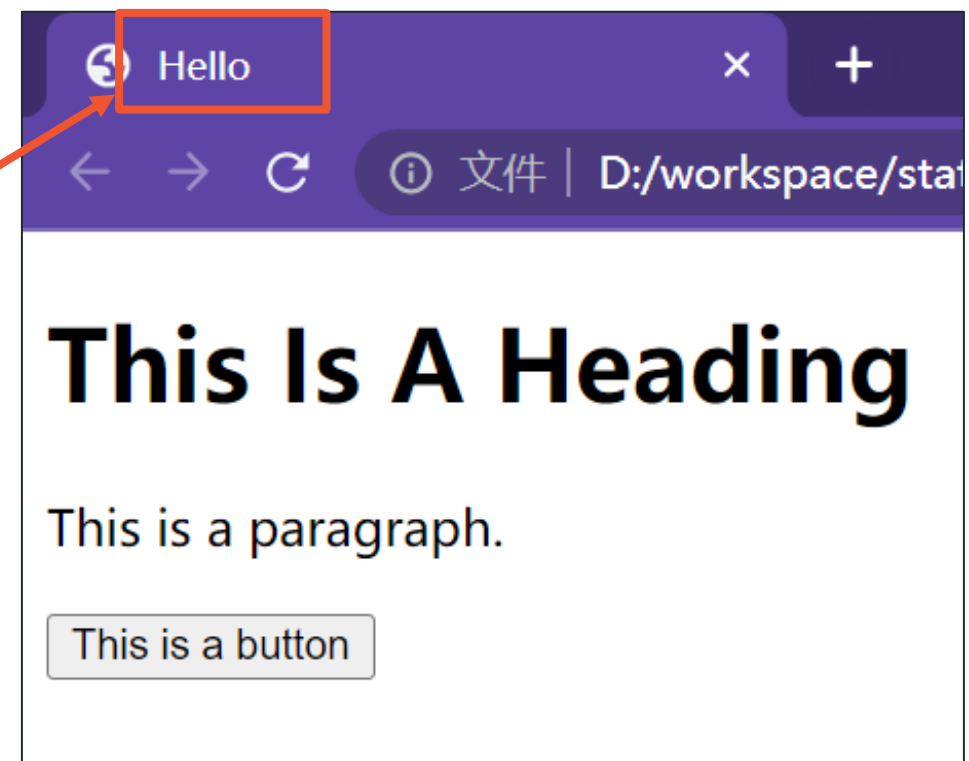
- **<head>** 标签描述存放**元数据**的元素。
- **元数据**[メタデータ]又被称作“数据的数据”，它记录一些和文档的内容无关而是和 **HTML 文档本身**有关的数据。比如文档**标题**，**字符集**，**样式 (CSS)**，**脚本 (JavaScript)** 等。

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

<title> 标签

- **<title>** 标签用于指定网页的**标题**。在使用浏览器打开网页时就可以看到该标题：

```
<!DOCTYPE html>
<html>
<head>
  <title>Hello</title>
</head>
```



<link>、<style> 和 <script> 标签

- **<link>** 标签用于链接到外部样式表，也就是外部的 CSS 文件。
- **<style>** 标签用于直接书写内部样式表，也就是直接在 HTML 文档里嵌入 CSS 代码。
- **<script>** 标签用于书写脚本或导入外部脚本，也就是 JavaScript 代码。
- 这些标签我们会在讲 CSS 和 JavaScript 时作详细介绍。

<meta> 标签

- 除了上述内容之外的元数据基本都通过 **<meta>** 标签定义。
- <meta> 标签最重要的作用之一是指定**字符集**:

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

- 除此之外，<meta> 标签还可用于定义作者、页面介绍、关键字、主题、**视口**（见附录）等数据。这些数据不会直接影响页面的内容，但是浏览器或其他使用该页面的程序（*爬虫*、搜索引擎等）可以利用这些信息提升用户体验。

标题

- **<h1> ~ <h6>** 标签定义标题。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

段落

- **<p>** 标签定义段落。段落内容的文字会被直接显示在网页上。
- 注意：标题和段落中的换行和多余的空白字符（比如连续的空格）会被省略！

```
1 <h1>Heading 1</h1>
2 <p>First paragraph.</p>
3 <p>
4   Second paragraph.
5   This is a longer paragraph.
6 </p>
```

Heading 1

First paragraph.

Second paragraph. This is a longer paragraph.

换行

- 若要在段落中添加换行，你可以使用 **
** 标签：

```
1 <h1>Heading 1</h1>
2 <p>First paragraph.</p>
3 <p>
4   Second paragraph.<br>This is a longer paragraph.
5 </p>
```

Heading 1

First paragraph.

Second paragraph.

This is a longer paragraph.

文本格式

- 使用以下标签括起文字可以使它们具有一些特殊的样式效果：

`` - 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

这是**粗体文字**。

这是*斜体文字*。

这是标记文字。

这是较小文字。

这是插入文字。

这是_{下标文字}。

这是^{上标文字}。

这是~~有多种特效的文字~~。

Try 
text-effect.html

符号实体

- **HTML 实体**[エンティティ]是一些以 “&” 开头, “;” 结尾的特殊字符串。
- 在想要显示某些特殊字符时, 我们可以使用符号名称或其 Unicode 编号书写 HTML 实体。
- 下表列出了一些重要的符号实体:

实体名	符号
 	(半角空格)
<	<
>	>
"	"
&	&

分割线

- **<hr>** 标签可以用来添加一条水平的分割线。

```
1 <h1>标题</h1>
2 <hr>
3 <p>段落 1</p>
4 <hr>
5 <p>段落 2</p>
```

标题

段落 1

段落 2

 标签

- **** 标签通常用作某些文本的**容器**。它本身不会对内容的文字产生任何影响。
- 但是当我们使用 **CSS** 设置文本样式时，**** 标签可以帮助我们指定需要改变样式的部分：

```
<h1>这是一个<span style="color:red">红色</span>标题</h1>
```

这是一个**红色**标题

<div> 标签

- **<div>** 标签通常用作其他 HTML 元素的**容器**。
- 和 `` 标签类似，`<div>` 本身不会影响其他元素的内容，但它可以将一些元素“打包”在一起，成为一个整体。
- `<div>` 标签主要有两个作用：
 1. 整理网页内元素的结构，加强代码可读性；
 2. 配合 CSS，统一设置某个 `<div>` 中所有元素的样式。



Q & A

Question and answer

超链接

- **<a>** 标签定义一个超链接。
- **超链接**[ハイパーリンク]可以链接向一个**本地**（和网页在同一台计算机上的）文件或者**互联网**上的文件。
- **<a>** 标签的语法如下：

```
<a href="链接">显示内容</a>
```
- 我们使用 **href 属性**定义超链接指向的文件地址（**URL**），在标签内容处定义超链接显示的文字。



Note

指向互联网上的文件要以
http:// 或 **https://** 开头。

超链接的 target 属性

- 超链接 **target** 属性指定在何处打开链接文件。
- 将 target 设为 “**_blank**” 可以使文件在新标签页中打开而不会关闭原页面：

```
<p>点击<a href="hello.html" target="_blank">这里</a>可以访问别的页面。</p>
```

图片

- **** 标签定义一张图片。
- **src** 属性指定图片文件的地址。
- **width** 和 **height** 属性指定图片的宽度和高度（单位为像素）。
- **alt** 属性指定一段文本，如果图片文件无法被加载，就会显示该文本。
- 也可以将图片放入一个 **<a>** 标签中，使其成为一个超链接。



列表

- HTML 中有两种列表：**无序列表**[箇条書き]和**有序列表**[番号付き箇条書き]。
 - **** 标签定义一个无序列表 (Unordered List) 。
 - **** 标签定义一个有序列表 (Ordered List) 。
 - **** 标签定义列表中的一项 (List Item) 。

```
<ul>
  <li>Apple</li>
  <li>Banana</li>
  <li>Cherry</li>
</ul>
```

- Apple
- Banana
- Cherry

```
<ol>
  <li>Apple</li>
  <li>Banana</li>
  <li>Cherry</li>
</ol>
```

1. Apple
2. Banana
3. Cherry

表格

- **<table>** 标签定义一个表格[テーブル]。
- 在表格中，**<tr>** 标签定义表格的一行。
- 在一行中，**<td>** 标签定义表格的一个单元格。
- 其中，**<th>** 定义一种特殊的单元格：表头。

Name	Salary
Tom	\$100
Mark	\$150
Zhang	\$130



- 默认的表格是没有边框的，我们之后会学习如何用 CSS 设置边框。

单元格的属性

- 单元格的 **colspan** 属性指定其宽度占几列。
- 单元格的 **rowspan** 属性指定其高度占几行。
- 结合使用， 可以创建各种大小的单元格：

	1	2	3
1	A	B	
2	C	D	
3			

Tips

表格元素除了用于绘制显示数据的表格之外，也能用于帮助我们控制页面布局。

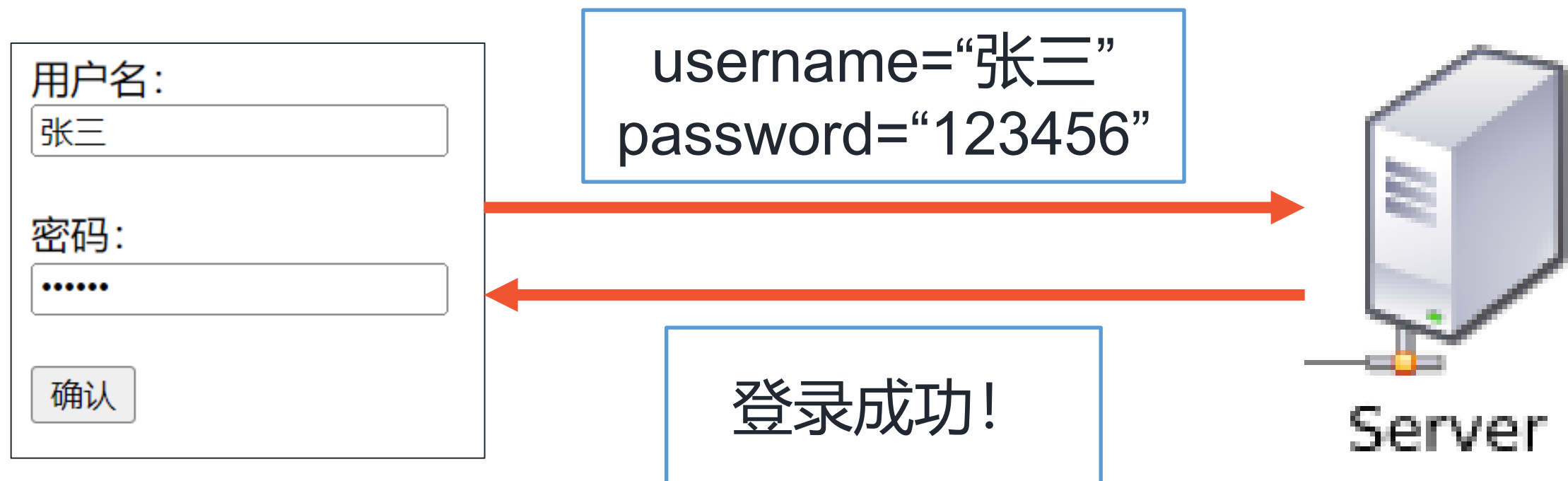
表格的标题

- 在表格里，**<caption>** 标签可以指定表格的标题：

	1	2	3
1	A	B	
2	C	D	
3			

表单

- **<form>** 标签定义一个表单。
- **表单**[フォーム]用于收集用户输入，并将用户输入的数据发送到服务器进行处理。



<input> 标签

- 在表单中， **<input>** 标签定义一个具体的用户输入。
- 通过设置 <input> 标签的 type 属性， 我们可以设置各种不同的输入类型， 如：

类型	定义
text	普通的文本输入
password	密码文本输入
radio	单选按钮
checkbox	多选按钮
file	文件输入
submit	提交按钮

<input> 的属性

- 下表列出了 <input> 标签常用的属性：

属性	定义
id	输入内容的 id（其它元素或代码会使用）
name	输入内容的名称（服务器会使用）
type	输入的类型
value	默认的值
placeholder	没有输入值时显示的文本
required	是否必须输入值
pattern	输入需要满足的格式（正则表达式）

<label> 标签

- 表单中的 **<label>** 标签定义了和输入对应的 **标签文本**[\[ラベル\]](#)。
- 通过使用 **<label>** 标签的 **for 属性**，我们可以指定和它对应的输入是哪个元素。我们需要把**对应输入的 id** 以及 **标签文本的 for** 设为一样的值：

```
<label for="name">用户名: </label>  
<input id="name" type="text">
```



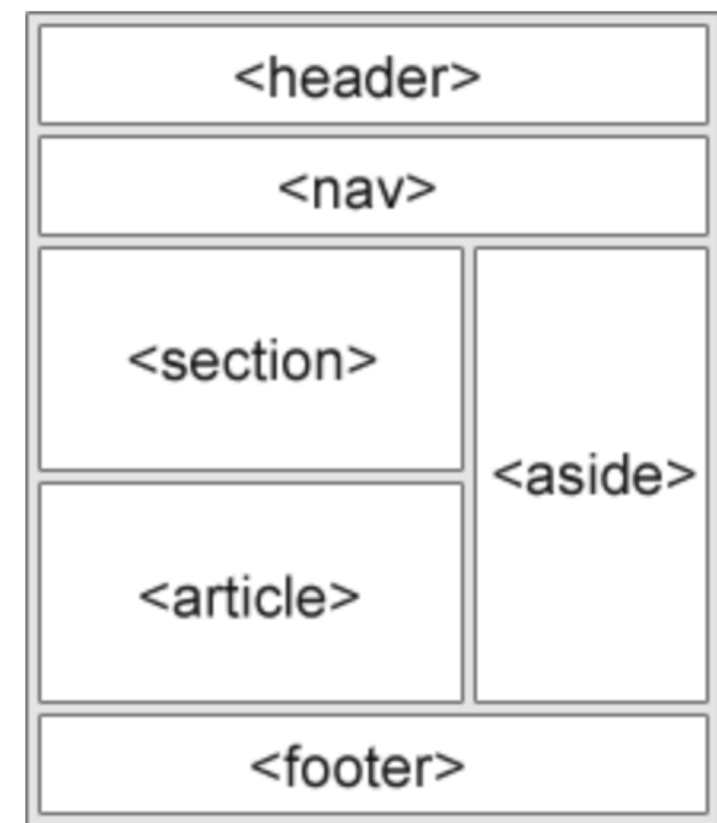
<form> 的 action 和 method 属性

- **action** 属性定义提交表单时发送的 HTTP 请求的地址 (**URL**)。在用户点击提交按钮后，用户的输入数据会提交给 URL 指定的程序，通常是服务端的代码程序。
- **method** 属性定义 HTTP 请求的方法。主要使用 **GET** 和 **POST** 两种方法。
- **GET** 方法将对用户可见，并只能传输少量数据。
- 如果表单数据包含敏感信息或隐私信息；或表单包括较大型数据（如文件），请使用 **POST** 方法。

HTML 语义元素

- HTML 提供了几种**语义元素**[セマンティック要素]，用于说明页面某个部分的含义。
- 在语法上，语义元素与一般 `<div>` 元素没有区别，但浏览器或其他程序（包括我们自己定义的样式表）可以根据其含义实现一些扩张功能。

- `<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



Q & A

Question and answer

Coffee ☕ Break

响应式网页设计

- **响应式网页设计** [レスポンスデザイン] 是指创建在所有设备上看起来都正常的网页。
- 响应式网页设计需要我们针对不同的屏幕尺寸调整元素的显示方式。
- 响应式设计的第一步是设置合适的**视口** [ビューポート]。视口可以由 `<meta>` 标签设置：

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

Coffee ☕ Break

视口

- 视口告诉浏览器网页被创建出来时的大小，从而防止网页不合适的变形：



Exercice ipsum dolor sit amet, consectetur 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. Duis autem vel eum iriure dolor in hendrerit in vulputate velit esse molestie consequat, vel illum dolore eu feugiat nulla facilisis at vero eros et accumsan et iusto odio dignissim qui blandit praesent luptatum zzril delenit augue dui dolore te feugait nulla facilisi. Nam liber tempor cum soluta nobis eleifend antion conone nihil innoerdiet domine

使用视口meta标签：



Lorem ipsum dolor sit amet, consectetur 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. Duis autem vel eum iriure dolor in hendrerit in vulputate velit esse molestie consequat, vel illum dolore eu feugiat nulla facilisis at vero eros et accumsan et iusto odio dignissim qui blandit praesent luptatum zzril delenit augue dui dolore te feugait nulla facilisi. Nam liber tempor cum soluta nobis eleifend antion conone nihil innoerdiet domine

Coffee ☕ Break

URL

- **URL** (Uniform Resource Locators) 代表了互联网上某一个文件的地址。我们一般所说的“网址”也是由 URL 表达的。
- 互联网上文件的 URL 一般遵循以下语法格式：
`scheme://prefix.domain:port/path/filename`
 - **scheme** 定义这是一个互联网上的地址（最常见的是 http 或 https）；
 - **prefix** 定义域名前缀（http 的默认值为 www）；
 - **domain** 定义域名（如 google.com）；
 - **port** 定义主机上的**端口号**[ポート]（http 的默认值为80）；
 - **path** 定义服务器上的**路径**（文件夹）；
 - **filename** 定义文件的名称（如 index.html）。

总结

Sum Up

1. 网页开发的基本概念：
 - ① 服务端、客户端和 HTTP 请求。
 - ② HTML、CSS 和 JavaScript 的关系。
2. HTML 的基本语法：元素、标签和属性。
3. HTML 的常用标签：
 - ① <head> 标签与元数据；
 - ② 文本、图像、超链接等基本标签；
 - ③ 表格、列表、表单等容器标签。

THANK YOU!