HTML ELEMENT SCRIPT INTERFACE

View online: https://www.construct.net/en/make-games/manuals/construct-3/scripting/scripting-reference/plugin-interfaces/html-element

The `IHTMLElementInstance` interface derives from IDOMInstance to add APIs specific to the HTML Element plugin.

In DOM mode the HTML Element can be accessed with standard DOM APIs using the `getElement()` method of `IDOMInstance`. However these methods can all still be used in worker mode. Most of the methods are asynchronous as in this case updates are managed by posting messages between the worker and the DOM.

HTML Element events

See instance event for standard instance event object properties.

"click"

Fired when any part of the HTML Element or its content is clicked. This event fires repeatedly for the target element followed by all parent elements up to the main HTML element, simulating event bubbling. The event object has two additional properties `targetId` and `targetClass` that can be used to identify the element clicked.

"animationend"

Fired when any CSS animation inside the HTML element finishes (based on the DOM event of the same name, but note this event is dispatched by Construct and not the DOM). The event object has three additional properties: `targetId` and `targetClass`, which can be used to identify the element whose animation ended, and `animationName` which identifies the name of the CSS animation that ended (as defined by the `@keyframes` rule).

HTML Element APIs

async setContent(str, type = "html", selector = "", isAll = false)

Replaces some content inside the HTML element with the given string `str`. The string is interpreted according to `type` which must be one of `"html"`, `"bbcode"` or `"text"`. The location to replace content is specified by a CSS selector string. This can be left blank to replace the content of the entire main HTML element. The `isAll` flag will update all elements matching the selector if set, otherwise only the first matching element is updated.

async insertContent(str, type = "html", atEnd = true, selector = "", isAll = false)

Insert the string of content `str` inside the HTML element. The string is interpreted according to `type` which must be one of `"html"`, `"bbcode"` or `"text"`. The location to insert content is specified by a CSS selector string. This can be left blank to insert the content to the main HTML element. The `atEnd` flag inserts content at the end if true, or the beginning if false. The `isAll` flag will update all elements matching the selector if set, otherwise only the first matching element is updated.

---

async setContentClass(mode, classArr, selector, isAll = false)

Adds, toggles or removes element classes according to `mode`, which must be one of `"add"`, `"toggle"` or `"remove"`. The classes to modify are given as an array of strings in `classArr`. The location to change is specified by a CSS selector string. This can be left blank to change the classes of the main HTML element. The `isAll` flag will update all elements matching the selector if set, otherwise only the first matching element is updated.

---

async setContentAttribute(mode, attrib, value, selector, isAll = false)

Adds or removes element attributes according to `mode`, which must be one of `"set"` or `"remove"`. The string of the attribute name to modify is specified by `attrib`, and its value as a string of `value` (ignored if removing the attribute). The location to change is specified by a CSS selector string. This can be left blank to change the attributes of the main HTML element. The `isAll` flag will update all elements matching the selector if set, otherwise only the first matching element is updated.

---

async setContentCssStyle(propName, value, selector, isAll)

Sets an element's style . The string of the CSS property name to modify is specified by `propName`, which may use either CSS naming (e.g. `"font-size"`) or JavaScript naming (e.g. `"fontSize"`). The value to set for this property is given by a string of `value`; set an empty string to revert the style to default. The location to change is specified by a CSS selector string. This can be left blank to change the style of the main HTML element. The `isAll` flag will update all elements matching the selector if set, otherwise only the first matching element is updated.

---

async setScrollPosition(selector, direction, position)

Set the horizontal or vertical scroll position of an element. The HTML element to scroll is given by the CSS selector string `selector`, which can be set to an empty string to scroll the main HTML element. This action only scrolls one element matching the selector. `direction` must be set to either `"left"` to set the scroll left (horizontal) position or `"top"` to set the scroll top (vertical) position. The *position* value is the scroll position to set in CSS pixels.

---

htmlContent

A string of the complete inner HTML of the main HTML element. This can also be assigned to change the full content of the HTML element.

> *Methods that change the HTML element, including assigning this property, are asynchronous. This means reading the value back won't update until the asynchronous methods have completed.*

textContent

A string of the complete inner text of the main HTML element, in plain text form (with no HTML tags). This can also be assigned to change the full text content of the HTML element.

> *Methods that change the HTML element, including assigning this property, are asynchronous. This means reading the value back won't update until the asynchronous methods have completed.*

async createSpriteImgElement(spriteInst, selector, insertAt, id, class_)

Creates an `<img>` element with the content of a given ISpriteInstance's current image, and inserts it to the HTML element. The location to insert is specified by a CSS selector string. The `insertAt` parameter must be a string of one of `"start"`, `"end"` or `"replace"` indicating how to insert the image element. The `id` and `class_` parameters are optional strings to set an ID or class for the inserted image element, which helps make it easy to style the inserted image with CSS.

> *This method provides a simple way to show a Sprite image on top of a HTML element, since normally HTML elements always show on top of Sprites.*

async positionInstanceAtElement(worldInst, selector)

Sets the position and size of a given IWorldInstance to match the position and size of a specific HTML element given by a CSS selector string.

> *This method provides a way to use invisible HTML and CSS for complex layouts, while displaying the actual content with other objects, allowing for full use of Z order, effects and so on.*