

HTML entity

HTML named or numeric character entity.

```
&(#\d+|#x[0-9a-fA-F]+|[a-zA-Z]+);
```

Token-by-token breakdown

Token	Meaning
&	literal text "&"
(	start of a capturing group
#	literal text "#"
\d+	one or more: digit (0–9)
	OR — try the alternative
#x	literal text "#x"
[0-9a-fA-F]+	one or more: any of: digits, a–f, A–F
	OR — try the alternative
[a-zA-Z]+	one or more: any of: lowercase letters, uppercase letters
)	end of group
;	literal text ";"

Quick usage in 7 languages

JavaScript	<pre>const re = /&(#\d+ #x[0-9a-fA-F]+ [a-zA-Z]+)/; re.test(value);</pre>
Python	<pre>import re re.match(r"&(#\d+ #x[0-9a-fA-F]+ [a-zA-Z]+)", value)</pre>
Java	<pre>Pattern.compile("&(#\d+ #x[0-9a-fA-F]+ [a-zA-Z]+)").matcher(value).matches();</pre>
C# / .NET	<pre>Regex.IsMatch(value, @"&(#\d+ #x[0-9a-fA-F]+ [a-zA-Z]+)");</pre>
Go	<pre>regexp.MustCompile(`&(#\d+ #x[0-9a-fA-F]+ [a-zA-Z]+)`).MatchString(value)</pre>
Ruby	<pre>/&(#\d+ #x[0-9a-fA-F]+ [a-zA-Z]+)/.match?(value)</pre>
PHP	<pre>preg_match('~&(#\d+ #x[0-9a-fA-F]+ [a-zA-Z]+);~', \$value);</pre>

Common pitfalls

- **Not anchored.** Without `^` and `$` this can match a substring anywhere in the input — add anchors if you need the whole value to conform.
- **ASCII letters only.** `[a-zA-Z]` excludes accented and non-Latin letters (é, ü, ß, ñ, and non-Latin scripts). For international input use Unicode properties like `\p{L}` with the `u` flag.
- **Escape it correctly per language.** In Java and JavaScript strings each backslash must be doubled (`\\d`); in Python, Go, and C# use raw/verbatim strings so the backslashes survive.
- **Validate beyond format.** Matching the format doesn't guarantee the value is real. Confirm the html entity against a source of truth (database, API, or checksum) where it matters.