The **`currentTarget`** read-only property of the `Event` interface identifies the current target for the event, as the event traverses the DOM. It always refers to the element to which the event handler has been attached, as opposed to `Event.target`, which identifies the element on which the event occurred and which may be its descendant.

Syntax

```
var currentEventTarget = event.currentTarget;
```

Value

`EventTarget`

Examples

`Event.currentTarget` is interesting to use when attaching the same event handler to several elements.

```
function hide(e){
  e.currentTarget.style.visibility = 'hidden';
  console.log(e.currentTarget);
  // When this function is used as an event handler: this === e.current
}

var ps = document.getElementsByTagName('p');

for(var i = 0; i < ps.length; i++){
  // Console: print the clicked <p> element
  ps[i].addEventListener('click', hide, false);
}
// Console: print <body>
document.body.addEventListener('click', hide, false);
```

```
    // Click around and make paragraphs disappear
```

Note: The value of `event.currentTarget` is **only** available while the event is being handled. If you `console.log()` the `event` object, storing it in a variable, and *then* look for the `currentTarget` key in the console, its value will be `null`. Instead, you can either directly `console.log(event.currentTarget)` to be able to view it in the console or use the `debugger` statement, which will pause the execution of your code thus showing you the value of `event.currentTarget`