Javascript Event Handling

Event Propagation Order

-capturing phase a.k.a

-at target inwards

-bubbling phase outwards

/\*\* Event Handler Registration

//code snippet

//<button id = “button”>Click Me</button>

//and the following event handler function

//function handler (){

//event handler code

//

* //method #1 (inline HTML event attribute

non-standard, universal

support, not recommended

<button id = “button” onclick = ‘handler()’ > Click Me </button

* // method 2 (pre-DOM 12)

non-standard, universal support

var button = document.getElementByID(‘button’);

button.onclick =handler();

//can register only one handler

//remove handler by assigning null value to the attribute

* Method 3 (DOM **11** event registration..

// standard. **aell**.supported

Var button = document.queryselector (‘button’);

button.addEventListener(‘click’,

handler);

button.addEventListener(‘Click’,handler,true’)

//can register more than one event handler

//can register handlers om the bubble capture phase.

//use removeEventListener( same arguments as re

//use dispatchEvent() to trigger event programmatically

* Method #4 (Microsoft, specific non-standard

var button = document.getElementById(‘button’)

button.attachEvent(‘onclick’,handler);

//can register more than one event handler

//can register handlers only on the bubble phase

//use detatchEvent() to trigger event programmatically

//cross-browser event registration

var button=document.getElementById(‘button’);

if (button.addEventListener

else if ( button.attachEvent

// for method #1 registration function handler(obj) {

//obj is the element that is handling the event

// for method #2 & #3

function handler() {

var element=thisl

} this is the element that is handling the event

Whenever an event is triggered information about the event is made available via the ‘ event’ object property of the ‘window’ object

For method #1 registration the event object may be passed

Creating

Event Object Properties & Methods

* Bubbles
* eventPhase
* target
* currentTarget
* preventDefault ()
* stopPropagation()
* stopImmediatePropagation()
* other event specific information
* Creating and Dispatching Events

Var clickEvent = new Event(‘click’)

var button = document.querysector(‘# button’);

button.dispatchEvent(clickEvent);

var re-new CustomEvent(‘pindot’)

button.addEventListener(‘pindot’, pindotHandler);

button.dispatchEvent(ce)