

# 语义化

## 语义元素

语义元素清楚地向浏览器和开发者描述其意义。

非语义元素的例子：<div> 和 <span> - 无法提供关于其内容的信息。

语义元素的例子：<form>、<table> 以及 <img> - 清晰地定义其内容。

```
<header>
  <h1>标题</h1>
</header>
<main></main>
<footer></footer>
```

开发者和搜索引擎能很直观地读懂这段代码是由头部、主体部分和尾部三个部分组成。

## 新的语义元素

标签	描述
<article>	定义文章。
<aside>	定义页面内容以外的内容。
<details>	定义用户能够查看或隐藏的额外细节。
<figcaption>	定义 <figure> 元素的标题。
<figure>	规定自包含内容，比如图示、图表、照片、代码清单等。
<footer>	定义文档或节的页脚。
<header>	规定文档或节的页眉。
<main>	规定文档的主内容。
<mark>	定义重要的或强调的文本。
<nav>	定义导航链接。
<section>	定义文档中的节。
<summary>	定义 <details> 元素的可见标题。
<time>	定义日期/时间。

## 为什么要语义化

- 1.) 为了在没有CSS的情况下，页面也能呈现出很好地内容结构、代码结构；
- 2.) 语义化更具可读性，代码简洁明了，利于开发者阅读和维护，方便以后扩展；
- 3.) 有利于SEO（搜索引擎优化）：和搜索引擎建立良好沟通，有助于爬虫抓取更多的有效信息：爬虫依赖于标签来确定上下文和各个关键字的权重；
- 4.) 利于团队协作，可以减少差异化，降低出错几率，使工作有个统一标准；
- 5.) 方便其他设备解析（如屏幕阅读器、盲人阅读器、移动设备）以意义的方式来渲染网页。

## 注意

- 1.) 尽可少使用无语义的标签div和span；
- 2.) 在语义不明显时，既可以使用div或者p时，尽量用p，因为p在默认情况下有上下间距，对兼容特殊终端有利；
- 3.) 不要使用纯样式标签，如：b、font、u等，改用css设置。
- 4.) 需要强调的文本，可以包含在strong或者em标签中（浏览器预设样式，能用CSS指定就不用他们），strong默认样式是加粗（不要用b），em是斜体（不用i）；

5.)使用表格时，标题要用caption，表头用thead，主体部分用tbody包围，尾部用tfoot包围。表头和一般单元格要区分开，表头用th，单元格用td；

6.)表单域要用fieldset标签包起来，并用legend标签说明表单的用途；

7.)每个input标签对应的说明文本都需要使用label标签，并且通过为input设置id属性，在label标签中设置如<label for="someid">来让说明文本和相对应的input关联起来。