

HTML: The Markup Language

[« global attributes](#)

[references »](#)

9. Data types (common microsyntaxes)

This section describes data types (microsyntaxes) that are referenced by attribute descriptions in the [HTML elements](#), and [Global attributes](#) sections.

String

For the purposes of this document, a *string* is defined as any mixture of [text](#) and [character references](#).

The [Attributes](#) section of this document describes additional restrictions on strings in attribute values — in particular, restrictions for the following cases:

- [unquoted attribute values](#)
- [single-quoted attribute values](#)
- [double-quoted attribute values](#)

Set of comma-separated strings

Zero or more [strings](#) that are themselves each zero or more characters, each optionally with leading and/or trailing [space characters](#), and each separated from the next by a single " ," (comma) character. Each string itself **must** not begin or end with any [space characters](#), and each string itself **must** not contain any " ," (comma) characters.

Token

A [string](#) that does not contain any [space characters](#).

Set of space-separated tokens

A [space-separated](#) set of zero or more [token](#) instances.

Unordered set of unique space-separated tokens

A [set of space-separated tokens](#) in which none of the tokens are duplicated.

Ordered set of unique space-separated tokens

A [set of space-separated tokens](#) in which none of the tokens are duplicated, but in which the order of the tokens is meaningful.

jump

Browsing-context name

Any [string](#), with the following restrictions:

- [must](#) not start with a "_" character
- [must](#) be at least one character long

Browsing-context name or keyword

Any [string](#) that is either of the following:

- a [browsing-context name](#)
- any [case-insensitive match](#) for one of the following literal strings: "_blank", "_self", "_parent", or "_top".

ID

Any [string](#), with the following restrictions:

- [must](#) be at least one character long
- [must](#) not contain any [space characters](#)

Note: Previous versions of HTML placed greater restrictions on the content of ID values (for example, they did not permit ID values to begin with a number).

ID reference

A valid ID reference to an element of type *type* is a string that exactly matches the value of the `id` attribute of an element of type *type* anywhere in the document.

List of ID references

An [unordered set of unique space-separated tokens](#), each of which is an [ID reference](#).

Name

Any [string](#), with the following restrictions:

- [must](#) be at least one character long
- [must](#) not contain any [space characters](#)

Hash-name reference

A valid hash-name reference to an element of type *type* is a string that starts with a "#" character, followed by a string which exactly matches the value of the `name` attribute of an element of type *type* anywhere in the document. **jump**

Integer

One or more characters in the range 0–9, optionally prefixed with a "-" character.

Positive integer

Any [non-negative integer](#), with the following restriction:

- [must](#) be greater than zero

Non-negative integer

One or more characters in the range 0–9.

Floating-point number

A floating-point number consists of the following parts, in exactly the following order:

1. Optionally, the first character may be a "-" character.
2. One or more characters in the range "0–9".
3. Optionally, the following parts, in exactly the following order:
 1. a "." character
 2. one or more characters in the range "0–9"
4. Optionally, the following parts, in exactly the following order:
 1. a "e" character or "E" character
 2. optionally, a "-" character or "+" character
 3. One or more characters in the range "0–9".

Positive floating-point number

A [non-negative floating-point number](#), with the following restriction:

- [must](#) be greater than zero

Non-negative floating-point number

A [floating-point number](#), with the following restriction:

- the first character may not be a "-" character

Date and time

A valid *date-time* as defined in [\[RFC 3339\]](#), with these additional qualifications:

- the literal letters `T` and `Z` in the date/time syntax [must](#) always be uppercase
- the *date-fullyear* production is instead defined as four or more digits representing a number greater than 0

jump

Examples:

```
1990-12-31T23:59:60Z
1996-12-19T16:39:57-08:00
```

Date or time

Any one of the following:

- [a date](#)
- [a time](#)
- [a date and time](#)

Date

A valid *full-date* as defined in [\[RFC 3339\]](#), with the additional qualification that the year component is four or more digits representing a number greater than 0.

Example:

```
1996-12-19
```

URL

A valid *IRI reference* as defined in [\[RFC 3987\]](#).

Note: The empty string is a valid *IRI reference*, so the empty string is allowed anywhere this reference lists the “URL” datatype as being allowed.

Example:

```
http://example.org/hello
```

URL potentially surrounded by spaces

A [URL](#), optionally with leading and/or trailing [space characters](#).

Note: The empty string is a valid [URL](#), so the empty string is allowed anywhere this reference lists the “URL potentially surrounded by spaces” datatype as being allowed.

Non-empty URL potentially surrounded by spaces

A [URL](#) that is not the empty string, optionally with leading and/or trailing [sp](#) [jump](#)

[characters](#).

Absolute URL potentially surrounded by spaces

A valid *IRI* as defined in [\[RFC 3987\]](#), optionally with leading and/or trailing [space characters](#).

Examples:

```
/hello
#canvas
http://example.org/
```

Sizes

An [unordered set of unique space-separated tokens](#), each of which [must](#) be one of the following:

- the literal string "any"
- two valid [non-negative integers](#) that do not have a leading "0" character and that are separated by a single "x" character.

MIME type

A string that identifies a valid *MIME media type* as defined in [\[RFC 2046\]](#).

Character encoding name

A [case-insensitive match](#) for any *character set name* for which the IANA [\[Character Sets\]](#) registry has a *Name* or *Alias* field labeled as “preferred MIME name”; or, if none of the *Alias* fields are so labeled, a [case-insensitive match](#) for a *Name* field in the registry.

Meta-charset string

The following parts, in exactly the following order:

1. The literal string "text/html;".
2. Optionally, one or more [space characters](#).
3. The literal string "charset=".
4. One of the following:
 - For documents in the [HTML syntax](#): A [character encoding name](#).
 - For documents in the [XML syntax](#): Any [case-insensitive match](#) for the string "UTF-8".

Refresh value

jump

Any one of the following:

- A [non-negative integer](#).
- The string consisting of the following parts, in exactly the following order:
 1. A [non-negative integer](#).
 2. A ";" character.
 3. One or more [space characters](#).
 4. A [case-insensitive match](#) for the string "url=".
 5. A [URL](#).

Default-style name

A [string](#).

Media-query list

A valid *media query list* as defined in [\[Media Queries\]](#).

Language tag

A valid *language tag* as defined in [\[BCP 47\]](#).

List of key labels

An [ordered set of unique space-separated tokens](#), each of which [must](#) be exactly one Unicode code point in length.

Dropzone value

An [unordered set of unique space-separated tokens](#), each of which is a [case-insensitive match](#) for one of the following:

copy

Indicates that dropping an accepted item on the element will result in a copy of the dragged data.

move

Indicates that dropping an accepted item on the element will result in the dragged data being moved to the new location.

link

Indicates that dropping an accepted item on the element will result in a link to the original data.

Any [string](#) with three characters or more, beginning with the literal string "string:".

Indicates that *Plain Unicode string* items, of the type

jump

indicated by the part of of the keyword after the "string:" string, can be dropped on this element.

Any [string](#) with three characters or more, beginning with the literal string "file:". Indicates that *File* items, of the type indicated by the part of of the keyword after the "file:" string, can be dropped on this element.

The value must not have more than one of the three tokens "copy", "move", or "link". If none are specified, the element represents a *copy* dropzone.

Zero

The literal string "0".

Functionbody

Any JavaScript code matching the [FunctionBody](#) production [\[ECMA 262\]](#).

Rectangle coordinates

A comma-separated list of four [integers](#), in exactly the following order:

1. an [integer](#) representing the distance in CSS pixels from the left edge of the image to the left side of the rectangle
2. an [integer](#) representing the distance in CSS pixels from the top edge of the image to the top side of the rectangle
3. an [integer](#), greater than the value of the first integer in this list, representing the distance in CSS pixels from the left edge of the image to the right side of the rectangle
4. an [integer](#), greater than the value of the second integer in this list, representing the distance in CSS pixels from the top edge of the image to the bottom side of the rectangle

Circle coordinates

A comma-separated list of three numbers, in exactly the following order:

1. an [integer](#) representing the distance in CSS pixels from the left edge of the image to the center of the circle
2. an [integer](#) representing the distance in CSS pixels from the top edge of the image to the center of the circle
3. a [non-negative integer](#), representing the radius of the circle, in CSS pixels

Polygon coordinates

A comma-separated list of at least six [integers](#), with the total number of integers in the list being even (that is, six or eight or ten numbers, and so on). Each [jump](#)

of integers represents a coordinate, in CSS pixels, given as the distances from, respectively, the left and the top of the image; all the coordinates together represent the points of the polygon, in order.

Sandbox “allow” keywords list

An [unordered set of unique space-separated tokens](#), each of which is a [case-insensitive match](#) for one of the following literal strings:

- "allow-forms"
- "allow-scripts"
- "allow-top-navigation"
- "allow-same-origin"

Note: Because an [unordered set of unique space-separated tokens](#) can contain zero tokens, this datatype also allows the following:

- the empty string
- any string consisting only of [space characters](#)

List of MIME types

A [set of comma-separated strings](#), each of which is a valid [MIME type](#), with no parameters.

List of character-encoding names

An [ordered set of unique space-separated tokens](#), each of which [must](#) be a valid [character encoding name](#) that specifies an [ASCII-compatible character encoding](#).

Pattern

A regular expression that [must](#) match the JavaScript *Pattern* production as specified in [\[ECMA 262\]](#).

Local date and time

The following parts, in exactly the following order:

1. A [date](#).
2. The literal string "T".
3. A [time](#).

Example:

```
1985-04-12T23:20:50.52
```

jump


```
1996-12-19T16:39:57
```

Date

A valid *full-date* as defined in [\[RFC 3339\]](#), with the additional qualification that the year component is four or more digits representing a number greater than 0.

Example:

```
1996-12-19
```

Month

The following parts, in exactly the following order:

1. Four or more digits representing a number greater than 0.
2. The literal string "-".
3. Two digits, representing the month *month*, in the range $1 \leq month, \leq 12$.

Example:

```
1996-12
```

Week

The following parts, in exactly the following order:

1. Four or more digits representing year *year*, where $year > 0$.
2. The literal string "-w".
3. Two digits, representing the week *week*, in the range $1 \leq week \leq maxweek$, where *maxweek* is either 52 or 53, depending on the particular year.

Example:

```
1996-w16
```

Time

A valid *partial-time* as defined in [\[RFC 3339\]](#).

Examples:

```
23:20:50.52  
17:39:57
```

jump

E-mail address

Any string that matches the following [\[ABNF\]](#) production:

```
1*( atext / "." ) "@" ldh-str 1*( "." ldh-str )
```

...where *atext* is as defined in [\[RFC 5322\]](#), and *ldh-str* is as defined in [\[RFC 1034\]](#).

That is, any string which matches the following regular expression:

```
/^[a-zA-Z0-9.!#$%&'*/+=?^_`{|}~-]+@[a-zA-Z0-9-]+(?:\.[a-zA-Z0-9-]+)*$/
```

Examples:

```
foo-bar.baz@example.com
```

List of e-mail addresses

A [set of comma-separated strings](#), each of which is a valid [email address](#).

Simple color

A string exactly seven characters long, consisting of the following parts, in exactly the following order:

1. A "#" character.
2. Six characters in the range 0–9, a–f, and A–F.

Note: Color keywords (for example, strings such as “red” or “green”) are not allowed.

String without line breaks

Any string that contains no line feed (U+000A, “LF”) or carriage return (U+000D, “CR”) characters.

Non-empty string

Any string that is not empty.

[« global attributes](#)

[references »](#)

[jump](#)