

Animation Triggers

Editor's Draft, 21 May 2026



▼ More details about this document

This version:

<https://drafts.csswg.org/animation-triggers-1/>

Latest published version:

<https://www.w3.org/TR/animation-triggers-1/>

Feedback:

[CSSWG Issues Repository](#)

[Inline In Spec](#)

Editors:

[Robert Flack](#) (Google)

[David Awogbemila](#) (Google)

[Yehonatan Daniv](#) (Wix.com)

Suggest an Edit for this Spec:

[GitHub Editor](#)

Copyright © 2026 [World Wide Web Consortium](#). W3C[®] [liability](#), [trademark](#) and [permissive document license](#) rules apply.

Abstract

Defines CSS properties and an API for creating mechanisms that affect animation playback, based on DOM events resulting from user interaction, or on entry/exit of ranges on animation timelines.

[CSS](#) is a language for describing the rendering of structured documents (such as HTML and XML) on screen, on paper, etc.

Status of this document

This is a public copy of the editors' draft. It is provided for discussion only and may change at any moment. Its publication here does not imply endorsement of its contents by W3C. Don't cite this document other than as work in progress.

Please send feedback by [filing issues in GitHub](#) (preferred), including the spec code “animation-triggers” in the title, like this: “[animation-triggers] ...*summary of comment*...”. All issues and comments are [archived](#). Alternately, feedback can be sent to the ([archived](#)) public mailing list www-style@w3.org.

This document is governed by the [18 August 2025 W3C Process Document](#).

Table of Contents

1	Introduction
1.1	Relationship to other specifications
1.2	Value Definitions
2	Triggers
2.1	Trigger Name Scoping: the ‘ trigger-scope ’ property
2.2	Timeline Triggers
2.2.1	Naming the Trigger: the ‘ timeline-trigger-name ’ property
2.2.2	Linking a Timeline: the ‘ timeline-trigger-source ’ property
2.2.3	The Activation Range: the ‘ timeline-trigger-activation-range ’ property
2.2.4	The Active Range: the ‘ timeline-trigger-active-range ’ property
2.2.5	The ‘ timeline-trigger ’ Shorthand
2.3	Event Triggers
2.3.1	Naming the Trigger: the ‘ event-trigger-name ’ property
2.3.2	Linking an Event: the ‘ event-trigger-source ’ property
2.3.3	The ‘ event-trigger ’ Shorthand
2.3.4	The ‘ animation-trigger ’ property
3	Privacy Considerations
4	Security Considerations
	Conformance
	Document conventions
	Conformance classes
	Partial implementations
	Implementations of Unstable and Proprietary Features
	Non-experimental implementations

Index

Terms defined by this specification

Terms defined by reference

References

Normative References

Property Index

Issues Index

§ 1. Introduction

This specification defines mechanisms for affecting an animation’s playback based on various user interactions invoking a specified trigger. By specifying a trigger for an animation, that animation becomes a *triggered animation*, making its playback start delayed until that trigger occurs. These triggers can be certain [Events](#) defined in [\[DOM\]](#), or timeline based, such as [view progress timelines](#). [\[SCROLL-ANIMATIONS-1\]](#)

This module provides both an imperative API building on the Web Animations API as well as a declarative API building on CSS Animations. [\[CSS-ANIMATIONS-2\]](#) [\[WEB-ANIMATIONS-1\]](#)

§ 1.1. Relationship to other specifications

Web Animations [\[WEB-ANIMATIONS-1\]](#) defines an abstract conceptual model for animations on the Web platform, with elements of the model including [animations](#) and [timelines](#), and associated programming interfaces. This specification extends the Web Animations model by defining [triggered animation](#) and allowing to perform [playback control](#) functions on them using triggers.

This specification introduces both programming interfaces for interacting with these concepts, as well as CSS properties that apply these concepts to CSS Animations [\[CSS-ANIMATIONS-2\]](#). To the extent the behavior of these CSS properties is described in terms of the programming interfaces, [User Agents](#) that do not support scripting may still conform to this specification by implementing the CSS features to behave as if the underlying programming interfaces were in place.

This specification uses the timeline and ranges concept introduced in the Scroll-Driven Animations module [\[SCROLL-ANIMATIONS-1\]](#) for specifying progress-based timelines and animations.

Like most operations in CSS besides [selector](#) matching, features in this specification operate over the

[flattened element tree](#).

§ 1.2. Value Definitions

This specification follows the [CSS property definition conventions](#) from [CSS2] using the [value definition syntax](#) from [CSS-VALUES-3]. Value types not defined in this specification are defined in CSS Values & Units [CSS-VALUES-3]. Combination with other CSS modules may expand the definitions of these value types.

In addition to the property-specific values listed in their definitions, all properties defined in this specification also accept the [CSS-wide keywords](#) as their property value. For readability they have not been repeated explicitly.

§ 2. Triggers

While CSS animations are, by default, automatically run as soon as the appropriate [‘animation’](#) values have been set on an element, the [‘animation-trigger’](#) property allows the animation’s start to be delayed until an appropriate trigger occurs, and even paused, restarted, or reset by triggers (making it a [triggered animation](#)).

Currently, two types of *triggers* are defined:

- [timeline triggers](#), managed by the [‘timeline-trigger’](#) properties, which allow animations to be triggered by entering or leaving certain timeline ranges. (Usually, [view progress timelines](#), so an animation can be started when an element comes on-screen.)
- [event triggers](#), managed by the [‘event-trigger’](#) properties, which allow animations to be triggered by certain user-interaction events, such as clicking an element or pressing certain keys.

A [trigger](#) is *defined* on some specific triggering element. All triggers have a *trigger name*, and the specific type of trigger dictates how and when it’s activated. A trigger can define multiple "types" of activation. (For example, [timeline triggers](#) can do different things on entry and exit.)

A [trigger](#) is *used* on potentially any element, creating a *trigger instance* on the element. (For example, [‘animation-trigger’](#) associates a [trigger instance](#) with a specific animation on the element.) The trigger-using element specifies what actions to take when the trigger activates.

NOTE: This design for [triggers](#) and [trigger instances](#), and the way they’re associated with [triggered animations](#) and [<animation-action>](#)s, is intentionally somewhat generic, intended to support using triggers for *other* purposes in the future. For now, though, triggered animations are the only user of this feature.

If a single element attempts to define multiple [triggers](#) of different types with the same [trigger name](#), it only exposes one of the triggers, with [event triggers](#) winning over [timeline triggers](#).

NOTE: This order is completely arbitrary (based on alphabetic order of the concept name), as this is just an error case.

§ 2.1. Trigger Name Scoping: the ‘[trigger-scope](#)’ property

[Trigger names](#) are global by default, usable by other elements regardless of their position. (Though they are [tree-scoped names](#), so have interactions with [shadow roots](#)). If multiple elements define [triggers](#) with the same trigger name, the trigger defined by the later element in [tree order](#) is used.

The ‘[trigger-scope](#)’ property can limit the scope of a name to a subtree of the document, so elements outside won’t see the chosen [trigger name](#), and elements inside will only see the version of the trigger name defined inside the scope.

<i>Name:</i>	‘trigger-scope’
<i>Value:</i>	none all <dashed-ident> #
<i>Initial:</i>	none
<i>Applies to:</i>	all elements
<i>Inherited:</i>	no
<i>Percentages:</i>	n/a
<i>Computed value:</i>	as specified
<i>Canonical order:</i>	per grammar

Animation not animatable
type:

This property scopes [trigger names](#) to the subtree of the matching element. Its values are:

‘none’

No changes in [trigger name](#) scope.

‘all’

Specifies that all [trigger names](#) defined by this element or its descendants—whose scope is not already limited by a descendant using [‘trigger-scope’](#)—to be in scope only for this element’s [flat tree](#) descendants; and limits descendants to only match trigger names to [triggers](#) within this subtree.

This value only affects [trigger names](#) in the same tree scope, as if it were a [strictly matched tree-scoped name](#). (That is, [‘trigger-scope: all’](#) acts identically to [‘trigger-scope: --foo, --bar, ...’](#), listing all relevant trigger names.)

‘<dashed-ident>’

Specifies that a matching [trigger name](#) defined by this element or its descendants—whose scope is not already limited by a descendant using [‘trigger-scope’](#)—to be in scope only for this element’s [flat tree](#) descendants; and limits descendants to only match these trigger names to [triggers](#) within this subtree.

The [<dashed-ident>](#) represents a [strictly matched tree-scoped name](#), i.e. it can only match against [trigger names](#) in the same shadow tree.[\[CSS-SCOPING-1\]](#)

§ 2.2. Timeline Triggers

A **timeline trigger** is a [trigger](#) which is activated when some [timeline](#) enters the trigger’s **activation range**, or leaves the trigger’s **active range**. It is defined on an element with the [‘timeline-trigger’](#) shorthand property, or its longhands.

A [timeline trigger](#) has a binary **trigger state** associated with it; it is initially "inactive". While it’s "inactive", the associated [timeline](#) entering (or starting in) the trigger’s [activation range](#) performs an associated **entry action** and switches the [trigger state](#) to "active"; while it’s "active", the associated timeline *leaving* the trigger’s [active range](#) performs an associated **exit action** and switches the trigger state to "inactive".

NOTE: By default, the [active range](#) is the same as the [activation range](#); even when manually specified, the active range is always a *superset* of the activation range. The two ranges allow, for example, an [‘animation-trigger’](#) to start an animation when an element is scrolled close to the center of the screen (using a [view progress timeline](#) with a relatively small window as the activation range), but not stop it until the element is fully off-screen (using [‘cover’](#) as the active range).

A [timeline trigger](#) can have one or two actions associated with it when used as a trigger on an element (such as by [‘animation-trigger’](#)). If two are specified, the first is the trigger’s [entry action](#) and the second is the trigger’s [exit action](#); if only one is specified, the first is the trigger’s entry action and its exit action is to do nothing.

An element can define multiple [timeline triggers](#), using the same [timeline](#) (potentially with different ranges) or different ones. The set of [‘timeline-trigger’](#) longhands form a [coordinating list property group](#), with [‘timeline-trigger-name’](#) as the [coordinating list base property](#), and each item in the [coordinated value list](#) defining the properties of a single timeline trigger.

§ 2.2.1. Naming the Trigger: the [‘timeline-trigger-name’](#) property

<i>Name:</i>	‘timeline-trigger-name’
--------------	--

<i>Value:</i>	none <dashed-ident>#
---------------	--

<i>Initial:</i>	none
-----------------	------

<i>Applies to:</i>	all elements
--------------------	------------------------------

<i>Inherited:</i>	no
-------------------	----

<i>Percentages:</i>	N/A
---------------------	-----

<i>Computed value:</i>	as specified
------------------------	--------------

<i>Canonical order:</i>	per grammar
-------------------------	-------------

<i>Animation type:</i>	not animatable
------------------------	----------------

If ‘[none](#)’ is specified, the element does not define any [timeline triggers](#).

If the same [<dashed-ident>](#) appears multiple times in the list, only the last one defines a [timeline trigger](#); the preceding ones have no effect.

§ 2.2.2. Linking a Timeline: the ‘[timeline-trigger-source](#)’ property

<i>Name:</i>	‘timeline-trigger-source’
<i>Value:</i>	<single-animation-timeline>#
<i>Initial:</i>	auto
<i>Applies to:</i>	all elements
<i>Inherited:</i>	no
<i>Percentages:</i>	N/A
<i>Computed value:</i>	list, each item either the keyword ‘ none ’, the keyword ‘ auto ’, a case-sensitive css identifier , a computed ‘ scroll() ’ function, or a computed ‘ view() ’ function
<i>Canonical order:</i>	per grammar
<i>Animation type:</i>	not animatable

The ‘[timeline-trigger-source](#)’ property specifies the [timeline trigger](#)’s associated [timeline](#). Values have the same meaning as those of ‘[animation-timeline](#)’, except that ‘[none](#)’ instead causes the corresponding entry in the [coordinated value list](#) to not define a timeline trigger.

§ 2.2.3. The Activation Range: the ‘[timeline-trigger-activation-range](#)’ property

<i>Name:</i>	‘timeline-trigger-activation-range’
--------------	--

<i><u>Value:</u></i>	[<u><'timeline-trigger-activation-range-start'></u> <u><'timeline-trigger-activation-range-end'></u> ?]#
<i><u>Initial:</u></i>	see individual properties
<i><u>Applies to:</u></i>	see individual properties
<i><u>Inherited:</u></i>	see individual properties
<i><u>Percentages:</u></i>	see individual properties
<i><u>Computed value:</u></i>	see individual properties
<i><u>Animation type:</u></i>	see individual properties
<i><u>Canonical order:</u></i>	per grammar

The `'timeline-trigger-activation-range'` property is a [shorthand property](#) that sets `'timeline-trigger-activation-range-start'` and `'timeline-trigger-activation-range-end'` together in a single declaration. It has the same syntax as the `'animation-range'` property.

The behavior of `'timeline-trigger-activation-range'` is defined in [Web Animations 2 § 3.6.5 Animation Trigger Ranges](#).

<i><u>Name:</u></i>	<code>'timeline-trigger-activation-range-start'</code>, <code>'timeline-trigger-activation-range-end'</code>
<i><u>Value:</u></i>	[normal <u><length-percentage></u> <u><timeline-range-name></u> <u><length-percentage></u> ?]#
<i><u>Initial:</u></i>	normal
<i><u>Applies to:</u></i>	all elements
<i><u>Inherited:</u></i>	no
<i><u>Percentages:</u></i>	relative to the specified named timeline range if one was specified, else to the entire timeline

<u>Computed value:</u>	list, each item either the keyword ‘ normal ’ or a timeline range and progress percentage
<u>Canonical order:</u>	per grammar
<u>Animation type:</u>	not animatable

The ‘[timeline-trigger-activation-range-start](#)’ and ‘[timeline-trigger-activation-range-end](#)’ properties specify the [timeline trigger](#)’s associated [activation range](#). Values have the same meaning as ‘[animation-range-start](#)’ and ‘[animation-range-end](#)’.

§ 2.2.4. The Active Range: the ‘[timeline-trigger-active-range](#)’ property

<u>Name:</u>	‘ timeline-trigger-active-range ’
<u>Value:</u>	[< ‘timeline-trigger-active-range-start’ > < ‘timeline-trigger-active-range-end’ >?]#
<u>Initial:</u>	see individual properties
<u>Applies to:</u>	see individual properties
<u>Inherited:</u>	see individual properties
<u>Percentages:</u>	see individual properties
<u>Computed value:</u>	see individual properties
<u>Animation type:</u>	see individual properties
<u>Canonical order:</u>	per grammar

The ‘[timeline-trigger-active-range](#)’ property is a [shorthand property](#) that sets ‘[timeline-trigger-active-range-start](#)’ and ‘[timeline-trigger-active-range-end](#)’ together in a single

declaration. It has the same syntax as the [‘animation-range’](#) property.

The behavior of [‘timeline-trigger-active-range’](#) is defined in [Web Animations 2 § 3.6.5 Animation Trigger Ranges](#).

<i>Name:</i>	‘timeline-trigger-active-range-start’, ‘timeline-trigger-active-range-end’
<i>Value:</i>	[auto normal <length-percentage> <timeline-range-name> <length-percentage>?]#
<i>Initial:</i>	auto
<i>Applies to:</i>	all elements
<i>Inherited:</i>	no
<i>Percentages:</i>	relative to the specified named timeline range if one was specified, else to the entire timeline
<i>Computed value:</i>	list, each item either the keyword ‘normal’ or a timeline range and progress percentage
<i>Canonical order:</i>	per grammar
<i>Animation type:</i>	not animatable

The [‘timeline-trigger-active-range-start’](#) and [‘timeline-trigger-active-range-end’](#) properties specify the [timeline trigger](#)’s associated [active range](#). Values have the same meaning as [‘animation-range-start’](#) and [‘animation-range-end’](#), with the following addition:

‘auto’

The start (for [‘timeline-trigger-active-range-start’](#)) or end (for [‘timeline-trigger-active-range-end’](#)) is equal to the start/end of the [timeline trigger](#)’s [activation range](#).

§ 2.2.5. The [‘timeline-trigger’](#) Shorthand

Name: ***‘timeline-trigger’***

Value: none | [<‘timeline-trigger-name’> <‘timeline-trigger-source’> <‘timeline-trigger-activation-range’> [/ <‘timeline-trigger-active-range’>]?]#

Initial: see individual properties

Applies to: see individual properties

Inherited: see individual properties

Percentages: see individual properties

Computed value: see individual properties

Animation type: see individual properties

Canonical order: per grammar

The ***‘timeline-trigger’ shorthand property*** sets all of ***‘timeline-trigger-name’***, ***‘timeline-trigger-source’***, ***‘timeline-trigger-activation-range’***, and optionally ***‘timeline-trigger-active-range’*** at once.

A value of ***‘none’*** is equivalent to ***‘none none normal’***.

NOTE: Due to significant potential ambiguities in the syntax (***‘timeline-trigger-name’*** vs ***‘timeline’*** names in ***‘timeline-trigger-source’***; ***activation ranges*** vs ***active ranges***), this shorthand’s values must be given in the specified order, rather than being settable in any order as is more common.

§ 2.3. Event Triggers

An ***event trigger*** is a ***trigger*** which is activated when certain ***Events*** are fired at the element. It is defined on an element with the ***‘event-trigger’*** shorthand property, or its longhands.

An ***event trigger*** can be defined as either stateless or stateful:

- If stateless, it has a single set of ***enter events*** that activate it.

- If stateful, it has two sets of events, its [enter events](#) and another set of *exit events*.

[Event triggers](#) are activated when one of its associated [Events](#) are fired on the page with the trigger-defining element as its [target](#). If it's stateful, it has a binary *trigger state* associated with it, initially "inactive": while "inactive", it only activates when the defining element receives one of its [enter events](#), performing an associated *enter action* and switching its [trigger state](#) to "active"; while "active", it only deactivates when it receives one of its [exit events](#), performing an associated *exit action* and switching its trigger state back to "inactive".

A stateless [event trigger](#) must be given exactly one action for its [trigger instance](#). A stateful one can be given one or two: the first is its [enter action](#), and the second, if provided, is its [exit action](#); if the second is not provided, the exit action is to do nothing.

NOTE: A stateful and stateless [event trigger](#) act differently even if you only assign a single action; a single-action stateful event trigger will effectively "turn off" until it receives one of its [exit events](#), ignoring any of the [enter events](#) after the first, while a stateless one will repeatedly trigger for every enter event.

An element can define multiple [event triggers](#), using the same [Events](#) or different ones. The set of [‘event-trigger’](#) longhands form a [coordinating list property group](#), with [‘event-trigger-name’](#) as the [coordinating list base property](#), and each item in the [coordinated value list](#) defining the properties of a single event trigger.

ISSUE 1 The proposal I drew this text from specified that it only cares if the element is the *target* of the event. We probably want to allow for bubbling and/or capturing, possibly as an opt in/out.

§ 2.3.1. Naming the Trigger: the [‘event-trigger-name’](#) property

<i>Name:</i>	‘event-trigger-name’
--------------	--------------------------------------

<i>Value:</i>	none <dashed-ident>#
---------------	--

<i>Initial:</i>	none
-----------------	------

<i>Applies to:</i>	all elements
--------------------	------------------------------

<i>Inherited:</i>	no
-------------------	----

<u>Percentages:</u>	N/A
<u>Computed value:</u>	as specified
<u>Canonical order:</u>	per grammar
<u>Animation type:</u>	not animatable

If ‘none’ is specified, the element does not define any [event triggers](#).

If the same [<dashed-ident>](#) appears multiple times in the list, only the last one defines an [event trigger](#); the preceding ones have no effect.

§ 2.3.2. Linking an Event: the ‘[event-trigger-source](#)’ property

<u>Name:</u>	‘ event-trigger-source ’
<u>Value:</u>	[none <event-trigger-event>+ [/ <event-trigger-event>+]?]#
<u>Initial:</u>	none
<u>Applies to:</u>	all elements
<u>Inherited:</u>	no
<u>Percentages:</u>	N/A
<u>Computed value:</u>	as specified
<u>Canonical order:</u>	per grammar
<u>Animation type:</u>	not animatable

The ‘[event-trigger-source](#)’ property specifies what event or events activate the [event trigger](#). Its values

are:

‘none’

The corresponding entry in the [coordinated value list](#) does not define a trigger.

<event-trigger-event>+ [/ <event-trigger-event>+]?

Defines what event(s) the [event trigger](#) responds to.

If a ‘/’ is used in the value, the [event trigger](#) is stateful; the set of events before the ‘/’ are the event trigger’s [enter events](#), while those after the ‘/’ are the [exit events](#). (The same events can occur in both sets.)

Otherwise, the [event trigger](#) is stateless, and the provided events are its [enter events](#).

<event-trigger-event> = activate | interest | click | touch | dblclick | keypress(<

ISSUE 2 Figure out the full set of events we want to handle.

1

§ 2.3.3. The **‘event-trigger’** Shorthand

<i>Name:</i>	<u>‘event-trigger’</u>
<i>Value:</i>	none [<u><‘event-trigger-name’></u> <u><‘event-trigger-source’></u>]#
<i>Initial:</i>	none
<i>Applies to:</i>	all elements
<i>Inherited:</i>	no
<i>Percentages:</i>	N/A
<i>Computed value:</i>	as specified
<i>Animation type:</i>	not animatable
<i>Canonical order:</i>	per grammar

The [‘event-trigger’ shorthand property](#) sets both [‘event-trigger-name’](#) and [‘event-trigger-source’](#) at once.

A value of **‘none’** is equivalent to [‘none none’](#).

§ 2.3.4. The [‘animation-trigger’](#) property

<i>Name:</i>	‘animation-trigger’
<i>Value:</i>	[none [<dashed-ident> <animation-action>+]+]#
<i>Initial:</i>	none
<i>Applies to:</i>	all elements
<i>Inherited:</i>	no
<i>Percentages:</i>	N/A
<i>Computed value:</i>	as specified
<i>Canonical order:</i>	per grammar
<i>Animation type:</i>	not animatable

The [‘animation-trigger’](#) property specifies whether the animation is a [triggered animation](#), and if it is, what trigger it responds to and what actions it takes in response. [‘animation-trigger’](#) is a [reset-only sub-property](#) of the [‘animation’](#) shorthand. Its values are:

‘none’

The corresponding animation is not a [triggered animation](#).

‘[[<dashed-ident>](#) [<animation-action>+](#)]+’

The corresponding animation is a [triggered animation](#), responding to the triggers named by each [<dashed-ident>](#), and responding by taking the action named by the corresponding [<animation-action>](#). (See § 2.1 [Trigger Name Scoping: the trigger-scope property](#) for how [<dashed-ident>](#)s are resolved to [triggers](#).)

How many [<animation-action>](#)s a trigger accepts, and what exactly activates them, is determined by the type of the trigger. (Event triggers take one and possibly an optional second, depending on whether they’re stateless or stateful; timeline triggers take one and optionally a second.) Specifying the wrong number of actions (too many or too few) is valid syntactically, but causes the trigger to have no effect.

If multiple triggers occur simultaneously, they take effect in the order specified.

If the same [<dashed-ident>](#) is specified multiple times, all but the last have no effect.

The possible ‘[<animation-action>](#)’ values, and what effect they have in each animation state:

If there is an "effect", it happens regardless of the current state, before the state-specific action

Keyword	Extra Effect	initial	playing	paused	finished
‘none’	—	—	—	—	—
‘play’	—	play()	—	play()	play()
‘play-once’	—	play()	—	play()	—
‘play-forwards’	set playback rate to positive	play()	—	play()	play()
‘play-backwards’	set playback rate to negative	play()	—	play()	play()
‘pause’	—	—	pause()	—	—
‘reset’	set progress to 0	—	pause()	pause()	pause()
‘replay’	set progress to 0	play()	—	play()	play()

§ 3. Privacy Considerations

There are no known privacy impacts of the features in this specification.

§ 4. Security Considerations

There are no known security impacts of the features in this specification.

§ Conformance

§ Document conventions

Conformance requirements are expressed with a combination of descriptive assertions and RFC 2119 terminology. The key words “MUST”, “MUST NOT”, “REQUIRED”, “SHALL”, “SHALL NOT”, “SHOULD”, “SHOULD NOT”, “RECOMMENDED”, “MAY”, and “OPTIONAL” in the normative parts of this document are to be interpreted as described in RFC 2119. However, for readability, these words do not appear in all uppercase letters in this specification.

All of the text of this specification is normative except sections explicitly marked as non-normative, examples, and notes. [\[RFC2119\]](#)

Examples in this specification are introduced with the words “for example” or are set apart from the normative text with `class="example"`, like this:

EXAMPLE 1

This is an example of an informative example.

Informative notes begin with the word “Note” and are set apart from the normative text with `class="note"`, like this:

Note, this is an informative note.

Advisements are normative sections styled to evoke special attention and are set apart from other normative text with `<strong class="advisement">`, like this:

UAs MUST provide an accessible alternative.

▼ TESTS

Tests relating to the content of this specification may be documented in “Tests” blocks like this one. Any such block is non-normative.

§ Conformance classes

Conformance to this specification is defined for three conformance classes:

style sheet

A [CSS style sheet](#).

renderer

A [UA](#) that interprets the semantics of a style sheet and renders documents that use them.

authoring tool

A [UA](#) that writes a style sheet.

A style sheet is conformant to this specification if all of its statements that use syntax defined in this module are valid according to the generic CSS grammar and the individual grammars of each feature defined in this module.

A renderer is conformant to this specification if, in addition to interpreting the style sheet as defined by the appropriate specifications, it supports all the features defined by this specification by parsing them correctly and rendering the document accordingly. However, the inability of a UA to correctly render a document due to limitations of the device does not make the UA non-conformant. (For example, a UA is not required to render color on a monochrome monitor.)

An authoring tool is conformant to this specification if it writes style sheets that are syntactically correct according to the generic CSS grammar and the individual grammars of each feature in this module, and meet all other conformance requirements of style sheets as described in this module.

§ Partial implementations

So that authors can exploit the forward-compatible parsing rules to assign fallback values, CSS renderers **must** treat as invalid (and [ignore as appropriate](#)) any at-rules, properties, property values, keywords, and other syntactic constructs for which they have no usable level of support. In particular, user agents **must not** selectively ignore unsupported component values and honor supported values in a single multi-value property declaration: if any value is considered invalid (as unsupported values must be), CSS requires that the entire declaration be ignored.

§ Implementations of Unstable and Proprietary Features

To avoid clashes with future stable CSS features, the CSSWG recommends [following best practices](#) for the implementation of [unstable](#) features and [proprietary extensions](#) to CSS.

§ Non-experimental implementations

Once a specification reaches the Candidate Recommendation stage, non-experimental implementations are possible, and implementors should release an unprefixed implementation of any CR-level feature they can demonstrate to be correctly implemented according to spec.

To establish and maintain the interoperability of CSS across implementations, the CSS Working Group requests that non-experimental CSS renderers submit an implementation report (and, if necessary, the testcases used for that implementation report) to the W3C before releasing an unprefixed implementation of any CSS features. Testcases submitted to W3C are subject to review and correction by the CSS Working Group.

Further information on submitting testcases and implementation reports can be found from on the CSS Working Group’s website at <http://www.w3.org/Style/CSS/Test/>. Questions should be directed to the public-css-testsuite@w3.org mailing list.

§ Index

§ Terms defined by this specification

[activation range](#), in § 2.2

[active range](#), in § 2.2

[all](#), in § 2.1

[<animation-action>](#), in § 2.3.4

[animation-trigger](#), in § 2.3.4

[auto](#), in § 2.2.4

[<dashed-ident>](#), in § 2.1

[\[<dashed-ident> <animation-action>+ \]+](#), in § 2.3.4

[enter action](#), in § 2.3

[enter event](#), in § 2.3

[entry action](#), in § 2.2

[event trigger](#), in § 2.3

[event-trigger](#), in § 2.3.3

[<event-trigger-event>](#), in § 2.3.2

[<event-trigger-event>+ \[/ <event-trigger-event>+ \]?](#), in § 2.3.2

[event-trigger-name](#), in § 2.3.1

[event-trigger-source](#), in § 2.3.2

exit action

[dfn for event trigger](#), in § 2.3

[dfn for timeline trigger](#), in § 2.2

[exit event](#), in § 2.3

none

[value for <animation-action>](#), in § 2.3.4

[value for animation-trigger](#), in § 2.3.4

[value for event-trigger](#), in § 2.3.3

[value for event-trigger-source](#), in § 2.3.2

[value for timeline-trigger](#), in § 2.2.5

[value for trigger-scope](#), in § 2.1

[pause](#), in § 2.3.4[play](#), in § 2.3.4[play-backwards](#), in § 2.3.4[play-forwards](#), in § 2.3.4[play-once](#), in § 2.3.4[replay](#), in § 2.3.4[reset](#), in § 2.3.4[timeline trigger](#), in § 2.2[timeline-trigger](#), in § 2.2.5[timeline-trigger-activation-range](#), in § 2.2.3[timeline-trigger-activation-range-end](#), in § 2.2.3[timeline-trigger-activation-range-start](#), in § 2.2.3[timeline-trigger-active-range](#), in § 2.2.4[timeline-trigger-active-range-end](#), in § 2.2.4[timeline-trigger-active-range-start](#), in § 2.2.4[timeline-trigger-name](#), in § 2.2.1[timeline-trigger-source](#), in § 2.2.2[trigger](#), in § 2[triggered animation](#), in § 1[trigger instance](#), in § 2[trigger name](#), in § 2[trigger-scope](#), in § 2.1

trigger state

[dfn for event trigger](#), in § 2.3[dfn for timeline trigger](#), in § 2.2

§ Terms defined by reference

[] defines the following terms:

playback control

[CSS-ANIMATIONS-1] defines the following terms:

animation

[CSS-ANIMATIONS-2] defines the following terms:

<single-animation-timeline>

animation-timeline

[CSS-CASCADE-5] defines the following terms:

reset-only sub-property

shorthand property

[CSS-SHADOW-1] defines the following terms:

flat tree

flattened element tree

strictly matched

tree-scoped name

[CSS-VALUES-4] defines the following terms:

+
<dashed-ident>
<length-percentage>
<string>
?
coordinated value list
coordinating list base property
coordinating list property group
CSS identifier
CSS-wide keywords
|

[DOM] defines the following terms:

Event
shadow root
target
tree order

[INFRA] defines the following terms:

user agent

[SCROLL-ANIMATIONS-1] defines the following terms:

<timeline-range-name>
animation-range
animation-range-end
animation-range-start
cover
named timeline range
scroll()
view progress timelines
view()

[SELECTORS-4] defines the following terms:

selector

[WEB-ANIMATIONS-1] defines the following terms:

animation
pause()
play()
timeline

§ References

§ Normative References

[CSS-ANIMATIONS-1]

David Baron; et al. *CSS Animations Level 1*. URL: <https://drafts.csswg.org/css-animations/>

[CSS-ANIMATIONS-2]

David Baron; Brian Birtles. *CSS Animations Level 2*. URL: <https://drafts.csswg.org/css-animations-2/>

[CSS-CASCADE-5]

Elika Etemad; Miriam Suzanne; Tab Atkins Jr.. *CSS Cascading and Inheritance Level 5*. URL: <https://drafts.csswg.org/css-cascade-5/>

[CSS-SCOPING-1]

Tab Atkins Jr.; Elika Etemad. *CSS Scoping Module Level 1*. URL: <https://drafts.csswg.org/css-scoping/>

[CSS-SHADOW-1]

CSS Shadow Module Level 1. Editor's Draft. URL: <https://drafts.csswg.org/css-shadow-1/>

[CSS-VALUES-3]

Tab Atkins Jr.; Erika Etemad. *CSS Values and Units Module Level 3*. URL: <https://drafts.csswg.org/css-values-3/>

[CSS-VALUES-4]

Tab Atkins Jr.; Erika Etemad. *CSS Values and Units Module Level 4*. URL: <https://drafts.csswg.org/css-values-4/>

[CSS2]

Bert Bos; et al. *Cascading Style Sheets Level 2 Revision 1 (CSS 2.1) Specification*. URL: <https://drafts.csswg.org/css2/>

[DOM]

Anne van Kesteren. *DOM Standard*. Living Standard. URL: <https://dom.spec.whatwg.org/>

[INFRA]

Anne van Kesteren; Domenic Denicola. *Infra Standard*. Living Standard. URL: <https://infra.spec.whatwg.org/>

[RFC2119]

S. Bradner. *Key words for use in RFCs to Indicate Requirement Levels*. March 1997. Best Current Practice. URL: <https://datatracker.ietf.org/doc/html/rfc2119>

[SCROLL-ANIMATIONS-1]

Brian Birtles; et al. *Scroll-driven Animations*. URL: <https://drafts.csswg.org/scroll-animations-1/>

[SELECTORS-4]

Erika Etemad; Tab Atkins Jr.. *Selectors Level 4*. URL: <https://drafts.csswg.org/selectors/>

[WEB-ANIMATIONS-1]

Brian Birtles; et al. *Web Animations*. URL: <https://drafts.csswg.org/web-animations-1/>

[WEB-ANIMATIONS-2]

Brian Birtles; Robert Flack. *Web Animations Module Level 2*. URL: <https://drafts.csswg.org/web-animations-2/>

§ Property Index

Name	Value	Initial	Applies to	Inh.	%ages	Animation type	Canonical order	Computed value
<u>‘animation-trigger’</u>	[none [<dashed-ident> <animation-	none	all elements	no	N/A	not animatable	per grammar	as specified

Name	Value	Initial	Applies to	Inh.	%ages	Anim- ation type	Canonical order	Computed value
	action>+]+]#							
‘event-trigger’	none [<‘event-trigger-name’> <‘event-trigger-source’>]#	none	all elements	no	N/A	not animatable	per grammar	as specified
‘event-trigger-name’	none <dashed-ident>#	none	all elements	no	N/A	not animatable	per grammar	as specified
‘event-trigger-source’	[none <event-trigger-event>+ [/ <event-trigger-event>+]?]#	none	all elements	no	N/A	not animatable	per grammar	as specified
‘timeline-trigger’	none [<‘timeline-trigger-name’> <‘timeline-trigger-source’> <‘timeline-trigger-activation-range’> [/ <‘timeline-trigger-active-range’>]?]#	see individual properties	see individual properties	see individual properties	see individual properties	see individual properties	per grammar	see individual properties
‘timeline-trigger-activation-range-start’	[<‘timeline-trigger-activation-range-start’> <‘timeline-trigger-activation-range-end’>?]#	see individual properties	see individual properties	see individual properties	see individual properties	see individual properties	per grammar	see individual properties
‘timeline-trigger-activation-range-end’	[normal <length-percentage> <timeline-range-name> <length-	normal	all elements	no	relative to the specified named timeline range if	not animatable	per grammar	list, each item either the keyword normal or a timeline range and

Name	Value	Initial	Applies to	Inh.	%ages	Anim- ation type	Canonical order	Computed value
	percentage>?]#				one was specified, else to the entire timeline			progress percentage
<u>‘timeline- trigger- activation- range-start’</u>	[normal <length- percentage> <timeline- range-name> <length- percentage>?]#	normal	all elements	no	relative to the specified named timeline range if one was specified, else to the entire timeline	not animatable	per grammar	list, each item either the keyword normal or a timeline range and progress percentage
<u>‘timeline- trigger- active- range’</u>	[<'timeline- trigger-active- range-start'> <'timeline- trigger-active- range-end'>?]#	see individual properties	see individual properties	see individual properties	see individual properties	see individual properties	per grammar	see individual properties
<u>‘timeline- trigger- active- range-end’</u>	[auto normal <length- percentage> <timeline- range-name> <length- percentage>?]#	auto	all elements	no	relative to the specified named timeline range if one was specified, else to the entire timeline	not animatable	per grammar	list, each item either the keyword normal or a timeline range and progress percentage
<u>‘timeline- trigger- active- range-start’</u>	[auto normal <length- percentage> <timeline- range-name> <length- percentage>?]#	auto	all elements	no	relative to the specified named timeline range if one was specified, else to the	not animatable	per grammar	list, each item either the keyword normal or a timeline range and progress percentage

Name	Value	Initial	Applies to	Inh.	%ages	Anim- ation type	Canonical order	Computed value
					entire timeline			
‘<u>timeline-trigger-name</u>’	none <dashed- ident>#	none	all elements	no	N/A	not animatable	per grammar	as specified
‘<u>timeline-trigger-source</u>’	<single- animation- timeline>#	auto	all elements	no	N/A	not animatable	per grammar	list, each item either the keyword none, the keyword auto, a case- sensitive css identifier, a computed scroll() function, or a computed view() function
‘<u>trigger-scope</u>’	none all <dashed- ident>#	none	all elements	no	n/a	not animatable	per grammar	as specified

§ Issues Index

- ISSUE 1

The proposal I drew this text from specified that it only cares if the element is the *target* of the event. We probably want to allow for bubbling and/or capturing, possibly as an opt in/out.
- ISSUE 2

Figure out the full set of events we want to handle.