

## 7 The global structure of an HTML document

### Contents

1. [Introduction to the structure of an HTML document](#)
2. [HTML version information](#)
3. [The HTML element](#)
4. [The document head](#)
  1. [The HEAD element](#)
  2. [The TITLE element](#)
  3. [The title attribute](#)
  4. [Meta data](#)
    - [Specifying meta data](#)
    - [The META element](#)
    - [Meta data profiles](#)
5. [The document body](#)
  1. [The BODY element](#)
  2. [Element identifiers: the id and class attributes](#)
  3. [Block-level and inline elements](#)
  4. [Grouping elements: the DIV and SPAN elements](#)
  5. [Headings: The H1, H2, H3, H4, H5, H6 elements](#)
  6. [The ADDRESS element](#)

### 7.1 Introduction to the structure of an HTML document

An HTML 4 document is composed of three parts:

1. a line containing [HTML version information](#),
2. a declarative header section (delimited by the [HEAD](#) element),
3. a body, which contains the document's actual content. The body may be implemented by the [BODY](#) element or the [FRAMESET](#) element.

White space (spaces, newlines, tabs, and comments) may appear before or after each section. Sections 2 and 3 should be delimited by the [HTML](#) element.

Here's an example of a simple HTML document:

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01//EN"
  "http://www.w3.org/TR/html4/strict.dtd">
<HTML>
  <HEAD>
    <TITLE>My first HTML document</TITLE>
  </HEAD>
  <BODY>
    <P>Hello world!
  </BODY>
</HTML>
```

## 7.2 HTML version information

A valid HTML document declares what version of HTML is used in the document. The *document type declaration* names the document type definition (DTD) in use for the document (see [\[ISO8879\]](#)).

HTML 4.01 specifies three DTDs, so authors must include one of the following document type declarations in their documents. The DTDs vary in the elements they support.

- The [HTML 4.01 Strict DTD](#) includes all elements and attributes that have not been [deprecated](#) or do not appear in frameset documents. For documents that use this DTD, use this document type declaration:

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01//EN"
    "http://www.w3.org/TR/html4/strict.dtd">
```

- The [HTML 4.01 Transitional DTD](#) includes everything in the strict DTD plus deprecated elements and attributes (most of which concern visual presentation). For documents that use this DTD, use this document type declaration:

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN"
    "http://www.w3.org/TR/html4/loose.dtd">
```

- The [HTML 4.01 Frameset DTD](#) includes everything in the transitional DTD plus frames as well. For documents that use this DTD, use this document type declaration:

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Frameset//EN"
    "http://www.w3.org/TR/html4/frameset.dtd">
```

The URI in each document type declaration allows user agents to download the DTD and any [entity sets](#) that are needed. The following (relative) URIs refer to DTDs and entity sets for HTML 4:

- "[strict.dtd](#)" -- default strict DTD
- "[loose.dtd](#)" -- loose DTD
- "[frameset.dtd](#)" -- DTD for frameset documents
- "[HTMLlat1.ent](#)" -- Latin-1 entities
- "[HTMLsymbol.ent](#)" -- Symbol entities
- "[HTMLspecial.ent](#)" -- Special entities

The binding between public identifiers and files can be specified using a catalog file following the format recommended by the Oasis Open Consortium (see [\[OASISOPEN\]](#)). A [sample catalog file for HTML 4.01](#) is included at the beginning of the section on SGML reference information for HTML. The last two letters of the declaration indicate the language of the DTD. For HTML, this is always English ("EN").

**Note.** *As of the 24 December version of HTML 4.01, the HTML Working Group commits to the following policy:*

- *Any changes to future HTML 4 DTDs will not invalidate documents that conform to the DTDs of the present specification.* The HTML Working Group reserves the right to correct known bugs.
- *Software conforming to the DTDs of the present specification may ignore features of future HTML 4 DTDs that it does not recognize.*

*This means that in a document type declaration, authors may safely use a system identifier that refers to the latest version of an HTML 4 DTD. Authors may also choose to use a system identifier that refers to a specific (dated) version of an HTML 4 DTD when validation to that*

*particular DTD is required. W3C will make every effort to make archival documents indefinitely available at their original address in their original form.*

## 7.3 The HTML element

```
<!ENTITY % html.content "HEAD, BODY">

<!ELEMENT HTML O O (%html.content;) -- document root element -->
<!ATTLIST HTML
  %i18n; -- lang, dir --
>
```

**Start tag: optional, End tag: optional**

**Attribute definitions**

**version = [cdata \[CN\]](#)**

**Deprecated.** The value of this attribute specifies which HTML DTD version governs the current document. This attribute has been deprecated because it is redundant with [version information](#) provided by the document type declaration.

**Attributes defined elsewhere**

- [lang](#) ([language information](#)), [dir](#) ([text direction](#))

After document type declaration, the remainder of an HTML document is contained by the [HTML](#) element. Thus, a typical HTML document has this structure:

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01//EN"
"http://www.w3.org/TR/html4/strict.dtd">
<HTML>
...The head, body, etc. goes here...
</HTML>
```

## 7.4 The document head

### 7.4.1 The HEAD element

```
<!-- %head.misc; defined earlier on as "SCRIPT|STYLE|META|LINK|OBJECT" -->
<!ENTITY % head.content "TITLE & BASE?">

<!ELEMENT HEAD O O (%head.content;) +(%head.misc;) -- document head -->
<!ATTLIST HEAD
  %i18n; -- lang, dir --
  profile %URI; #IMPLIED -- named dictionary of meta info --
>
```

**Start tag: optional, End tag: optional**

**Attribute definitions**

**profile = [uri \[CT\]](#)**

This attribute specifies the location of one or more meta data profiles, separated by white space. For future extensions, user agents should consider the value to be a list even though this specification only considers the first URI to be significant. [Profiles](#) are discussed below in the section on [meta data](#).

### Attributes defined elsewhere

- `lang` ([language information](#)), `dir` ([text direction](#))

The [HEAD](#) element contains information about the current document, such as its title, keywords that may be useful to search engines, and other data that is not considered document content. User agents do not generally render elements that appear in the [HEAD](#) as content. They may, however, make information in the [HEAD](#) available to users through other mechanisms.

## 7.4.2 The `TITLE` element

```
<!-- The TITLE element is not considered part of the flow of text.
      It should be displayed, for example as the page header or
      window title. Exactly one title is required per document.
-->
<!ELEMENT TITLE - - (#PCDATA) -(%head.misc;.) -- document title -->
<!ATTLIST TITLE %i18n>
```

Start tag: **required**, End tag: **required**

### Attributes defined elsewhere

- `lang` ([language information](#)), `dir` ([text direction](#))

Every HTML document **must** have a [TITLE](#) element in the [HEAD](#) section.

Authors should use the [TITLE](#) element to identify the contents of a document. Since users often consult documents out of context, authors should provide context-rich titles. Thus, instead of a title such as "Introduction", which doesn't provide much contextual background, authors should supply a title such as "Introduction to Medieval Bee-Keeping" instead.

For reasons of accessibility, user agents must always make the content of the [TITLE](#) element available to users (including [TITLE](#) elements that occur in frames). The mechanism for doing so depends on the user agent (e.g., as a caption, spoken).

Titles may contain [character entities](#) (for accented characters, special characters, etc.), but may not contain other markup (including comments). Here is a sample document title:

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01//EN"
      "http://www.w3.org/TR/html4/strict.dtd">
<HTML>
<HEAD>
<TITLE>A study of population dynamics</TITLE>
... other head elements...
</HEAD>
<BODY>
... document body...
</BODY>
</HTML>
```

## 7.4.3 The `title` attribute

### Attribute definitions

`title` = [text](#) [CS]

This attribute offers advisory information about the element for which it is set.

Unlike the [TITLE](#) element, which provides information about an entire document and may only appear once, the [title](#) attribute may annotate any number of elements. Please consult an element's definition to verify that it supports this attribute.

Values of the [title](#) attribute may be rendered by user agents in a variety of ways. For instance, visual browsers frequently display the title as a "tool tip" (a short message that appears when the pointing device pauses over an object). Audio user agents may speak the title information in a similar context. For example, setting the attribute on a link allows user agents (visual and non-visual) to tell users about the nature of the linked resource:

```
...some text...
Here's a photo of
<A href="http://someplace.com/neatstuff.gif" title="Me scuba diving">
  me scuba diving last summer
</A>
...some more text...
```

The [title](#) attribute has an additional role when used with the [LINK](#) element to designate an [external style sheet](#). Please consult the section on [links and style sheets](#) for details.

***Note.** To improve the quality of speech synthesis for cases handled poorly by standard techniques, future versions of HTML may include an attribute for encoding phonemic and prosodic information.*

#### 7.4.4 Meta data

***Note.** The W3C Resource Description Framework (see [RDF10]) became a W3C Recommendation in February 1999. RDF allows authors to specify machine-readable metadata about HTML documents and other network-accessible resources.*

HTML lets authors specify meta data -- information about a document rather than document content -- in a variety of ways.

For example, to specify the author of a document, one may use the [META](#) element as follows:

```
<META name="Author" content="Dave Raggett">
```

The [META](#) element specifies a property (here "Author") and assigns a value to it (here "Dave Raggett").

This specification does not define a set of legal meta data properties. The meaning of a property and the set of legal values for that property should be defined in a reference lexicon called a [profile](#). For example, a profile designed to help search engines index documents might define properties such as "author", "copyright", "keywords", etc.

#### Specifying meta data

In general, specifying meta data involves two steps:

1. Declaring a property and a value for that property. This may be done in two ways:
  1. From within a document, via the [META](#) element.
  2. From outside a document, by linking to meta data via the [LINK](#) element (see the section on [link types](#)).
2. Referring to a [profile](#) where the property and its legal values are defined. To designate a profile, use the [profile](#) attribute of the [HEAD](#) element.

Note that since a profile is defined for the [HEAD](#) element, the same profile applies to all [META](#) and [LINK](#) elements in the document head.

User agents are not required to support meta data mechanisms. For those that choose to support meta data, this specification does not define how meta data should be interpreted.

## The [META](#) element

```
<!ELEMENT META - O EMPTY          -- generic metainformation -->
<!ATTLIST META
  %i18n;
  http-equiv NAME          #IMPLIED -- HTTP response header name --
  name       NAME          #IMPLIED -- metainformation name --
  content    CDATA        #REQUIRED -- associated information --
  scheme     CDATA        #IMPLIED -- select form of content --
>
```

Start tag: **required**, End tag: **forbidden**

### Attribute definitions

For the following attributes, the permitted values and their interpretation are [profile](#) dependent:

**name** = [name](#) [CS]

This attribute identifies a property name. This specification does not list legal values for this attribute.

**content** = [cdata](#) [CS]

This attribute specifies a property's value. This specification does not list legal values for this attribute.

**scheme** = [cdata](#) [CS]

This attribute names a scheme to be used to interpret the property's value (see the section on [profiles](#) for details).

**http-equiv** = [name](#) [CI]

This attribute may be used in place of the `name` attribute. HTTP servers use this attribute to gather information for HTTP response message headers.

### Attributes defined elsewhere

- [lang](#) ([language information](#)), [dir](#) ([text direction](#))

The [META](#) element can be used to identify properties of a document (e.g., author, expiration date, a list of key words, etc.) and assign values to those properties. This specification does not define a normative set of properties.

Each [META](#) element specifies a property/value pair. The `name` attribute identifies the property and the [content](#) attribute specifies the property's value.

For example, the following declaration sets a value for the Author property:

```
<META name="Author" content="Dave Raggett">
```

The [lang](#) attribute can be used with [META](#) to specify the language for the value of the [content](#) attribute. This enables speech synthesizers to apply language dependent pronunciation rules.

In this example, the author's name is declared to be French:

```
<META name="Author" lang="fr" content="Arnaud Le Hors">
```

**Note.** The [META](#) element is a generic mechanism for specifying meta data. However, some HTML elements and attributes already handle certain pieces of meta data and may be used by authors instead of [META](#) to specify those pieces: the [TITLE](#) element, the [ADDRESS](#) element, the [INS](#) and [DEL](#) elements, the [title](#) attribute, and the [cite](#) attribute.

**Note.** When a property specified by a [META](#) element takes a value that is a [URI](#), some authors prefer to specify the meta data via the [LINK](#) element. Thus, the following meta data declaration:

```
<META name="DC.identifier"
      content="http://www.ietf.org/rfc/rfc1866.txt">
```

might also be written:

```
<LINK rel="DC.identifier"
      type="text/plain"
      href="http://www.ietf.org/rfc/rfc1866.txt">
```

### [META](#) and HTTP headers

The [http-equiv](#) attribute can be used in place of the `name` attribute and has a special significance when documents are retrieved via the Hypertext Transfer Protocol (HTTP). HTTP servers may use the property name specified by the [http-equiv](#) attribute to create an [\[RFC822\]](#)-style header in the HTTP response. Please see the HTTP specification ([\[RFC2616\]](#)) for details on valid HTTP headers.

The following sample [META](#) declaration:

```
<META http-equiv="Expires" content="Tue, 20 Aug 1996 14:25:27 GMT">
```

will result in the HTTP header:

```
Expires: Tue, 20 Aug 1996 14:25:27 GMT
```

This can be used by caches to determine when to fetch a fresh copy of the associated document.

**Note.** Some user agents support the use of [META](#) to refresh the current page after a specified number of seconds, with the option of replacing it by a different URI. Authors should **not** use this technique to forward users to different pages, as this makes the page inaccessible to some users. Instead, automatic page forwarding should be done using server-side redirects.

### [META](#) and search engines

A common use for [META](#) is to specify keywords that a search engine may use to improve the quality of search results. When several [META](#) elements provide language-dependent information about a document, search engines may filter on the [lang](#) attribute to display search results using the language preferences of the user. For example,

```
<-- For speakers of US English -->
<META name="keywords" lang="en-us"
      content="vacation, Greece, sunshine">
<-- For speakers of British English -->
<META name="keywords" lang="en"
      content="holiday, Greece, sunshine">
<-- For speakers of French -->
<META name="keywords" lang="fr"
      content="vacances, Gr&egrave;ce, soleil">
```



The effectiveness of search engines can also be increased by using the [LINK](#) element to specify links to translations of the document in other languages, links to versions of the document in other media (e.g., PDF), and, when the document is part of a collection, links to an appropriate starting point for browsing the collection.

Further help is provided in the section on [helping search engines index your Web site](#).

### [META](#) and PICS

The Platform for Internet Content Selection (PICS, specified in [PICS](#)) is an infrastructure for associating labels (meta data) with Internet content. Originally designed to help parents and teachers control what children can access on the Internet, it also facilitates other uses for labels, including code signing, privacy, and intellectual property rights management.

This example illustrates how one can use a [META](#) declaration to include a PICS 1.1 label:

```
<HEAD>
<META http-equiv="PICS-Label" content='
(PICS-1.1 "http://www.gcf.org/v2.5"
  labels on "1994.11.05T08:15-0500"
  until "1995.12.31T23:59-0000"
  for "http://w3.org/PICS/Overview.html"
  ratings (suds 0.5 density 0 color/hue 1))
'>
<TITLE>... document title ...</TITLE>
</HEAD>
```

### [META](#) and default information

The [META](#) element may be used to specify the default information for a document in the following instances:

- The [default scripting language](#).
- The default [style sheet language](#).
- The [document character encoding](#).

The following example specifies the [character encoding](#) for a document as being ISO-8859-5

```
<META http-equiv="Content-Type" content="text/html; charset=ISO-8859-5">
```

### Meta data profiles

The [profile](#) attribute of the [HEAD](#) specifies the location of a meta data profile. The value of the [profile](#) attribute is a URI. User agents may use this URI in two ways:

- As a globally unique name. User agents may be able to recognize the name (without actually retrieving the profile) and perform some activity based on known conventions for that profile. For instance, search engines could provide an interface for searching through catalogs of HTML documents, where these documents all use the same profile for representing catalog entries.
- As a link. User agents may dereference the URI and perform some activity based on the actual definitions within the profile (e.g., authorize the usage of the profile within the current HTML document). This specification does not define formats for profiles.



This example refers to a hypothetical profile that defines useful properties for document indexing. The properties defined by this profile -- including "author", "copyright", "keywords", and "date" -- have their values set by subsequent [META](#) declarations.

```
<HEAD profile="http://www.acme.com/profiles/core">
  <TITLE>How to complete Memorandum cover sheets</TITLE>
  <META name="author" content="John Doe">
  <META name="copyright" content="© 1997 Acme Corp.">
  <META name="keywords" content="corporate, guidelines, cataloging">
  <META name="date" content="1994-11-06T08:49:37+00:00">
</HEAD>
```

*As this specification is being written, it is common practice to use the date formats described in [\[RFC2616\]](#), section 3.3. As these formats are relatively hard to process, we recommend that authors use the [\[ISO8601\]](#) date format. For more information, see the sections on the [INS](#) and [DEL](#) elements.*

The [scheme](#) attribute allows authors to provide user agents more context for the correct interpretation of meta data. At times, such additional information may be critical, as when meta data may be specified in different formats. For example, an author might specify a date in the (ambiguous) format "10-9-97"; does this mean 9 October 1997 or 10 September 1997? The [scheme](#) attribute value "Month-Day-Year" would disambiguate this date value.

At other times, the [scheme](#) attribute may provide helpful but non-critical information to user agents.

For example, the following [scheme](#) declaration may help a user agent determine that the value of the "identifier" property is an ISBN code number:

```
<META scheme="ISBN" name="identifier" content="0-8230-2355-9">
```

Values for the [scheme](#) attribute depend on the property name and the associated [profile](#).

***Note.** One sample profile is the Dublin Core (see [\[DCORE\]](#)). This profile defines a set of recommended properties for electronic bibliographic descriptions, and is intended to promote interoperability among disparate description models.*

## 7.5 The document body

### 7.5.1 The BODY element

```
<!ELEMENT BODY O O (%block;|SCRIPT)+ +(INS|DEL) -- document body -->
<!ATTLIST BODY
  %attrs;                -- %coreattrs, %i18n, %events --
  onload                 %Script;    #IMPLIED -- the document has been loaded --
  onunload               %Script;    #IMPLIED -- the document has been removed --
>
```

Start tag: **optional**, End tag: **optional**

Attribute definitions

background = [uri](#) [CT]

**Deprecated.** The value of this attribute is a URI that designates an image resource. The image generally tiles the background (for visual browsers).

text = [color](#) [CI]

**Deprecated.** This attribute sets the foreground color for text (for visual browsers).

link = [color](#) [CI]

**Deprecated.** This attribute sets the color of text marking unvisited hypertext links (for visual browsers).

vlink = **color [CI]**

**Deprecated.** This attribute sets the color of text marking visited hypertext links (for visual browsers).

alink = **color [CI]**

**Deprecated.** This attribute sets the color of text marking hypertext links when selected by the user (for visual browsers).

### *Attributes defined elsewhere*

- [id](#), [class](#) ([document-wide identifiers](#))
- [lang](#) ([language information](#)), [dir](#) ([text direction](#))
- [title](#) ([element title](#))
- [style](#) ([inline style information](#))
- [bgcolor](#) ([background color](#))
- [onload](#), [onunload](#) ([intrinsic events](#))
- [onclick](#), [ondblclick](#), [onmousedown](#), [onmouseup](#), [onmouseover](#), [onmousemove](#), [onmouseout](#), [onkeypress](#), [onkeydown](#), [onkeyup](#) ([intrinsic events](#))

The body of a document contains the document's content. The content may be presented by a user agent in a variety of ways. For example, for visual browsers, you can think of the body as a canvas where the content appears: text, images, colors, graphics, etc. For audio user agents, the same content may be spoken. Since [style sheets](#) are now the preferred way to specify a document's presentation, the presentational attributes of [BODY](#) have been [deprecated](#).

### DEPRECATED EXAMPLE:

The following HTML fragment illustrates the use of the [deprecated](#) attributes. It sets the background color of the canvas to white, the text foreground color to black, and the color of hyperlinks to red initially, fuchsia when activated, and maroon once visited.

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN"
"http://www.w3.org/TR/html4/loose.dtd">
<HTML>
<HEAD>
<TITLE>A study of population dynamics</TITLE>
</HEAD>
<BODY bgcolor="white" text="black"
link="red" alink="fuchsia" vlink="maroon">
... document body...
</BODY>
</HTML>
```

Using [style sheets](#), the same effect could be accomplished as follows:

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01//EN"
"http://www.w3.org/TR/html4/strict.dtd">
<HTML>
<HEAD>
<TITLE>A study of population dynamics</TITLE>
<STYLE type="text/css">
BODY { background: white; color: black }
A:link { color: red }
A:visited { color: maroon }
A:active { color: fuchsia }
</STYLE>
</HEAD>
```

```
<BODY>
... document body...
</BODY>
</HTML>
```

Using external (linked) style sheets gives you the flexibility to change the presentation without revising the source HTML document:

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01//EN"
"http://www.w3.org/TR/html4/strict.dtd">
<HTML>
<HEAD>
<TITLE>A study of population dynamics</TITLE>
<LINK rel="stylesheet" type="text/css" href="smartstyle.css">
</HEAD>
<BODY>
... document body...
</BODY>
</HTML>
```

**Framesets and HTML bodies.** Documents that contain framesets replace the [BODY](#) element by the [FRAMESET](#) element. Please consult the section on [frames](#) for more information.

## 7.5.2 Element identifiers: the `id` and `class` attributes

### Attribute definitions

`id` = [name](#) [CS]

This attribute assigns a name to an element. This name must be unique in a document.

`class` = [cdata-list](#) [CS]

This attribute assigns a class name or set of class names to an element. Any number of elements may be assigned the same class name or names. Multiple class names must be separated by white space characters.

The [id](#) attribute assigns a unique identifier to an element (which may be verified by an SGML parser). For example, the following paragraphs are distinguished by their [id](#) values:

```
<P id="myparagraph"> This is a uniquely named paragraph.</P>
<P id="yourparagraph"> This is also a uniquely named paragraph.</P>
```

The [id](#) attribute has several roles in HTML:

- As a [style sheet](#) selector.
- As a target [anchor](#) for hypertext links.
- As a means to reference a particular element from a [script](#).
- As the name of a declared [OBJECT](#) element.
- For general purpose processing by user agents (e.g. for identifying fields when extracting data from HTML pages into a database, translating HTML documents into other formats, etc.).

The [class](#) attribute, on the other hand, assigns one or more class names to an element; the element may be said to belong to these classes. A class name may be shared by several element instances. The [class](#) attribute has several roles in HTML:

- As a [style sheet](#) selector (when an author wishes to assign style information to a set of elements).
- For general purpose processing by user agents.

In the following example, the [SPAN](#) element is used in conjunction with the [id](#) and [class](#) attributes to markup document messages. Messages appear in both English and French versions.

```
<!-- English messages -->
<P><SPAN id="msg1" class="info" lang="en">Variable declared twice</SPAN>
<P><SPAN id="msg2" class="warning" lang="en">Undeclared variable</SPAN>
<P><SPAN id="msg3" class="error" lang="en">Bad syntax for variable name</SPAN>

<!-- French messages -->
<P><SPAN id="msg1" class="info" lang="fr">Variable d&eacute;clar&eacute;e deux fois</SPAN>
<P><SPAN id="msg2" class="warning" lang="fr">Variable ind&eacute;finie</SPAN>
<P><SPAN id="msg3" class="error" lang="fr">Erreur de syntaxe pour variable</SPAN>
```

The following CSS style rules would tell visual user agents to display informational messages in green, warning messages in yellow, and error messages in red:

```
SPAN.info { color: green }
SPAN.warning { color: yellow }
SPAN.error { color: red }
```

Note that the French "msg1" and the English "msg1" may not appear in the same document since they share the same [id](#) value. Authors may make further use of the [id](#) attribute to refine the presentation of individual messages, make them target anchors, etc.

Almost every HTML element may be assigned identifier and class information.

Suppose, for example, that we are writing a document about a programming language. The document is to include a number of preformatted examples. We use the [PRE](#) element to format the examples. We also assign a background color (green) to all instances of the [PRE](#) element belonging to the class "example".

```
<HEAD>
<TITLE>... document title ...</TITLE>
<STYLE type="text/css">
PRE.example { background : green }
</STYLE>
</HEAD>
<BODY>
<PRE class="example" id="example-1">
...example code here...
</PRE>
</BODY>
```

By setting the [id](#) attribute for this example, we can (1) create a hyperlink to it and (2) override class style information with instance style information.

**Note.** The [id](#) attribute shares the same name space as the [name](#) attribute when used for anchor names. Please consult the section on [anchors with id](#) for more information.

### 7.5.3 Block-level and inline elements

Certain HTML elements that may appear in [BODY](#) are said to be "block-level" while others are "inline" (also known as "text level"). The distinction is founded on several notions:

#### Content model

Generally, block-level elements may contain inline elements and other block-level elements. Generally, inline elements may contain only data and other inline elements. Inherent in this structural distinction is the idea that block elements create "larger" structures than inline elements.

## Formatting

By default, block-level elements are formatted differently than inline elements. Generally, block-level elements begin on new lines, inline elements do not. For information about white space, line breaks, and block formatting, please consult the section on [text](#).

## Directionality

For technical reasons involving the [\[UNICODE\]](#) bidirectional text algorithm, block-level and inline elements differ in how they inherit directionality information. For details, see the section on [inheritance of text direction](#).

[Style sheets](#) provide the means to specify the rendering of arbitrary elements, including whether an element is rendered as block or inline. In some cases, such as an inline style for list elements, this may be appropriate, but generally speaking, authors are discouraged from overriding the conventional interpretation of HTML elements in this way.

The alteration of the traditional presentation idioms for block level and inline elements also has an impact on the bidirectional text algorithm. See the section on [the effect of style sheets on bidirectionality](#) for more information.

## 7.5.4 Grouping elements: the DIV and SPAN elements

```
<!ELEMENT DIV - - (%flow;)*           -- generic language/style container -->
<!ATTLIST DIV
  %attrs;                               -- %coreattrs, %i18n, %events --
>
<!ELEMENT SPAN - - (%inline;)*         -- generic language/style container -->
<!ATTLIST SPAN
  %attrs;                               -- %coreattrs, %i18n, %events --
>
```

Start tag: **required**, End tag: **required**

Attributes defined elsewhere

- [id](#), [class](#) ([document-wide identifiers](#))
- [lang](#) ([language information](#)), [dir](#) ([text direction](#))
- [title](#) ([element title](#))
- [style](#) ([inline style information](#))
- [align](#) ([alignment](#))
- [onclick](#), [ondblclick](#), [onmousedown](#), [onmouseup](#), [onmouseover](#), [onmousemove](#), [onmouseout](#), [onkeypress](#), [onkeydown](#), [onkeyup](#) ([intrinsic events](#))

The [DIV](#) and [SPAN](#) elements, in conjunction with the [id](#) and [class](#) attributes, offer a generic mechanism for adding structure to documents. These elements define content to be inline ([SPAN](#)) or block-level ([DIV](#)) but impose no other presentational idioms on the content. Thus, authors may use these elements in conjunction with [style sheets](#), the [lang](#) attribute, etc., to tailor HTML to their own needs and tastes.

Suppose, for example, that we wanted to generate an HTML document based on a database of client information. Since HTML does not include elements that identify objects such as "client", "telephone number", "email address", etc., we use [DIV](#) and [SPAN](#) to achieve the desired structural and presentational effects. We might use the [TABLE](#) element as follows to structure the information:

```
<!-- Example of data from the client database: -->
<!-- Name: Stephane Boyera, Tel: (212) 555-1212, Email: sb@foo.org -->

<DIV id="client-boyera" class="client">
  <P><SPAN class="client-title">Client information:</SPAN>
```

```

<TABLE class="client-data">
<TR><TH>Last name:<TD>Boyer</TR>
<TR><TH>First name:<TD>Stephane</TR>
<TR><TH>Tel:<TD>(212) 555-1212</TR>
<TR><TH>Email:<TD>sb@foo.org</TR>
</TABLE>
</DIV>

<DIV id="client-lafon" class="client">
<P><SPAN class="client-title">Client information:</SPAN>
<TABLE class="client-data">
<TR><TH>Last name:<TD>Lafon</TR>
<TR><TH>First name:<TD>Yves</TR>
<TR><TH>Tel:<TD>(617) 555-1212</TR>
<TR><TH>Email:<TD>yves@coucou.com</TR>
</TABLE>
</DIV>

```

Later, we may easily add style sheet declarations to fine tune the presentation of these database entries.

For another example of usage, please consult the example in the section on [the class and id attributes](#).

Visual user agents generally place a line break before and after [DIV](#) elements, for instance:

```
<P>aaaaaaaa<DIV>bbbbbbbb</DIV><DIV>cccc<P>cccc</DIV>
```

which is typically rendered as:

```

aaaaaaaa
bbbbbbbb
cccc

cccc

```

## 7.5.5 Headings: The H1, H2, H3, H4, H5, H6 elements

```

<!ENTITY % heading "H1|H2|H3|H4|H5|H6">
<!--
  There are six levels of headings from H1 (the most important)
  to H6 (the least important).
-->

<!ELEMENT (%heading;) - - (%inline;)* -- heading -->
<!ATTLIST (%heading;)
  %attrs;                -- %coreattrs, %i18n, %events --
>

```

**Start tag: required, End tag: required**

*Attributes defined elsewhere*

- [id](#), [class](#) ([document-wide identifiers](#))
- [lang](#) ([language information](#)), [dir](#) ([text direction](#))
- [title](#) ([element title](#))
- [style](#) ([inline style information](#))
- [align](#) ([alignment](#))
- [onclick](#), [ondblclick](#), [onmousedown](#), [onmouseup](#), [onmouseover](#), [onmousemove](#), [onmouseout](#), [onkeypress](#), [onkeydown](#), [onkeyup](#) ([intrinsic events](#))

A heading element briefly describes the topic of the section it introduces. Heading information may be used by user agents, for example, to construct a table of contents for a document automatically.

There are six levels of headings in HTML with [H1](#) as the most important and [H6](#) as the least. Visual browsers usually render more important headings in larger fonts than less important ones.

The following example shows how to use the [DIV](#) element to associate a heading with the document section that follows it. Doing so allows you to define a style for the section (color the background, set the font, etc.) with style sheets.

```
<DIV class="section" id="forest-elephants" >
<H1>Forest elephants</H1>
<P>In this section, we discuss the lesser known forest elephants.
...this section continues...
<DIV class="subsection" id="forest-habitat" >
<H2>Habitat</H2>
<P>Forest elephants do not live in trees but among them.
...this subsection continues...
</DIV>
</DIV>
```

This structure may be decorated with style information such as:

```
<HEAD>
<TITLE>... document title ...</TITLE>
<STYLE type="text/css">
DIV.section { text-align: justify; font-size: 12pt}
DIV.subsection { text-indent: 2em }
H1 { font-style: italic; color: green }
H2 { color: green }
</STYLE>
</HEAD>
```

### Numbered sections and references

*HTML does not itself cause section numbers to be generated from headings. This facility may be offered by user agents, however. Soon, style sheet languages such as CSS will allow authors to control the generation of section numbers (handy for forward references in printed documents, as in "See section 7.2").*

*Some people consider skipping heading levels to be bad practice. They accept H1 H2 H1 while they do not accept H1 H3 H1 since the heading level H2 is skipped.*

## 7.5.6 The ADDRESS element

```
<!ELEMENT ADDRESS - - (%inline;)* -- information on author -->
<!ATTLIST ADDRESS
  %attrs;                -- %coreattrs, %i18n, %events --
  >
```

Start tag: **required**, End tag: **required**

Attributes defined elsewhere

- [id](#), [class](#) (document-wide identifiers)
- [lang](#) (language information), [dir](#) (text direction)
- [title](#) (element title)
- [style](#) (inline style information)
- [onclick](#), [ondblclick](#), [onmousedown](#), [onmouseup](#), [onmouseover](#), [onmousemove](#), [onmouseout](#), [onkeypress](#), [onkeydown](#), [onkeyup](#) (intrinsic events)



The [ADDRESS](#) element may be used by authors to supply contact information for a document or a major part of a document such as a form. This element often appears at the beginning or end of a document.

For example, a page at the W3C Web site related to HTML might include the following contact information:

```
<ADDRESS>
<A href="../../People/Raggett/">Dave Raggett</A>,
<A href="../../People/Arnaud/">Arnaud Le Hors</A>,
contact persons for the <A href="Activity">W3C HTML Activity</A><BR>
$Date: 2018/03/20 02:36:52 $
</ADDRESS>
```

---

[previous](#) [next](#) [contents](#) [elements](#) [attributes](#) [index](#)