

# HTML атрибут: max (макс)

Атрибут `max` определяет максимальное значение, которое является приемлемым и допустимым для входных данных, содержащих атрибут. Если значение `value` элемента больше этого значения, элемент не проходит [проверку \(validation\)](#). Это значение должно быть больше или равно значению атрибута `min`. Если атрибут `max` присутствует, но не указан или недействителен, значение не применяется. Если атрибут `max` действителен и непустое значение превышает максимально допустимое для атрибута, проверка ограничения предотвратит отправку формы.

Действителен для числовых типов ввода, включая типы [date](#), [Month](#), [Week](#), [Time](#), [DateTime-local](#), [Number](#) и [Range](#), а также элементов `<progress>` и `<meter>`. Атрибут `max` представляет собой число, указывающее наиболее положительное значение, которое элемент управления формы считается действительным.

Если значение превышает максимально допустимое значение, `validityState.rangeOverflow` будет `true` истина), и элемент управления будет соответствовать псевдоклассам `:out-of-range` и `:invalid`.

## Синтаксис

Синтаксис для **max** значений по вводу **type**

Тип ввода	Синтаксис	Пример
<a href="#">date</a>	yyyy-mm-dd	<code>&lt;input type=«date» max=«2019-12-25» step=«1»&gt;</code>
<a href="#">Month</a>	yyyy-mm	<code>&lt;input type=«month» max=«2019-12» step=«12»&gt;</code>
<a href="#">Week</a>	yyyy-W##	<code>&lt;input type=«week» max=«2019-W23» step=«»&gt;</code>
<a href="#">Time</a>	hh:mm	<code>&lt;input type=«time» max=«17:00» step=«900»&gt;</code>
<a href="#">DateTime-local</a>	yyyy-mm-ddThh:mm	<code>&lt;input type=«datetime-local» max=«2019-12-25T23:59»&gt;</code>
<a href="#">Number</a>	<code>&lt;number&gt;</code>	<code>&lt;input type=«number» min=«0» step=«5» max=«100»&gt;</code>
<a href="#">Range</a>	<code>&lt;number&gt;</code>	<code>&lt;input type=«range» min=«60» step=«5» max=«100»&gt;</code>



**Примечание.** Если данные, введенные пользователем, не соответствуют установленному максимальному значению, значение считается недействительным при проверке ограничений и будет соответствовать псевдоклассам `:out-of-range` и `:invalid`

См. раздел [Проверка на стороне клиента](#) и [rangeOverflow](#) дополнительную информацию.

Для `<progress>` элемента атрибут `max` описывает, сколько работы `<progress>` требует задача, указанная элементом. Если он присутствует, он должен иметь значение больше нуля и быть допустимым числом с плавающей запятой. Для `<meter>` элемента атрибут `max` определяет верхнюю числовую границу измеряемого диапазона. Оно должно быть больше минимального значения (`min` атрибута), если оно указано. В обоих случаях, если он опущен, значение по умолчанию равно 1.

## Синтаксис **max** значений для **других элементов**

Тип ввода	Синтаксис	Пример
<code>&lt;progress&gt;</code>	<code>&lt;number&gt;</code>	<code>&lt;progress id=«file» max=«100» value=«70»&gt; 70% &lt;/progress&gt;</code>
<code>&lt;meter&gt;</code>	<code>&lt;number&gt;</code>	<code>&lt;meter id=«fuel» min=«0» max=«100» low=«33» high=«66» optimum=«80» value=«40»&gt; at 40/100&lt;/meter&gt;</code>

## Проблемы доступности

Предоставьте инструкции, которые помогут пользователям понять, как заполнять форму и использовать отдельные элементы управления формой. Укажите все необходимые и необязательные входные данные, форматы данных и другую соответствующую информацию. При использовании `max` атрибута убедитесь, что это максимальное требование понятно пользователю. Предоставления инструкций внутри `<label>` может быть достаточно. Если вы предоставляете инструкции вне этикеток, что обеспечивает более гибкое позиционирование и дизайн, рассмотрите возможность использования `aria-labelledby` или `aria-describedby`.

/

При использовании в качестве атрибута `<output>` атрибут `for` имеет значение, которое представляет собой разделенный пробелами список значений `id` элементов, которые используются для создания выходных данных.

В следующем примере форма предоставляет ползунок, значение которого может находиться в диапазоне от 0и 100, а также `<input>` элемент, в который можно ввести второе число. Два числа складываются, и результат отображается в элементе `<output>` каждый раз, когда изменяется значение любого из элементов управления.

## Примеры

Использование атрибута для элементов `<label>` и `<output>`.

From: <http://synoinstall-gqctx9n8ug2b3eq1.direct.quickconnect.to/> - worldwide open-source software

Permanent link: <http://synoinstall-gqctx9n8ug2b3eq1.direct.quickconnect.to/doku.php?id=software:development:web:docs:web:html:attributes:max&rev=1693479572>

Last update: 2023/08/31 13:59

