

Validating Form Fields

Why Validate Content?

The simplest way of validation is to mark a form element as mandatory. The validation then checks whether the element has been filled with content as required.

A second validation level is to validate the field contents themselves, for example to prevent incorrect or incomplete entries. For example, an input field for (German) postal codes should consist of exactly five digits or an e-mail address should contain an @ sign. A validation of entries can be done using regular expressions.

The following form elements therefore contain the configuration option **Validation Expression**:

- [Input Field](#)
- [Input Field \(Mobile\)](#)

In this field you can enter the desired regular expressions. When validating the data entered in the input field, the system then checks whether the entries comply with the required rules.

Time of the Validation

Responsive Forms	Mobile Forms
<ul style="list-style-type: none">• The form must contain a button with activated option Validate Fields.• The validation of all fields in this form is performed as soon as this button is pressed.	<ul style="list-style-type: none">• Mobile form fields are validated directly when the element is processed for the first time. Exceptions: Checkbox and App Link.• If the navigation element contains the Next button, the validation of all fields is triggered when this button is pressed.

Special Characters in Regular Expressions

The following characters have special meaning in regular expressions:

Characters	Meaning
^	The circumflex stands for the beginning of the input.
\$	The dollar symbol marks the end of an entry.
*	The asterisk represents any number. The preceding character may not occur at all or may occur any number of times.
.	The dot stands for any single character.
+	The plus sign means: The preceding element must occur at least once .
?	The question mark asks if the preceding element does not occur at all or occurs exactly once .
	The vertical line ("pipe") means: Either - Or. Example: When checking for a b, input a would be as valid as input b, but input c would not.
[]	The square brackets enclose a character selection. The expression is valid if one of the characters occurs in the input. A hyphen can be used to define a range within the parentheses. Example: [1234] checks the same as [1-4]. Valid entries are 2 pears as well as 3 apples.
[^]	The combination of square brackets and circumflex stands for a negated selection. Example: [^123] means that all entries that do not contain 1, 2 or 3 are valid.
{ }	The curly braces help to query frequencies. Examples: {n} The preceding element occurs exactly n times . {n,m} The preceding character occurs n to m times . {0,n} The preceding element occurs zero to n times . {n,} The preceding character occurs at least n times .

On this Page:

- [Why Validate Content?](#)
 - [Time of the Validation](#)
- [Special Characters in Regular Expressions](#)
 - [\\$](#)
- [Helpful Regular Expressions](#)

Related Pages:

- [Form Elements](#)
 - [Input Field](#)
 - [Button](#)

Related Documentation:

- [MOBILE](#) (German)
 - [Eingabefelder](#)
 - [Navigation](#)


\	<p>The backslash helps if you want to find characters that cannot be entered using the keyboard.</p> <p>Examples:</p> <ul style="list-style-type: none"> \0 finds the null character \s finds all white spaces \t finds the tab character \n finds the line break character
---	--



For a complete list of the special characters used in regular expressions, see <https://developer.mozilla.org>.

Helpful Regular Expressions

We have compiled a list of frequently used regular expressions for you here:

Regular Expression	Meaning
\s	The <code>s</code> stands for white space. <code>\s</code> finds all Unicode spaces, i.e. spaces, breaks, tabs and typographical special spaces.
\s*	The field may not contain any or any number of spaces.
^. {2,10}\$	Limits the number of characters that can be entered to a minimum of 2 and a maximum of 10.
.+@.+	Checks whether the input contains an @ character.
[0-9]	Only numbers with the digits zero to nine are accepted as input.
^[0-9]+\$	Only numbers are accepted as input and the field must not remain empty.
^[0-9]{5}\$	Only digits are accepted as input and exactly five digits must be entered.
^[0-9/-]*\$	The field may only contain (any number of) digits, slash, space and minus sign, or it may be blank. This expression can be used, for example, to validate the entry of phone numbers.
^[0-9.,]*\$	The field may only contain (any number of) digits, periods and commas or may be empty. This expression can be used, for example, to check the entry of prices or numbers with thousands separators.
^[A-Z]*\$	Only (any number of) capital letters may be entered (without umlauts).
^[a-z]*\$	Only (any number of) lower case letters may be entered (without umlauts).
^[a-zA-Z]*\$	Any number of lowercase and uppercase letters may be entered (without umlauts).
^[a-zA-Z]*\$	You can enter any number of lowercase and uppercase letters including spaces (without umlauts).
^[a-zA-Z-öäü ÖÄÜß]*\$	Any number of upper and lower case letters including umlauts and ß are allowed.
a-zA-Z0-9öäü ÖÄÜ \$%&]*\$	Any number of small and capital letters, the numbers 0 to 9 as well as the umlauts and the special characters \$, %, & are permitted. Suitable, for example, for checking a password that may only contain the special characters specified in the regular expression.
	<div>  <p>A limitation of the password to a certain number of characters can also be checked.</p> <p>Example: A password which may contain upper and lower case letters, numbers and the special characters \$, %, & but no umlauts and must consist of at least 8, but not more than 16 characters.</p> <p><code>^[a-zA-Z0-9\$%&]{8,16}\$</code></p> </div>
^\D*\$	Any number of entries other than decimal-numeric characters (letters, umlauts, special characters, but no numbers) are allowed.

<code>^[a-zA-Z0-9]{0,5}\$</code>	Only a maximum of 5 letters and numbers are accepted. No spaces, umlauts or special characters.
<code>^[a-zA-Z0-9._-]+@[a-zA-Z0-9._-]+\.[a-z]{2,}\$</code>	Checks whether a valid e-mail address has been entered.



Further helpful hints on regular expressions can be found at <https://wiki.selfhtml.org>. A tester for self-created regular expressions can be found at <http://www.regexpal.com>.