



目录.

1. Introduction to HTML

2. Multimedia and embedding

3. HTML Tables.

Part I: Introduction to HTML

1. Introduction to HTML Overview

① Getting Started with HTML

② What's in the head? Metadata in HTML

③ HTML text fundamentals

④ Creating hyper links

⑤ Advanced text formatting

⑥ Document and website structure

⑦ Debugging HTML

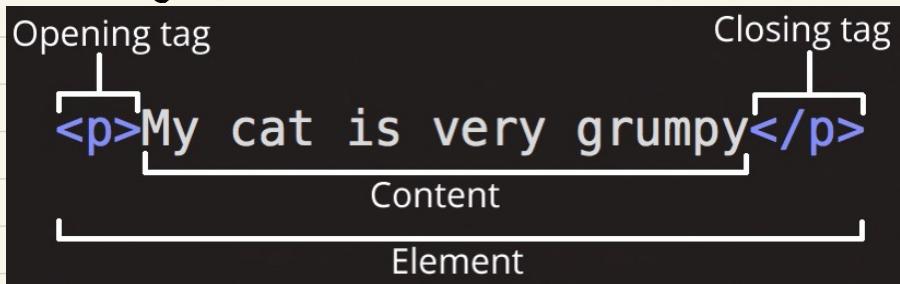
2. Getting Started with HTML

2.1 What is HTML

HTML不是编程语言，是一种标记语言。

HTML对大小写不敏感，最佳实践均采用小写

2.2 Anatomy of an HTML Element



2.2.1 Block versus inline Elements 块级元素和行内元素

- Block-level elements form a visible block on a page.
A block-level element appears on a new line following the content that precedes it. Any content that follows a block-level element also appears on a new line.
Block-level elements are usually structural elements on page. For example, a block-level element might represent headings, paragraphs, lists, navigation menus, or footers.
A block-level element wouldn't be nested inside an inline element, but it might be nested in another block-level element.

- inline elements are contained within block-level elements, and surround only small parts of the documents' content (not entire paragraphs or groupings of content). An inline element will not cause a newline to appear in the document.
It is typically used with text, for example, an `<a>` element creates hyperlink, and elements such as `` or `` create emphasis

Note: HTML5重新定义了元素分类，不再只是 block 和 inline

Note: HTML 中的 block 和 inline 不能和 CSS 中的盒子类型 (Type of CSS boxes) 搞混，尽管它们的名字相同。
改变 CSS 的 display 不会改变元素的分类，也不会影响它能嵌套的元素以及它能被哪些元素嵌套。

2.2.2 Empty elements 空元素

Not all elements follow the pattern of an opening tag, content and a closing tag.

Some elements consist of a single tag, which is typically used to insert/embed something in the document.

For example, the `` element embeds an image file onto a page.

2.3 Attributes

```
<p class="editor-note">My cat is very grumpy</p>
```

Attribute
↓
class="editor-note"

Elements can also have attributes.

Attributes contain extra information about the element that won't appear in the content.

2.3.1 Boolean attributes. 布尔attribute

Sometimes you will see attributes written without values. These are called Boolean attributes.

Boolean attributes can only have one value which is generally the same as the attribute name.

只要元素上写了这个attribute，那不管给它赋什么值，空值也好，它都是true。

2.3.2 Omitting quotes around attribute values 值省略双引号

attribute的值可以不打双引号，但最好别这么做

```
<a href="https://www.mozilla.org/">favorite website</a>
```

2.3.3 Single or double quotes? 单引号还是双引号

都可以，也可以相互嵌套，但一定要正确匹配。

2-4 Anatomy of an HTML Document

2-4-1 How individual elements combine to form an entire HTML page.

```
<!DOCTYPE html>
<html>
  <head>
    <meta charset="utf-8">
    <title>My test page</title>
  </head>
  <body>
    
  </body>
</html>
```

① `<!DOCTYPE html>` doctype 文档类型

在以前，doctype 像是一个指向一个规则集的 link

② `<html></html>` `<html>` 元素：文档根元素
root element

③ `<head></head>` `<head>` 元素：非内容要素容器

including things like: keywords, page description,

CSS, character set declarations 等等。

④ `<meta charset = "utf-8">` 文档编码，
utf8应该是最通用的

⑤ `<title>` 标题

⑥ `<body>` 主体内容容器

2.4.2 Whitespace in HTML

No matter how much whitespace you use inside HTML element Content (which can include one or more space character, but also line breaks), the HTML parser reduces each sequence of whitespace to a single space when rendering the code.

不管使用多少空格，都会被 HTML parser 缩减成 1 个。

<p>Dogs are silly.</p>

<p>Dogs are
silly.</p>

上面两段代码的效果是一样的

2.5 Entity references: Including special characters in HTML

转义字符

In HTML, the characters < > " , and & are special characters. They are part of syntax itself.

So how do you include one of these special characters in your text: Entity references.

Literal character	Character reference equivalent
<	<
>	>
"	"
'	'
&	&

lt = less than

gt = greater than

quot = quotation

~~~

## 2.6 HTML Comments

<!-- comments -->

## 3. What's in the head? Metadata in HTML

The head of an HTML document is the part that is not displayed in the web browser when the page is loaded.

Web browser use information contained in the head to render the HTML document correctly.



**Object:** To learn about the HTML head, its purpose, the most important items it can contain, and what effect it can have on the HTML document.

### 3.1 What is the HTML head?

The head's job is to contain **metadata** about the document.

In larger pages, the head can get quite large.

### 3.2 Adding a title

```
<head>
  <meta charset="utf-8">
  <title>My test page</title> ←
</head>
```

### 3.3 Metadata: the `<meta>` element

Metadata is data that describes data 描述数据的数据

HTML 通过 `<meta>` 元素来添加 metadata.

#### 3.3.1 Specifying your document's character encoding 文档编码

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

你应该永远使用 utf-8 编码

#### 3.3.2 Adding an author and description

Many `<meta>` elements include name and content attributes.

- name specifies the type of meta element it is
- content specifies the actual meta content

```
<meta name="author" content="Chris Mills">
```

```
<meta name="description" content="The MDN Web Docs Learning Area aims to provide  
complete beginners to the Web with all they need to know to get  
started with developing web sites and applications.">
```

Note: Many `<meta>` features just aren't used anymore  
for example, the keyword `<meta>` element.

#### 3.3.3 Other types of metadata

有很多 metadata 是某些公司的 proprietary creations (专有创作)

## 比如 Facebook 的 Open Graph Data

```
<meta property="og:image" content="https://developer.mozilla.org/static/img/opengraph-logo.png"
<meta property="og:description" content="The Mozilla Developer Network (MDN) provides
information about Open Web technologies including HTML, CSS, and APIs for both Web sites
and HTML5 Apps. It also documents Mozilla products, like Firefox OS.">
<meta property="og:title" content="Mozilla Developer Network">
```

当网页中添加这些 metadata 后，其链接被分享到特定的网站时，会被转成好看的卡片形式，例如 MDN 被分享到 Facebook 时，会变成下面这样：



### 3.4 Adding custom icons to your site

**favicon** = favorites icon，用在 Tab 标栏和书签栏

favicon 的图片格式是 .xx.ico 也可以使用 .gif 或 .png

```
<link rel="icon" href="favicon.ico" type="image/x-icon">
```

## 3.5 Applying CSS and JavaScript to HTML

- The `<link>` element should always go inside the head of your document  
This takes two attributes: `rel="stylesheet"` and `href`

```
<link rel="stylesheet" href="my-css-file.css">
```

- The `<script>` element should also go into the head, and should include a `src` attribute containing the path to the JavaScript and `defer` which basically instructs the browser to load the JavaScript after the page has finished parsing the HTML. This is useful as it makes sure that the HTML is all loaded before the JavaScript runs.

```
<script src="my-js-file.js" defer></script>
```

See also Script loading strategies

## 3.6 Setting the primary language of the document

It's worth mentioning that you can (and really **should**) set the language of your page. This can be done by adding the `lang` attribute to the opening HTML tag

```
<html lang="en-US">
```

这有利于搜索引擎的搜索排名，也可以帮助有视力障碍的人使用 Screen Reader.

## 4. HTML text fundamentals



**Objective:** Learn how to mark up a basic page of text to give it **structure** and **meaning** — including paragraphs, headings, lists, emphasis and quotations.

### 4.1 The basics: headings and paragraphs

Structured content makes the reading experience easier and more enjoyable

#### 4.1.1 Implementing structural hierarchy 实现结构层次

建立结构层次的最佳实践：

① 一个页面只能有一个 `<h1>` 这是最高层次（主标题）

② 不要使用 `<h3>` 作为副标题

③ 一个页面最好不要超过 3 个层次结构，如果有更深的结构，建议扩展页面。

#### 4.2.2 Why do we need structure?

★ 你对语义化的认识？

我觉得语义化就是让信息被更高效的传达，实行方式是

① 合理的页面结构安排 ② 合适的元素标签选用

信息传达的对象有两个：人和搜索引擎

① 对于人来说，可以从更高效的获取页面上他所需要的内容。

② 对于搜索引擎来说，有助于搜索排名的提高。

## 4.2 Lists

4.2.1 Unordered <ul>

4.2.2 Ordered <ol>

4.2.3 Nesting Lists 多级列表

## 4.3 Emphasis and Importance

Emphasis 强调，重读 <em> 浏览器默认是斜体 (italic).  
但不要用这个标签来表示斜体。

importance 强调重要性 <strong> 浏览器默认是粗体 (bold).  
但不要用这个标签来表示粗体。

Italic, bold and underline <i> <b> <u>  
斜体，粗体，下划线。

They came about so people could write bold, italics, or underlined  
text in an era when CSS was still supported poorly or not  
at all.

Elements like this, which only affect presentation and not semantics, are known as presentational Elements and should no longer be used.

### Best rule of thumb (经验法则)

<i> = 外文词汇, 分类名称、术语, 想法, 专用词汇

<b> = 关键词、产品名, 引导语

<u> = 正确的名字、拼写错误

## 5. Creating hyper links



Objective: To Learn how to implement a hyperlink effectively and Link multiple files together

### 5.1 What is a hyper link?

Hyperlinks allow us to link documents to other documents or resources.

Note =

**Note:** A URL can point to HTML files, text files, images, text documents, video and audio files, or anything else that lives on the Web. If the web browser doesn't know how to display or handle the file, it will ask you if you want to open the file (in which case the duty of opening or handling the file is passed to a suitable native app on the device) or download the file (in which case you can try to deal with it later on).

## 5.2 Anatomy of a link

A basic link is created by wrapping the text or other content inside an `<a>` element and using the `href` attribute

```
<p>I'm creating a link to  
<a href="https://www.mozilla.org/en-US/">the Mozilla homepage</a>.  
</p>
```

### 5.2.1 Adding supporting information with the title attribute

The `title` contains additional information about the link, such as which kind of information the page contains, or things to be aware of on the website.

```
<p>I'm creating a link to  
<a href="https://www.mozilla.org/en-US/"  
    title="The best place to find more information about Mozilla's  
    mission and how to contribute">the Mozilla homepage</a>.  
</p>
```

### 5.2.2 Block level Links

Almost any content can be made into a link, even block-level elements.

```
<a href="https://www.mozilla.org/en-US/">  
      
</a>
```

## 5.3 A quick primer on URLs and paths

To fully understand link targets, need to understand URLs and file paths.

A URL = Uniform Resource Locator

### 5.3.1 Document fragments

It's possible to link to a specific part of an HTML document, known as a document fragment.

To do this:

- ① Assign an id attribute to the element you want to link to

```
<h2 id="Mailing_address">Mailing address</h2>
```

- ② Then to link to that specific id, you'd include it at the end of the URL, preceded by a hash/pound symbol (#)

```
<a href="contacts.html#Mailing_address">mailing address</a>
```

```
<a href="#Mailing_address">company mailing address</a>
```

## 5.3.2 Absolute vs relative URLs

absolute URL = points to a location defined by its absolute location on the web, including protocol and domain  
协议 域名

relative URL = Points to a location that is relative to file you are linking from

A relative URL will point to different places depending on the actual location of the file you refer from.

## 5.4 Link best practises

### 5.4.1 Use clear link wording

Good link text: [Download Firefox](#) ☺

```
<p><a href="https://firefox.com/">  
    Download Firefox  
</a></p>
```



Bad link text: [Click here](#) ☹ to download Firefox

```
<p><a href="https://firefox.com/">  
    Click here  
</a>  
    to download Firefox</p>
```



Other TIPS:

① 不要使用链接作为link text，很丑

② Don't say "link" or "link to" in the link text

③ keep your link text as short as possible

### 5.4.2 Linking to non-HTML Resources — leave clear signposts.

清晰木标识

Let's look at some examples, to see what kind of text can be used here:

```
<p><a href="https://www.example.com/large-report.pdf">  
    Download the sales report (PDF, 10MB)  
</a></p>  
  
<p><a href="https://www.example.com/video-stream/" target="_blank">  
    Watch the video (stream opens in separate tab, HD quality)  
</a></p>  
  
<p><a href="https://www.example.com/car-game">  
    Play the car game (requires Flash)  
</a></p>
```

### 5.4.3 Use the download attribute when linking to a download

You can use the download attribute name to provide a default save file name.

```
<a href="https://download.mozilla.org/?product=firefox-latest-ssl&os=win64&lang=en-US"  
    download="firefox-latest-64bit-installer.exe">  
    Download Latest Firefox for Windows (64-bit) (English, US)  
</a>
```

## 5.5 E-mail links

In its most basic and commonly used form, a mailto: link indicates the email address of the intended recipient.

```
<a href="mailto:nowhere@mozilla.org">Send email to nowhere</a>
```

### 5.5.1 specifying details

具体内容查看相关文档

## b. Advanced text formatting



**Objective:** To learn how to use lesser-known HTML elements to mark up advanced semantic features

### b.1 Description lists

描述列表

The purpose of Description lists is to mark up a set of items and their associated descriptions, such as terms and definitions, or questions and answers.

<dl> description lists

<dt> description term

<dd> description definition

#### b.1.1 Description list example

```
<dl>
  <dt>soliloquy</dt>
  <dd>In drama, where a character speaks to themselves, representing their inner thoughts
  <dt>monologue</dt>
  <dd>In drama, where a character speaks their thoughts out loud to share them with the audience
  <dt>aside</dt>
  <dd>In drama, where a character shares a comment only with the audience for humorous or
</dl>
```

#### b.1.2 Multiple descriptions for one term

```
<dl>
  <dt>aside</dt>
  <dd>In drama, where a character shares a comment only with the audience for humorous or
  <dd>In writing, a section of content that is related to the current topic, but doesn't
</dl>
```

## 6.2 Quotations 引用

### 6.2.1 Blockquotes

if a section of block level content is quoted from somewhere else, you should wrap it inside a `<blockquote>` element to signify this, and include a URL pointing to the source of the quote inside a `cite` attribute

```
<p>Here below is a blockquote...</p>
<blockquote cite="https://developer.mozilla.org/en-US/docs/Web/HTML/Element/blockquote">
  <p>The <strong>HTML <code>&lt;blockquote&gt;</code> Element</strong> (or <em>HTML Block Quotation Element</em>) indicates that the enclosed text is an extended quotation.</p>
</blockquote>
```

### 6.2.2 inline quotations.

use the `<q>` element with a `site` attribute.

## 6.3 Abbreviations 缩写

首字母缩写

`<abbr>` this is used to wrap around an abbreviation or acronym, and provide a full expansion of the term (→ included inside a `title` attribute)

### 6.3.1 Abbreviation example

```
<p>We use <abbr title="Hypertext Markup Language">HTML</abbr> to structure our
```

```
<p>I think <abbr title="Reverend">Rev.</abbr> Green did it in the kitchen with
```

## 6.4 Marking up contact details 联系方式

<address> : element for marking up contact details.

for example

```
<address>
  Chris Mills, Manchester, The Grim North, UK
</address>
```

It can include more complex markup

```
<address>
  <p>
    Chris Mills<br>
    Manchester<br>
    The Grim North<br>
    UK
  </p>

  <ul>
    <li>Tel: 01234 567 890</li>
    <li>Email: me@grim-north.co.uk</li>
  </ul>
</address>
```

## b.5 SuperScript and subScript 上标和下标

```
<p>My birthday is on the 25<sup>th</sup> of May 2001.</p>
<p>Caffeine's chemical formula is C<sub>8</sub>H<sub>10</sub>N<sub>4</sub>O<sub>2</sub>. </p>
<p>If x<sup>2</sup> is 9, x must equal 3 or -3.</p>
```

The output of this code looks like so:

My birthday is on the 25<sup>th</sup> of May 2001.  
Caffeine's chemical formula is C<sub>8</sub>H<sub>10</sub>N<sub>4</sub>O<sub>2</sub>.

## b.6 Representing computer code 代码

There are a number of elements available for marking up code

<code> marking up generic pieces of computer code

<pre> for retaining whitespace (generally code blocks)

<var> marking up variables

<kbd> marking up keyboard input entered into the computer

<samp> marking up the output of a program.

## 6.7 Marking up times and dates 时间和日期

```
<time datetime="2016-01-20">20 January 2016</time>
```

```
<!-- Standard simple date -->
<time datetime="2016-01-20">20 January 2016</time>
<!-- Just year and month -->
<time datetime="2016-01">January 2016</time>
<!-- Just month and day -->
<time datetime="01-20">20 January</time>
<!-- Just time, hours and minutes -->
<time datetime="19:30">19:30</time>
<!-- You can do seconds and milliseconds too! -->
<time datetime="19:30:01.856">19:30:01.856</time>
<!-- Date and time -->
<time datetime="2016-01-20T19:30">7.30pm, 20 January 2016</time>
<!-- Date and time with timezone offset -->
<time datetime="2016-01-20T19:30+01:00">7.30pm, 20 January 2016 is 8.30pm in France</time>
<!-- Calling out a specific week number -->
<time datetime="2016-W04">The fourth week of 2016</time>
```

## 7. Document and website structure



**Objective:** learn how to structure your document using semantic tags, and how to work out the structure of a simple website

### 7.1 Basic sections of a document

Webpages can and will look pretty different from one another, but they all tend to share similar standard components:

- ① header
- ② Navigation bar
- ③ main content
- ④ sidebar
- ⑤ footer

A "typical website" could be structured something like this:

The diagram illustrates a website structure with the following components:

- Header:** A grey header bar containing the word "Header".
- Navigation Bar:** A pink navigation bar below the header with links to "HOME", "OUR TEAM", "PROJECTS", and "CONTACT". It also features a search bar with a placeholder "Search query" and a "Go!" button.
- Main Content Area:** A white rectangular area containing:
  - Article heading:** *Article heading*
  - Text:** *... ipsum dolor sit amet, consectetur adipiscing elit. Donec a diam lectus. Set sit amet ipsum mauris. Maecenas congue ligula as quam viverra nec consectetur ant hendrerit. Donec et mollis dolor. Praesent et diam eget libero egestas mattis sit amet vitae augue. Nam tincidunt congue enim, ut porta lorem lacinia consectetur.*
  - Section:** *subsection*
  - Text:** *Donec at libero sed accu vehicula ultricies a non tortor. Lorem ipsum dolor sit amet, consectetur adipiscing elit. Aenean ut gravida lorem. Ut turpis felis, pulvinar a semper sed, adipiscing id dolor.*
  - Text:** *Pellentesque autor nisi id magna consequat sagittis. Curabitur dapibus, enim sit amet elit pharetra tincidunt feugiat nisl imperdiet. Ut convallis libero in una ultrices accumsan. Donec sed odio eros.*
  - Section:** *Another subsection*
  - Text:** *Donec viverra mi quis quam pulvinar at malesuada arcu rhoncus. Cum sociis natoque penatibus et manis dis parturient montes, nascetur ridiculus mus. In rutrum accumsan ultricies. Mauris vitae nisi at sem facilisis semper ac in est.*
  - Text:** *Vivamus fermentum semper porta. Nunc diam velit, adipiscing ut tristique vitae sagittis vel odio. Maecenas convallis ullamcorper ultricies. Curabitur ornare, ligula semper consectetur sagittis, nisi diam iaculis velit, et fringilla sem nunc vel mi.*
- Sidebar:** A pink sidebar on the right side containing:
  - Related:**
    - Oh I do like to be beside the seaside*
    - Oh I do like to be beside the sea*
    - Although in the North of England*
    - It never stops raining*
    - Oh well...*
- Page Footer:** A small white footer bar at the bottom with the text "©Copyright 2050 by nobody. All rights reversed."

## 7.2 HTML for structuring content

We need to respect semantics and use the right element for the right job.

In your HTML code, you can mark up sections of content based on their functionality

HTML provides dedicated tags that you can use to represent such Sections:

header = <header>

navigation bar = <nav>

main content <main> with various subsections represented by <article> <section> and <div>

sidebar = <aside> often placed inside <main>

footer = <footer>

## 7.3 HTML layout Elements in more detail

It's good to understand the overall meaning of all the HTML sectioning elements in detail (要慢慢来)

For now, these are the main definitions that you should try to understand:

- `<main>` is for content *unique to this page*. Use `<main>` only *once* per page, and put it directly inside `<body>`. Ideally this shouldn't be nested within other elements.
- `<article>` encloses a block of related content that makes sense on its own without the rest of the page (e.g., a single blog post).
- `<section>` is similar to `<article>`, but it is more for grouping together a single part of the page that constitutes one single piece of functionality (e.g., a mini map, or a set of article headlines and summaries), or a theme. It's considered best practice to begin each section with a `heading`; also note that you can break `<article>`s up into different `<section>`s, or `<section>`s up into different `<article>`s, depending on the context.
- `<aside>` contains content that is not directly related to the main content but can provide additional information indirectly related to it (glossary entries, author biography, related links, etc.).
- `<header>` represents a group of introductory content. If it is a child of `<body>` it defines the global header of a webpage, but if it's a child of an `<article>` or `<section>` it defines a specific header for that section (try not to confuse this with `titles` and `headings`).
- `<nav>` contains the main navigation functionality for the page. Secondary links, etc., would not go in the navigation.
- `<footer>` represents a group of end content for a page.

### 7.3.1 Non-Semantic Wrappers

Sometimes you'll come across a situation where you can't find an ideal semantic element to group some items together or wrap some content.

For cases like these, HTML provides `<div>` and `<span>` elements

## 不要过分依赖 div

**Warning:** Divs are so convenient to use that it's easy to use them too much. As they carry no semantic value, they just clutter your HTML code. Take care to use them only when there is no better semantic solution and try to reduce their usage to the minimum otherwise you'll have a hard time updating and maintaining your documents.

### 7.3.2 Linebreaks and horizontal rules

Two elements that you'll use occasionally and will want to know about are `<br>` and `<hr>`

`<br>` = the linebreak element

`<br>` creates a line break in a paragraph.

It is the **only** way to force a rigid structure in a situation where you want a series of fixed short lines.

`<p>There once was a man named O'Dell<br>`

Who loved to write HTML`<br>`

But his structure was bad, his semantics were sad`<br>`  
and his markup didn't read very well.`</p>`

(考题)

`<hr>` = the thematic break element

It create a horizontal rule in the document the denotes a thematic change in the text.



## (练习项目)

### 7.4 planning a simple website

- ① Bear in mind that you'll have a few elements common to most pages = such as the navigation menu, and footer content.

Note down what you want to have common to every page.

Common to every page

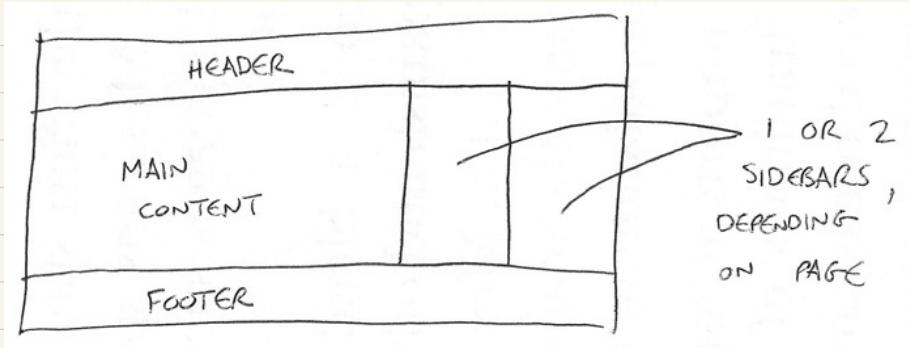
Header: title & logo

Footer: contact details and copyright notice  
Links to <sup>①</sup> terms + conditions

② Site language chooser

③ Accessibility policy

- ② Next, draw a rough sketch of what you might want the structure of each page to look like



③ Now, brainstorm all the other (not common to every page) content you want to have on your website

Write a big list down

Search for flights  
Hotels / other accommodation  
Transport  
Things to do

Special offers

Popular holiday packages

e.g. Winter sun

Disneyworld

Skiing

Search results

Country-specific info

Accommodation / attraction reviews

Visa / entry requirements

Money / Currency

Languages

Insurance

Buy  
holidays / other things

④ Next, try to sort all these content into groups 分类

### Search

Flights  
Hotels  
Other accommodation  
Transport  
Things to do

### Specials

Special offers

Popular holidays

### Search results

Also include search functionality here to refine results

### Country-specific info

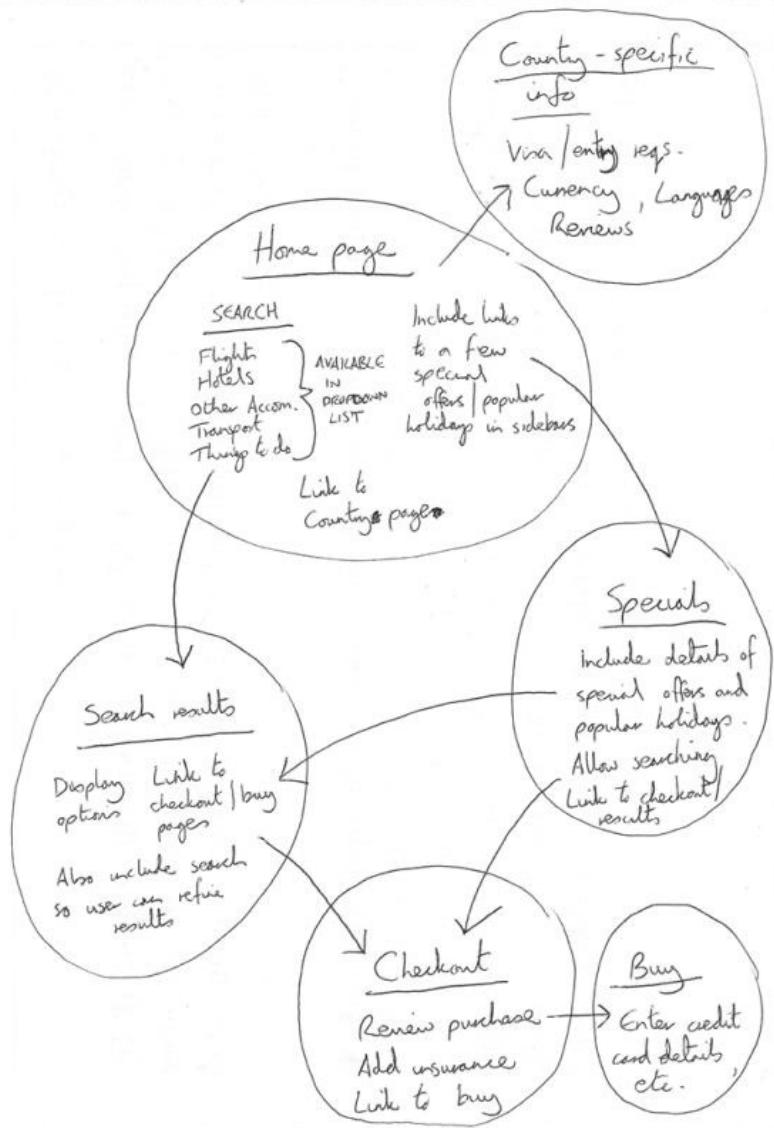
Reviews  
Visa / entry requirements  
Money / currency  
Languages

### Buy things

Shopping cart  
Checkout  
Insurance

⑤ Now try to sketch a rough sitemap

have a bubble for each page and draw lines to show the typical workflow between pages



## 8. Debugging HTML



**Objective:** Learn the basics of using debugging tools to find problems in HTML

### 8.1 Debugging isn't scary

Debugging doesn't have to be scary though — **the key** to being comfortable with writing and debugging any programming language or code is **familiarity** with both the **language and Tools.**

熟悉语言和工具

### 8.2 HTML and debugging

#### 8.2.1 permissive code

So what do we mean by **permissive**?

There are two main types of error you'll come across:

① **syntax errors:** 语法错误。随着对语言语法的熟悉，此类错误很容易解决

② **logic errors:** 逻辑错误。此类问题比较难解决，因为没有可以直接定位原因的错误信息

HTML 没有语法错误 because browser parse it **permissively**. meaning that the page still displays even if there are syntax errors. 浏览器会按照它的 **built-in rules** (内置规则) 进行修复

## 8.2.2 HTML Validation

In a small example, it is easy to search through the lines and find errors. But what about a huge, complex HTML document?

The Best Strategy is to start by running your HTML page through the Markup Validation Service

(<https://validator.w3.org>)

This webpage takes an HTML document as an input, goes through it and gives you a report to tell you what is wrong with your HTML

## 8.2.3 Interpreting the error messages

The error messages are usually helpful, but sometimes they are not so helpful; with a bit of practise you can work out how to interpret these to fix your code.

**Note:** An attribute missing a closing quote can result in an open element because the rest of the document is interpreted as the attribute's content.

例子看 MDN 文档

## Part II Multimedia and embedding

### 1. Multimedia and embedding overview

① Images in HTML

② Video and audio content

③ From <object> to <iframe> other embedding technologies

④ Adding vector graphics to the web

⑤ Responsive Images

## 2. Images In HTML



**Objective:** To learn how to embed simple images in HTML, annotate them with captions, and how HTML images relate to CSS background images.

### 2.1 How do we put an image on a webpage?

We use `<img>` element, this is an **empty element** (meaning that it has no text content or closing tag) that requires a minimum of one attribute to be useful = `src`

① 图片在同级目录下

```

```

使用相对路径

② 图片在同级子目录下

```

```

**Note:** Search engines also read image filenames and count them towards SEO. Therefore, you should give your image a descriptive filename; `dinosaur.jpg` is better than `img835.png`.

③ 使用绝对路径

```

```

最好别这样使用，因为这会导致浏览器做额外的工作：从DNS服务器查找域名的IP地址。



Note: Elements like `<img>` and `<video>` are sometimes referred to as **replaced elements**. This is because the element's content and size are defined by an external resource (like an image or video file), not by the contents of the element itself. You can read more about them at [Replaced elements](#).

Note: CSS 对 替换元素唯一的影响就是对它的内容在盒子中的位置。

## 2.1.1 Alternative text

故意把图片名称写错

alt attribute (the easiest way to **test** your alt text is to purposely misspell your filename)

## 2.1.2 Width and height

You can use the width and height attributes to specify the width and height of your image.

```

```

在 `<img>` 的属性上设定图片的宽高可以让页面更快加载和平滑加载。

但是！千万不能用 HTML attributes 去修改图片尺寸

如果需要修改，用 CSS 去修改

## 2-13 Images title

```

```



但是并不推荐使用这个attribute

说明文字

## 2.2 Annotating images with figures and figure captions

There are a number of ways that you could add a caption to go with your image, for example:

```
<div class="figure">
  

  <p>A T-Rex on display in the Manchester University Museum.</p>
</div>
```

但是这会出现图片和 caption 不匹配的问题

更好的解决方案是使用 `<figure>` 和 `<figcaption>` 元素

```
<figure>
  

  <figcaption>A T-Rex on display in the Manchester University Museum.</figcaption>
</figure>
```

`<figure>` 不单单只为图片服务，也可以为代码段、音频、视频、公式、表格等服务

## 2-3 CSS background images.

```
p {
  background-image: url("images/dinosaur.jpg");
}
```

if an `image` has meaning, in terms of your content, you should use an HTML `image`

if an `image` is purely decoration, you should use CSS `background images`.

### 3. Video and audio content



**Objective:** To learn how to embed video and audio content into a web page, and add captions/subtitles to video

#### 3.1 Video and audio on web

以前很长一段时间网页都没有播放视频和音频的功能，只能使用 flash

后来 HTML5 标准添加添加了 `<video>` 和 `<audio>` 元素以及控制它们的 JavaScript API

##### 3.1.1 The `<video>` element

```
<video src="rabbit320.webm" controls>
  <p>Your browser doesn't support HTML5 video.
  Here is a <a href="rabbit320.webm">link to the video</a> instead.</p>
</video>
```

`src` = path to the video

`controls` = 播放控件，你可以使用这个 attribute 也可以使用相关的 JavaScript API 自定义自己的控件。



(总有一天我要做一个自己的播放控件)

The paragraph inside `<video>` tags = This is called fallback content (后备内容), 当浏览器不支持 `<video>` 标签时提供备用措施。

使用多种资源格式提高兼容性

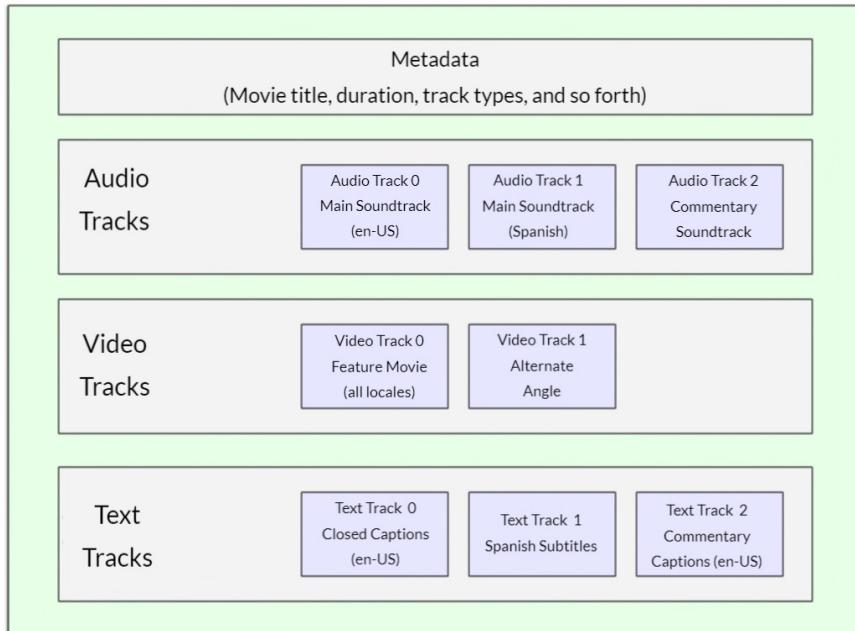
### 3.1.2 Using multiple source formats to improve compatibility

Different browsers support different video (and audio) formats.

#### Contents of a media file

封装格式

Formats like MP3, MP4 and WebM are called **Container formats**. They define a **structure** in which the audio and/or video **tracks** that make up the media are stored, along with **metadata** describing the media, what **codecs** are used to encode its channel (编解码器)



Video and audio Tracks hold **data** in the appropriate format for the **codecs** used to encode that media

Each audio track is encoded using an audio codec  
(data) 编码器

Each video track is encoded using a video codec

不同的封装格式包含不同的音视频编码器

举个例子：

WebM = 音频 codec = Vorbis、Opus  
视频 codec = VP8、VP9

MP4 = 音频 codec = AAC、MP3  
视频 codec = H.264

Ogg = 音频 codec = Vorbis  
视频 codec = Theora

一些特有的音视频封装格式所使用的 codec 就是它本身，比如：FLAC

Codec 的 **主要任务** 是对音视频进行压缩，因为原始的数据通常来说都很大

每个浏览器都支持一系列的 codecs，每种 codec 都有其优缺点

有一件事需要了解：同一个浏览器的 PC 版和 mobile 版可能支持的音视频格式也不相同

如何解决这个问题：

不同的浏览器，甚至同一浏览器的不同版本支持的音视频格式不相同

```
<video controls>
  <source src="rabbit320.mp4" type="video/mp4">
  <source src="rabbit320.webm" type="video/webm">
  <p>Your browser doesn't support HTML5 video.
    Here is a <a href="rabbit320.mp4">link to the video</a> instead.</p>
</video>
```

将 src attribute 从 <video> 元素上移出来，用 <source> 元素进行代替，

浏览器会从上到下找它支持的格式，一旦它找到了它支持的格式，就使用那种格式的音视频。

每个 <source> 元素都有一个 type attribute，建议最好加上它。

因为没有 type 的话，浏览器会加载每一个资源去看它支不支持，加上 type 的话，浏览器会直接跳过它不支持的格式。

### 3.1.3 Other <video> features

```
<video controls width="400" height="400"  
        autoplay loop muted preload="auto"  
        poster="poster.png">  
    <source src="rabbit320.mp4" type="video/mp4">  
    <source src="rabbit320.webm" type="video/webm">  
    <p>Your browser doesn't support HTML video.  
        Here is a <a href="rabbit320.mp4">link to the video</a> instead.</p>  
</video>
```

width and height : 可以通过 attribute 或 CSS 设置视频宽高

autoplay : Makes the audio or video start playing right now, while the rest of the page is loading 最好别自动播放

loop : 循环播放

muted : 静音播放

poster : The URL of an image which will be displayed before the video is played.

preload : Used for buffering large files ; it can take one of three values :

- "none" does not buffer the file
- "auto" buffers the file
- "metadata" buffers only the metadata for the file

### 3.1.4 The <audio> element

```
<audio controls>
  <source src="viper.mp3" type="audio/mp3">
  <source src="viper.ogg" type="audio/ogg">
  <p>Your browser doesn't support HTML5 audio.
    Here is a <a href="viper.mp3">link to the audio</a> instead.</p>
</audio>
```

<audio>做的事情跟<video>差不多，只是没有  
下面这些 attributes:

width、height、poster

### 3.2 Displaying video text tracks 字幕

我们可以使用 WebVTT 格式文件和<track>元素来添加字幕

A typical WebVTT file will look something like this:

```
WEBVTT

1
00:00:22.230 --> 00:00:24.606
This is the first subtitle.

2
00:00:30.739 --> 00:00:34.074
This is the second.

...
```

有几种不同类型的字幕

subtitles = 外文翻译字幕

captions = 同步字幕

timed descriptions = 定时字幕

To get this displayed along with the HTML media playback, you need to:

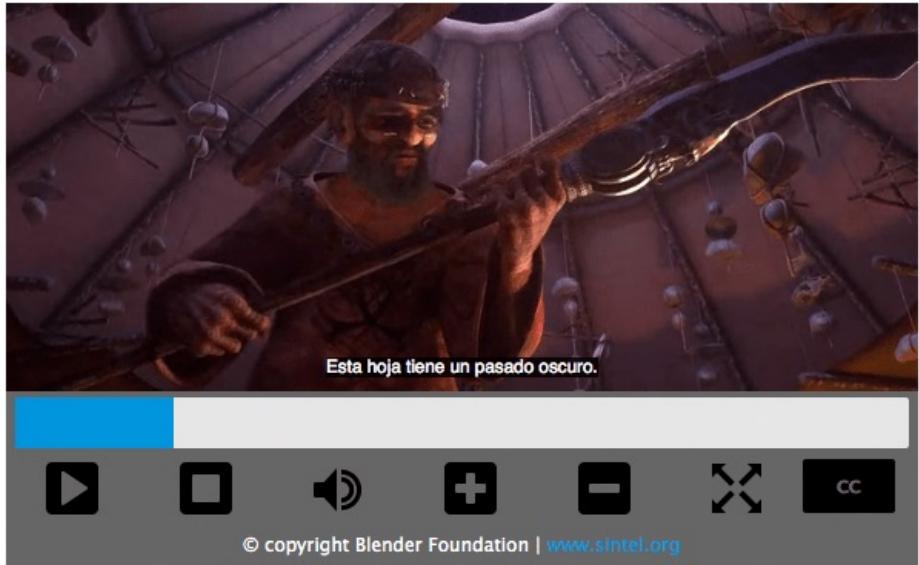
1. Save it as a `.vtt` file in a sensible place.
2. Link to the `.vtt` file with the `<track>` element. `<track>` should be placed within `<audio>` or `<video>`, but after all `<source>` elements. Use the `kind` attribute to specify whether the cues are subtitles, captions, or descriptions. Further, use `srclang` to tell the browser what language you have written the subtitles in. Finally, add `label` to help readers identify the language they are searching for.

Here's an example:

```
<video controls>
  <source src="example.mp4" type="video/mp4">
  <source src="example.webm" type="video/webm">
  <track kind="subtitles" src="subtitles_es.vtt" srclang="es" label="Spanish">
</video>
```



This will result in a video that has subtitles displayed, kind of like this:



## 4. From object to iframe - other embedding technologies



**Objective:** To Learn how to embed items into web pages using `<object>` `<embed>` and `<iframe>` like pdf documents and other web pages.

### 4.1 A short history of embedding

A long time ago on the Web, it was popular to use **frames** to create websites — small parts of a website stored in individual HTML pages. These were embedded in a master document called a **frameset**, which allowed you to specify the area on the screen that each frame filled, rather like sizing the columns and rows of a table. These were considered the height of coolness in the mid to late 90s, and there was evidence that having a webpage split up into smaller chunks like this was better for download speeds — especially noticeable with network connections being so slow back then. They did however have many problems, which far outweighed any positives as network speeds got faster, so you don't see them being used anymore.

A little while later (late 90s, early 2000s), plugin technologies became very popular, such as [Java Applets](#) and [Flash](#) — these allowed web developers to embed rich content into webpages such as videos and animations, which just weren't available through HTML alone. Embedding these technologies was achieved through elements like `<object>`, and the lesser-used `<embed>`, and they were very useful at the time. They have since fallen out of fashion due to many problems, including accessibility, security, file size, and more. These days major browsers have stopped supporting plugins such as Flash.

Finally, the `<iframe>` element appeared (along with other ways of embedding content, such as `<canvas>`, `<video>`, etc.) This provides a way to embed an entire web document inside another one, as if it were an `<img>` or other such element, and is used regularly today.

### Active learning - classic embedding uses

Youtube 和谷歌地图都可从通过 `<iframe>` 来分享内容。

具体例子看MDN

## 4.2 iframes in detail

`<iframe>` elements are designed to allow you to embed other web documents into the current document.

This is great for incorporating third-party content into your website

There are some serious **Security concerns** to consider with `<iframe>`s, but this doesn't mean that you shouldn't use them in your website. It just requires some knowledge and careful thinking.

```
<head>
  <style> iframe { border: none } </style>
</head>
<body>
  <iframe src="https://developer.mozilla.org/en-US/docs/Glossary"
          width="100%" height="500" allowfullscreen sandbox>
    <p>
      <a href="/en-US/docs/Glossary">
        Fallback link for browsers that don't support iframes
      </a>
    </p>
  </iframe>
</body>
```

### border: none

If used, the `<iframe>` is displayed without a surrounding border. Otherwise, by default, browsers display the `<iframe>` with a surrounding border (which is generally undesirable).

### allowfullscreen

If set, the `<iframe>` is able to be placed in fullscreen mode using the [Fullscreen API](#) (somewhat beyond the scope of this article.)

### src

This attribute, as with `<video>/<img>`, contains a path pointing to the URL of the document to be embedded.

### width and height

These attributes specify the width and height you want the iframe to be.

### Fallback content

In the same way as other similar elements like `<video>`, you can include fallback content between the opening and closing `<iframe></iframe>` tags that will appear if the browser doesn't support the `<iframe>`. In this case, we have included a link to the page instead. It is unlikely that you'll come across any browser that doesn't support `<iframe>`s these days.

### sandbox

This attribute, which works in slightly more modern browsers than the rest of the `<iframe>` features (e.g. IE 10 and above) requests heightened security settings; we'll say more about this in the next section.

**Note:** In order to improve speed, it's a good idea to set the iframe's `src` attribute with JavaScript after the main content is done with loading. This makes your page usable sooner and decreases your official page load time (an important SEO metric.)

## 4.2.1 Security Concerns





## 4.3 The <embed> and <object> elements

## 5. Adding vector graphics to the Web

### 6. Responsive images

## <4> HTML Tables

1. HTML Tables overview

2. HTML table basics

3. HTML Table advanced features and accessibility