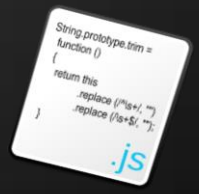
Telerik Academy

JavaScript Event Model

Create a usable interface



Telerik Software Academy

Learning & Development Team

http://academy.telerik.com

Telerik Academy

Table of Contents

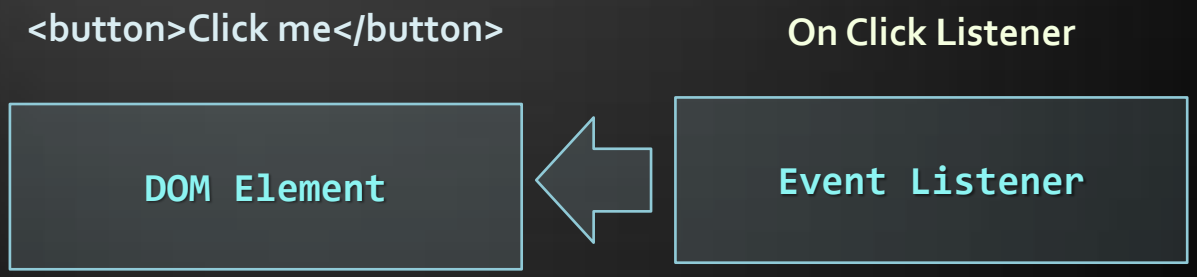
- JavaScript Event Model
- Event Registration
  - HTML Attributes, DOM Properties and Methods
- Event Object
- Cross-Browser Event Handler
- Capturing and Bubbling Events
- Custom Events

Telerik Academy

JavaScript Event Model

Event Model

- The DOM event model provides a way for the user to interact with the browser environment
- The DOM event model consists of events and event listeners attached to the DOM objects



Event Types

- DOM provides a set of common event types that are used in 99% of the time
  - Mouse events
  - Touch events
  - Form events
  - Keyboard events
  - DOM events
- Full list of all DOM event types:
  - http://www.w3.org/TR/DOM-Level-3-Events/#event-types-list
- You could also defined Custom Event Types

Telerik Academy

Common Event Types

- Mouse Events
  - click
  - hover
  - mouseup
  - mousedown
  - mouseover
  - mouseout
- Keyboard Events
  - keydown
  - keypress
  - keyup

Telerik Academy

Common Event Types (2)

- UI Events
  - load
  - abort
  - select
  - resize
  - change
- Focus Events
  - blur
  - focus
  - focusin
  - focusout

Common Event Types (3)

- Touch Events
  - touchstart
  - touchend
  - touchcancel
  - touchleave
  - touchmove

No event for tap: use click

Telerik Academy

Event Registration

Event Handlers

- The developer could register an event handler/listener for a specific event type and DOM element
- The registration can be performed with:
  - HTML Attributes
  - Using DOM element properties
  - Using DOM event handler

As HTML Attibute

- Event handlers can be attached by simply setting a value to the handler attribute
  - This value is pure JavaScript and is not always a function

```
<button>Click Me</button>
```

```
<button onclick="buttonClickFunction()">Click Me</button>
```

```
function buttonClickFunction() {
  console.log("You click the Button");
}
```

Telerik Academy

Register Event Handlers using HTML Attributes

Live Demo

Telerik Academy

Using DOM Element Properties

- Use standard DOM events on certain DOM element and assign a reference to a function
  - Can be anonymous

```
<button id="click-button">Click me</button>
var button = document.getElementById("click-button");
button.onclick = function onButtonClick() {
  console.log("You clicked the button");
}
```

Telerik Academy

Using DOM Element Properties

Live Demo

Using DOM

- The standard way for attaching event handlers to DOM
  - The Basic Syntax is:

```
domElement.addEventListener(eventType,
                            eventHandler,
                            isCaptureEvent)
```

  - Example:

```
var button = document.getElementById("click-button");

button.addEventListener("click", function () {
  console.log("You clicked me");
}, false);
```

Telerik Academy

Registering Event Handlers Using DOM

Live Demo

Telerik Academy

The Event Object

Get the Event data

Event Object

- The event handlers have access to the event object passed as function parameter
- The event object contains information about:
  - The type of the event
  - The target of the event
  - The key that was pressed when a keyboard event was fired
  - The mouse button that was pressed when a mouse event was fired
  - The position of the mouse on the screen

Event Object (2)

- The event object is accessible as the only argument of the function handler

```
function onButtonClick(event) {
  console.log(event.target);
  console.log(event.type);
  console.log("(" + event.clientX + ", " + event.clientY + ")");
}

button.addEventListener("click", onButtonClick, false);
```

- Yet, there is IE - it does not pass event object
  - Keeps the event object in `window.event`
  - Fortunately there is a simple fix

```
function onButtonClick(event) {
  if(!event) event = window.event;
  // Your code…
}
```

Telerik Academy

Event Object

Live Demo

Telerik Academy

Cross-Browser Event Handler

Remember a certain browser?

Telerik Academy

Cross-browser Compatibility

- addEventListener is not supported everywhere
  - Older versions of IE have their own method to attach event handlers
    - attachEvent("on" + eventType, handler)

```
domElement.attachEvent("on" + eventType, eventHander);
```

- Use feature detection:

```
// Up to IE8
if (document.attachEvent){
  domElement.attachEvent(…);}

// IE 9, IE 10, Firefox, Chrome, Opera, Safari
else if (document.addEventListener) {
  domElement.addEventListener(…); }

// Reeeally old browsers
else { domElement["on" + eventType] = handler; }
```

Telerik Academy

CrossBrowser Event Handler

- This can be wrapped in a method:
  - Create a function with three parameters
    - Target element
    - Event type
    - Event handler
  - Use the method your browser supports

Telerik Academy

Cross-Browser Event Handler

Live Demo

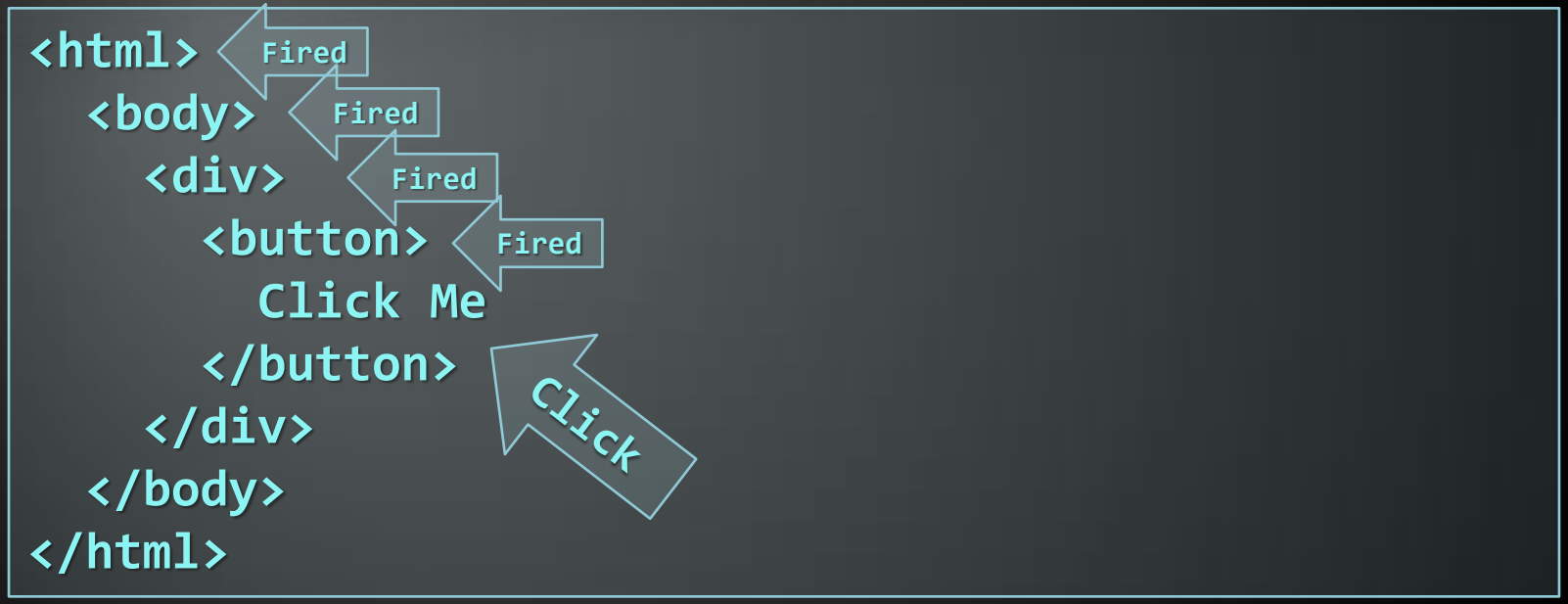
Telerik Academy

Capturing and Bubbling Events

Top to Bottom and the other way around

Event Chain

- When the user clicks on an HTML element, the event is also fired on all of its parents



- The button is still the target, but the click event is fired on all of its parents
  - An event is fired on all elements in the chain

Telerik Academy

Event Chain

Live Demo

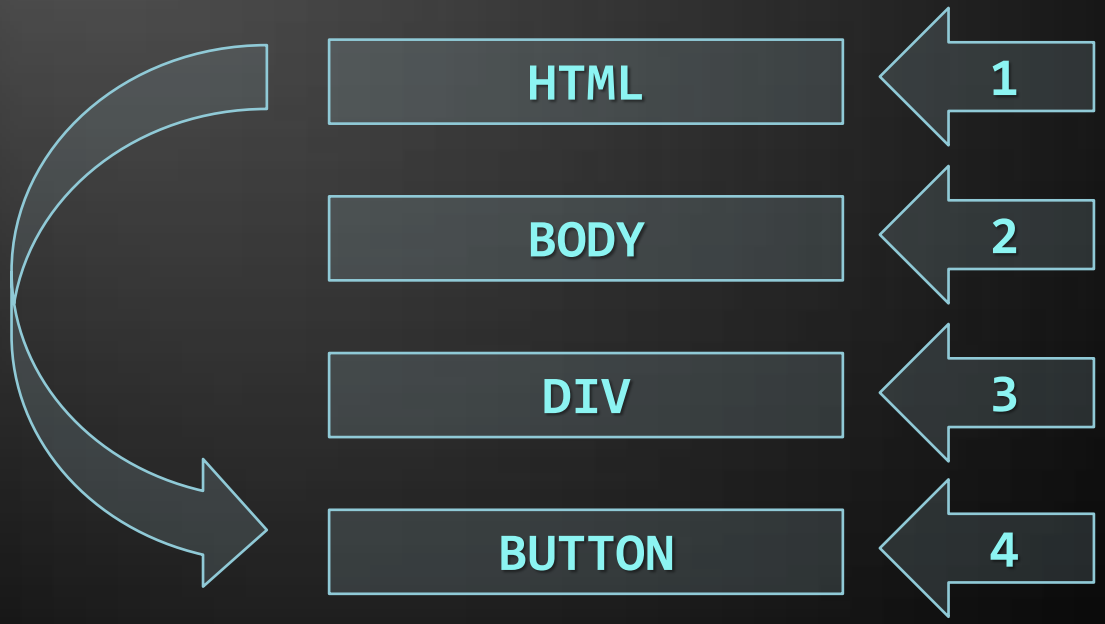
Telerik Academy

Two Types of Event Chains

- There are two types of event chains
  - Capturing and Bubbling
- Bubbling handlers bubble up to the parent
  - The first executed handler is on the target
  - Then its parent's, and its parent's, etc…
- Capturing handlers go down the chain
  - The first executed handler is on the parent of all
  - The last executed handler is on the target

Capturing

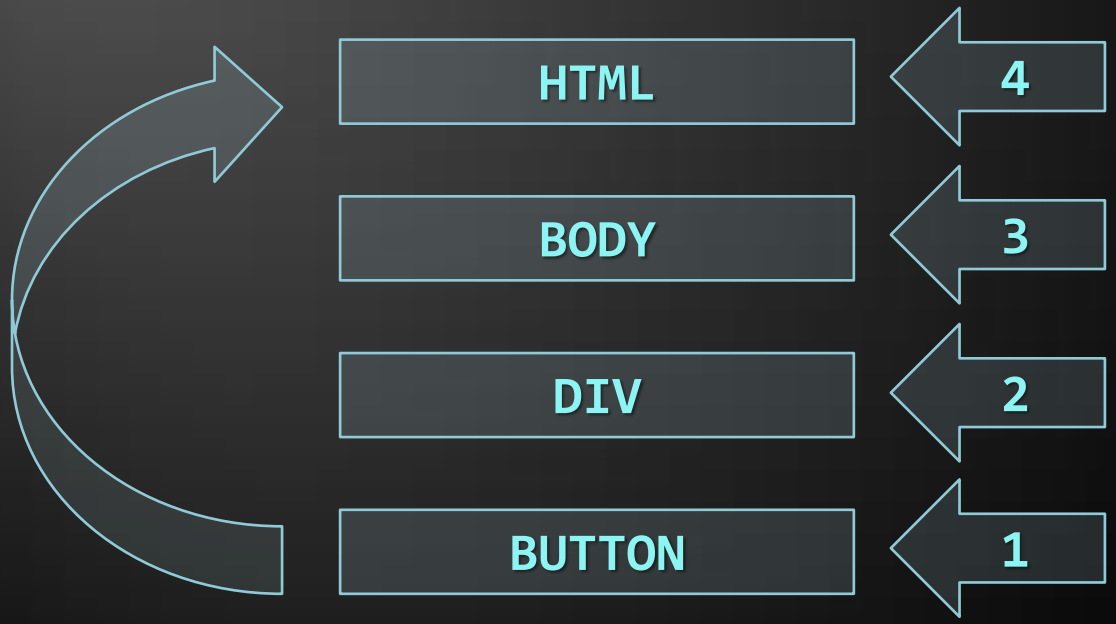
- Capturing goes down the event chain
  - The first executed handler is the one of the parent of all



User clicks the Button

Bubbling

- Bubbling bubbles up the event chain
  - The first executed handler is the one on the target

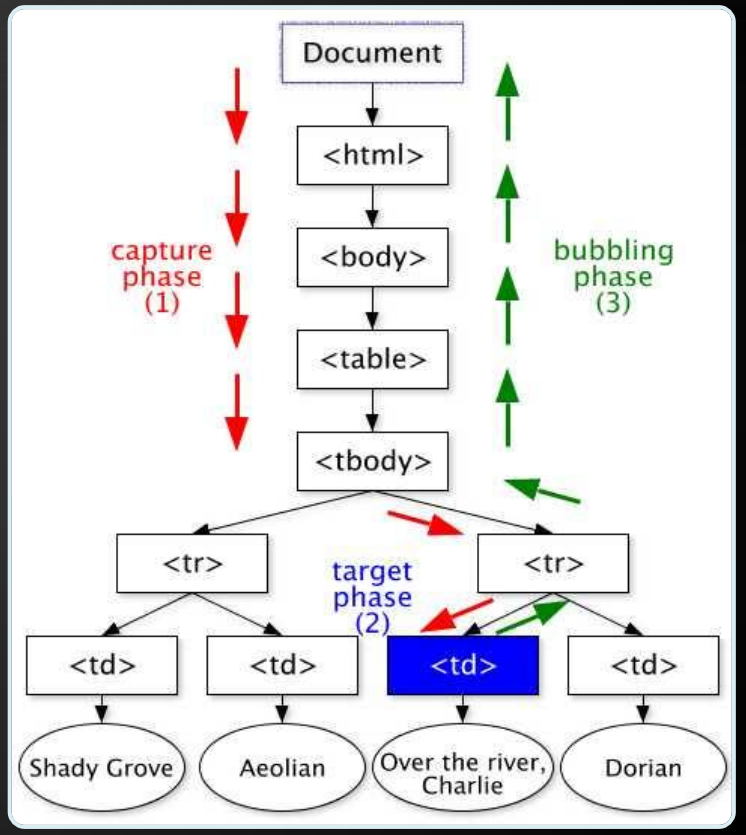


User clicks the Button

Telerik Academy

Capturing and Bubbling

Live Demo



Telerik Academy

Custom Events

CustomEvent

- To create custom events use the CustomEvent() constructor

```
var event = new CustomEvent(eventType);
```

- Example:
  - Create custom event tripleclick

```
var event = new CustomEvent("tripleClick");
```

  - Get body element to attach custom event to and addEventListener

```
var body = document.getElementsByTagName("body")[0];
body.addEventListener("tripleClick", function() {
        alert("You click three times");
}, false);
```

Custom Events (2)

- To trigger the custom event use:

```
body.dispatchEvent(event);
```

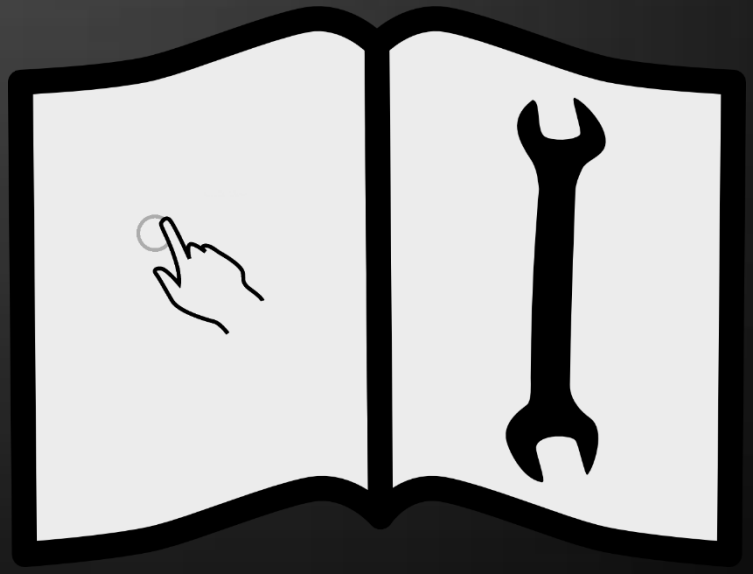
- Example

```
(function() {
  var button = document.getElementById("btn-click");
  var counter = 0;
  button.addEventListener("click", function() {
    counter++;
    if(counter == 3) {
      body.dispatchEvent(event); // Fire the event
    }
  }, false)
}());
```

Telerik Academy

Custom Events

Live Demo



Telerik Academy

JavaScript Event Model

http://academy.telerik.com

Telerik Academy

Homework

1. Create a TODO list with the following UI controls
   - Form input for new Item
   - Button for adding the new Item
   - Button for deleting some item
   - Show and Hide Button