PES UNIVERSITY

CELEBRATING 50 YEARS

WEB TECHNOLOGIES

Events

Vinay Joshi
Department of
Computer Science and Engineering

EVENT

Basic Concepts and Definitions

Spurthi N Anjan

Department of Computer Science and Engineering

EVENT

What are Events?

EVENT

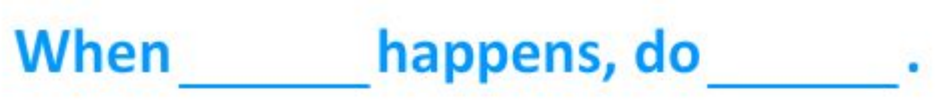
Events

Examples of HTML events:

- When a user clicks the mouse
- When a web page has loaded
- When an image has been loaded
- When the mouse moves over an element
- When an input field is changed
- When an HTML form is submitted
- When a user strokes a key

EVENT

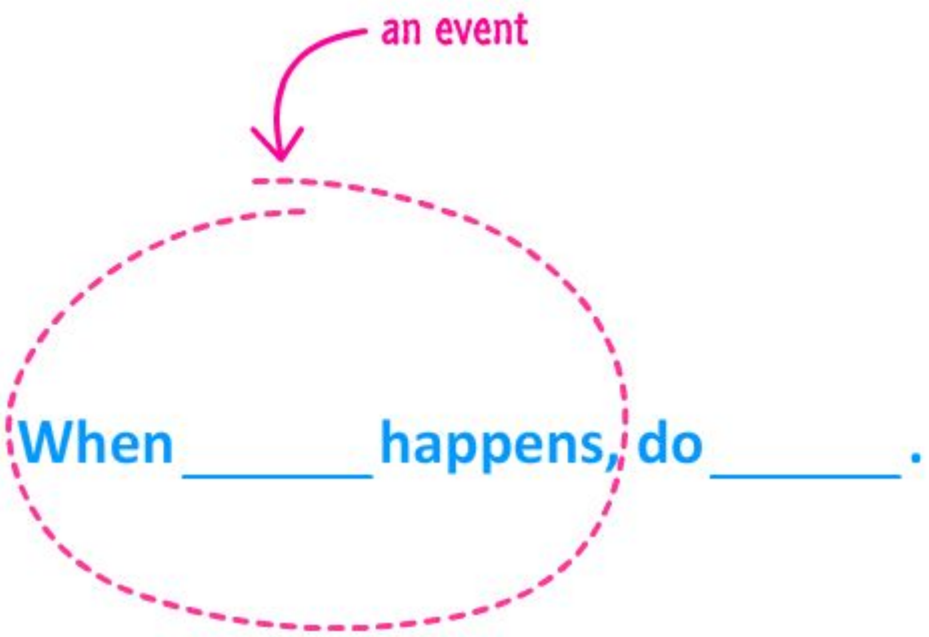
What are Events?



When ______ happens, do ______.

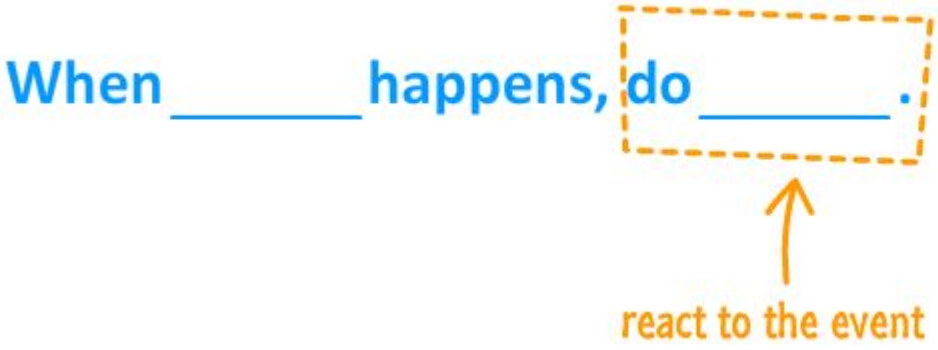
EVENT

What are Events?



EVENT

What are Events?

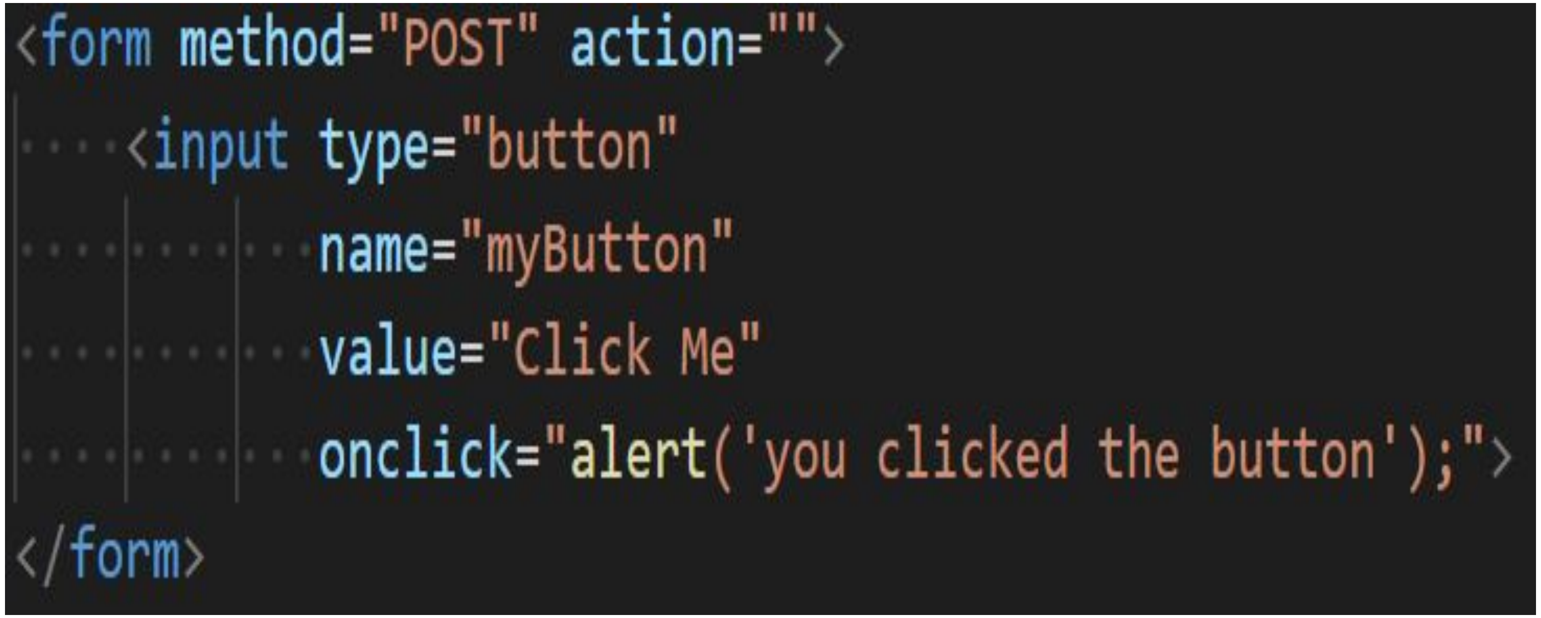


When ______ happens, do ______.

react to the event

EVENT

A simple event handler Example



```
<form method="POST" action="">
    <input type="button"
           name="myButton"
           value="Click Me"
           onclick="alert('you clicked the button');">
</form>
```

Event Handlers and Event Listeners

- Events are created by activities associated with specific HTML elements.
- The process of connecting an event handler to an event is called **registration**.
- There are two distinct approaches to event handler registration,
  - Assign element attributes
    - Inline event handlers
  - Assign handler addresses to object properties
    - Event handler properties and Event listeners

Event Handlers and Event Listeners

- Events are created by activities associated with specific HTML elements.

  For example, the click event can be caused by the browser user clicking a radio button or the link of an anchor tag.

- The process of connecting an event handler to an event is called registration.

- There are two distinct approaches to event handler registration,

  - one that assigns tag attributes

  - one that assigns handler addresses to object properties.

EVENT

Assigning events to elements

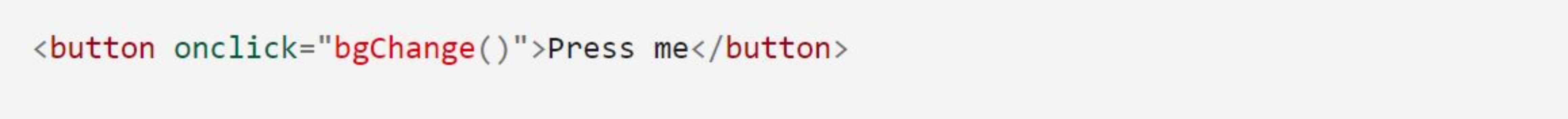
There are three ways to assign events to elements:

- Inline event handlers
- Event handler properties
- Add Event listeners

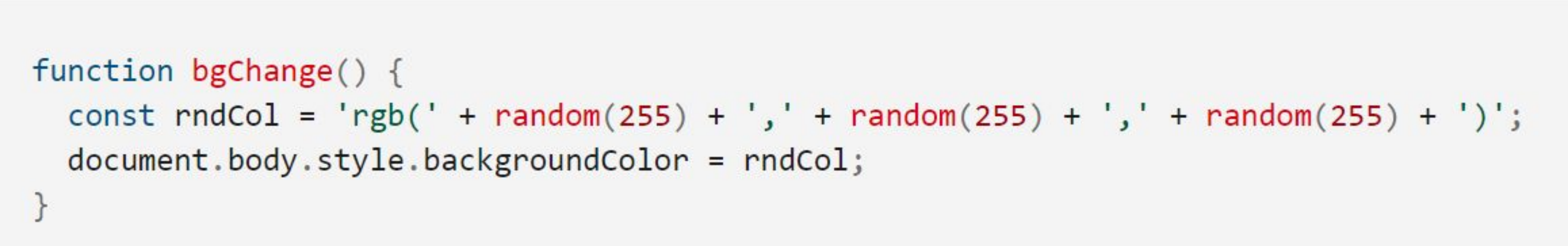
EVENT

Assigning events to elements

- Inline event handlers



```
<button onclick="bgChange()">Press me</button>
```



```
function bgChange() {
  const rndCol = 'rgb(' + random(255) + ',' + random(255) + ',' + random(255) + ')';
  document.body.style.backgroundColor = rndCol;
}
```

Event Handler Properties

- An element can be assigned the event handler property

element.on<event> = handler;

- It involves two parts
  - the correct event name it is to be listening for
  - the handler callback function.

For example:

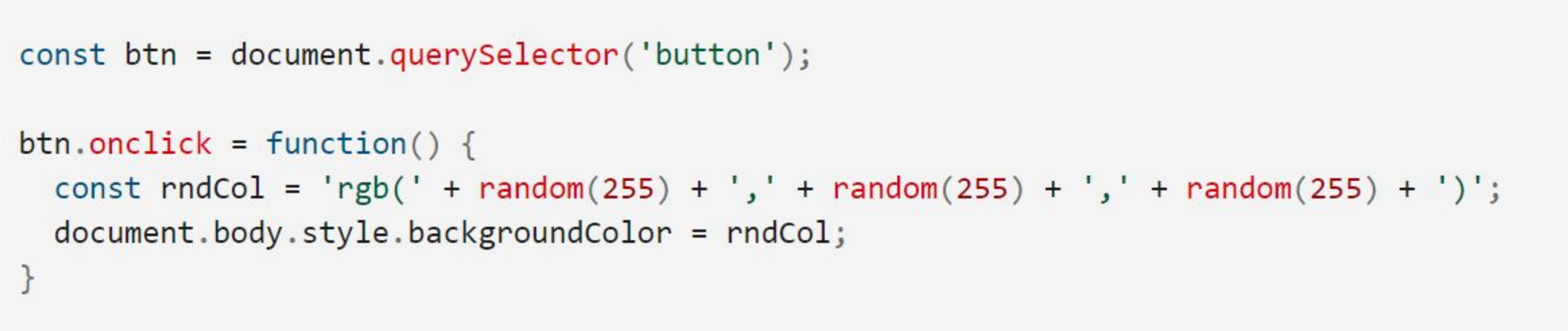
div.onclick = change; or

div.onmouseover = function(){ … };

EVENT

Assigning events to elements

- Event handler properties



```
const btn = document.querySelector('button');

btn.onclick = function() {
  const rndCol = 'rgb(' + random(255) + ',' + random(255) + ',' + random(255) + ')';
  document.body.style.backgroundColor = rndCol;
}
```

Event Listeners

- An event listener watches for an event on an element.

element.addEventListener(event, handler)

- It takes two mandatory parameters
  - the event it is to be listening for.
  - the handler callback function.

For Example:

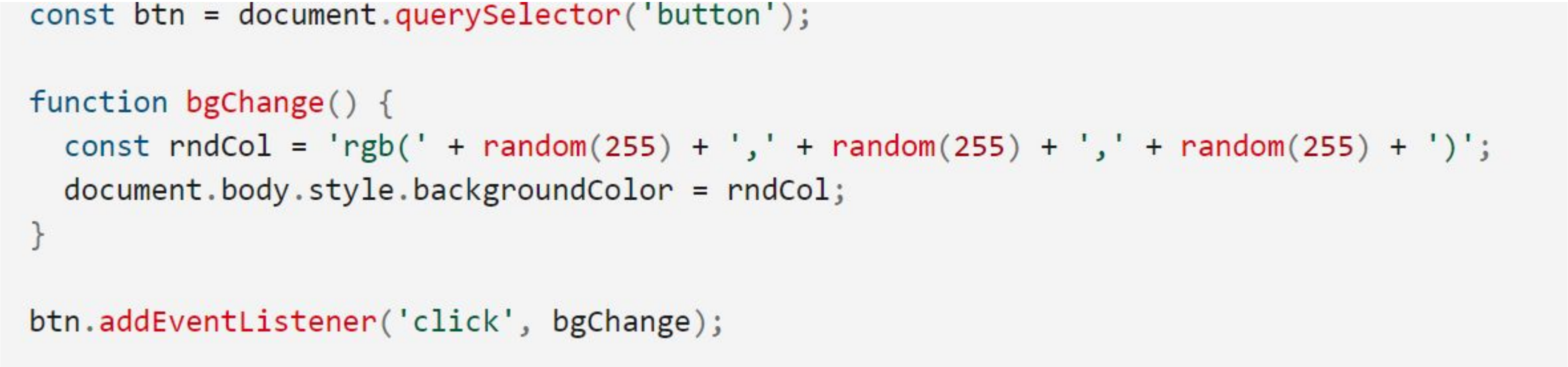
div.addEventListener(“click”, change);

div.addEventListener(“keypress”, function(){ … });

EVENT

Assigning events to elements

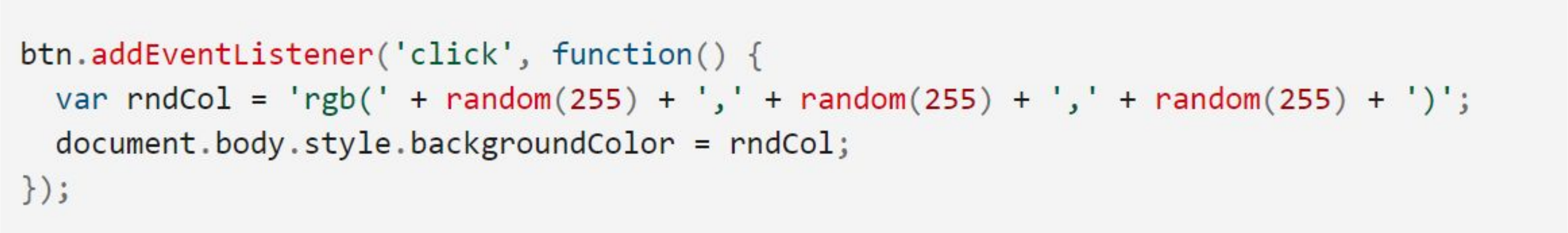
- Event listeners



```
const btn = document.querySelector('button');

function bgChange() {
  const rndCol = 'rgb(' + random(255) + ',' + random(255) + ',' + random(255) + ')';
  document.body.style.backgroundColor = rndCol;
}

btn.addEventListener('click', bgChange);
```



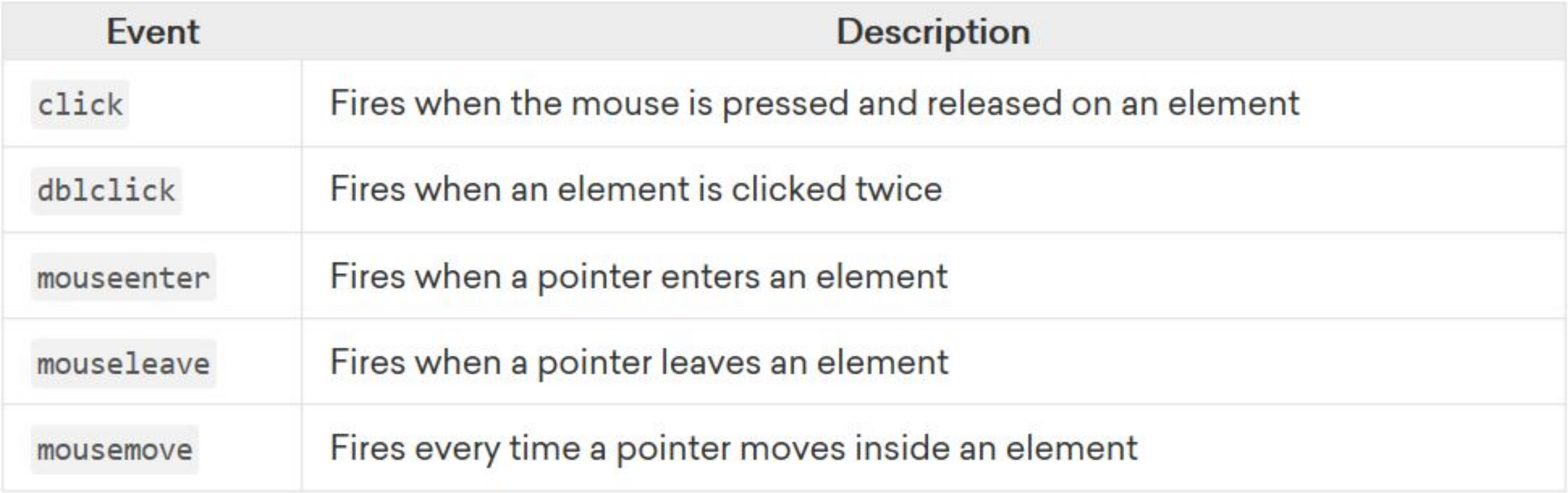
```
btn.addEventListener('click', function() {
  var rndCol = 'rgb(' + random(255) + ',' + random(255) + ',' + random(255) + ')';
  document.body.style.backgroundColor = rndCol;
});
```

Events

Event Sources and example events

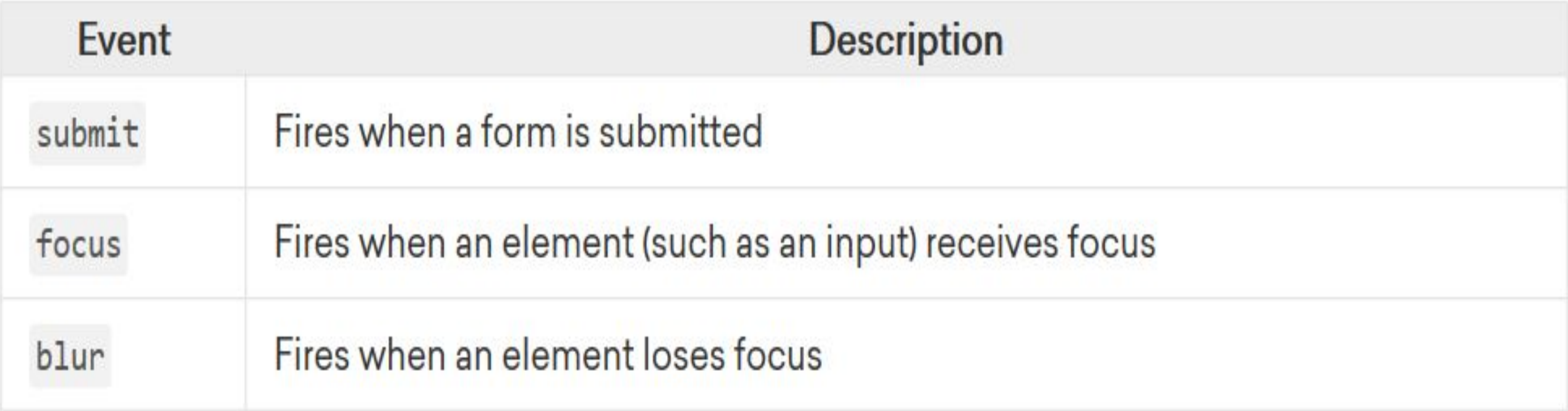
Source	Event	Fires When…
Mouse	click	the mouse is clicked and released on an element
	dblclick	an element is clicked twice
	mousemove	every time a mouse pointer moves inside an element
	mouseover	every time a mouse pointer is placed over an element
Keyboard	keydown	when a key is pressed down
	keyup	when a key pressed is released
	keypress	when a key is pressed and released
Form	submit	a form is submitted
	reset	a form reset button is clicked
	focus	an input element is clicked and receives focus
	blur	an input element loses focus

Mouse Events



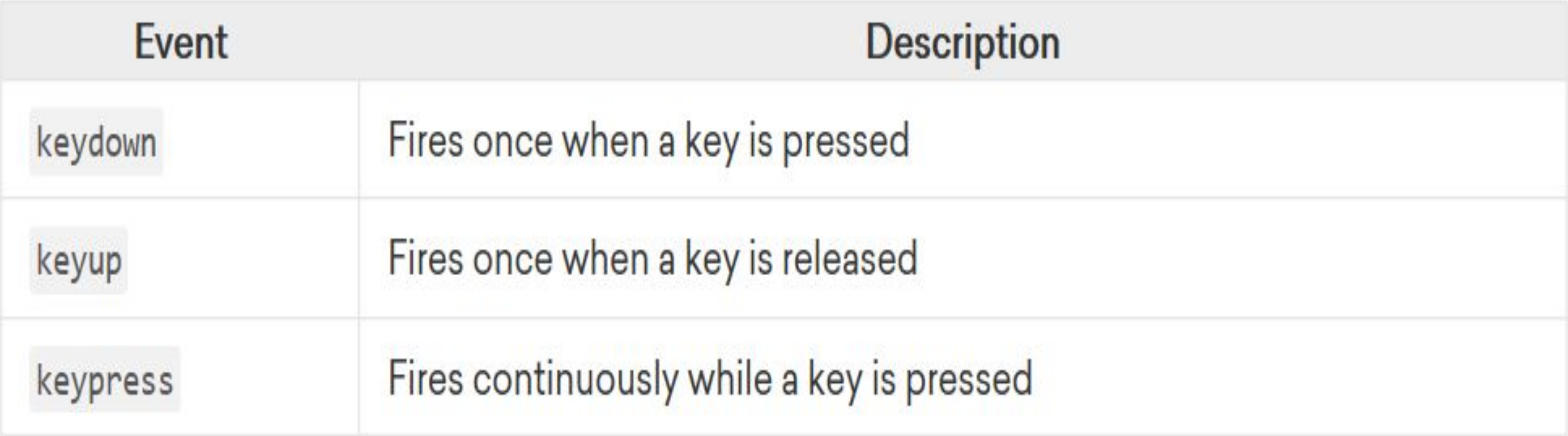
Event	Description
`click`	Fires when the mouse is pressed and released on an element
`dblclick`	Fires when an element is clicked twice
`mouseenter`	Fires when a pointer enters an element
`mouseleave`	Fires when a pointer leaves an element
`mousemove`	Fires every time a pointer moves inside an element

Form Events



Event	Description
`submit`	Fires when a form is submitted
`focus`	Fires when an element (such as an input) receives focus
`blur`	Fires when an element loses focus

Keyboard Events



Event	Description
`keydown`	Fires once when a key is pressed
`keyup`	Fires once when a key is released
`keypress`	Fires continuously while a key is pressed

Event Concepts

Event Object

- Objects have properties and methods, and respond to events.
  - Properties – specify attributes or characteristic of object .
  - Methods – specify functions object can perform.
  - Events – methods corresponding to user actions.
- These nodes can be elements, attributes, text content, comments.

Event

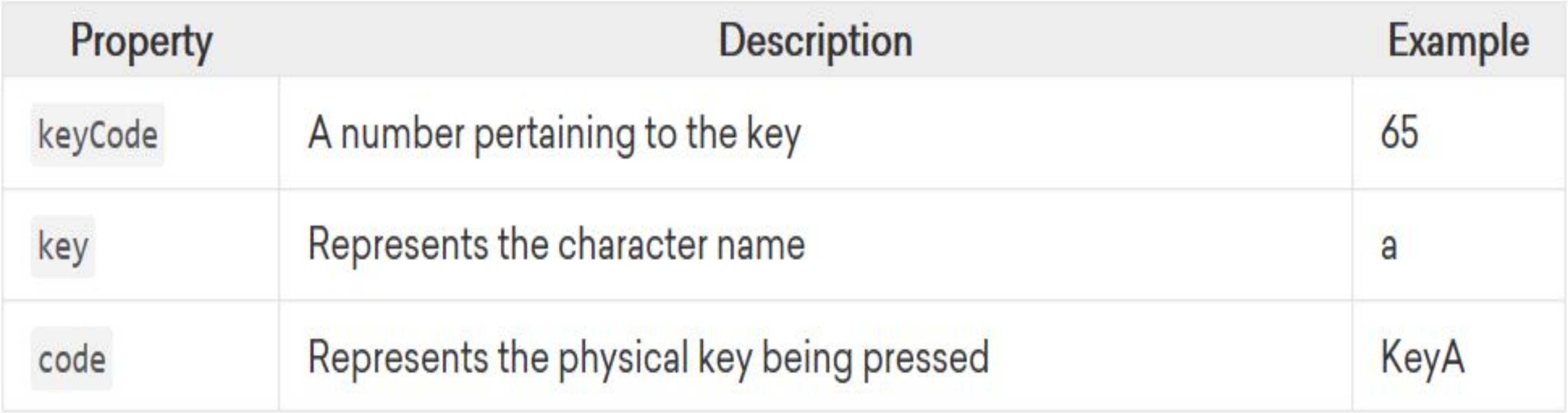
Event Object Properties

- Event object holds the context or details of the event

Property	IE5-8 Equivalent	Specifies
target	srcElement	the target of the event (most specific element).
type	-	the name of event fired (without the on prefix)
altKey / shiftKey / ctrlKey / metaKey	-	true/false to signify if Alt Key or Shift Key or Ctrl Key or Meta Key was pressed
charCode	keyCode	Unicode character code of the pressed key
key	-	Key Character Name (‘a’ or ‘F1’ or ‘CAPS LOCK’)
button	-	Returns which mouse button was pressed
clientX, clientY / offsetX, offsetY / screenX, screenY	-	the coordinates of the mouse pointer when the event triggered, relative to, the current window / target element / screen

Events

Event Object Properties (contd.)



Property	Description	Example
keyCode	A number pertaining to the key	65
key	Represents the character name	a
code	Represents the physical key being pressed	KeyA

Event Concept

Event Object Method

Method	IE5-8 Equivalent Property	Purpose
preventDefault()	returnValue	It cancels the default behavior of the event (if possible).
stopPropogation()	cancelBubble	It stops any further bubbling/ capturing of the event.

Event Delegation

- When an event occurs in an element inside another element, and both elements have registered a handle for that event.
- Propagation is the process of calling all the listeners for the given event type, attached to the nodes on the branch.
- Each listener will be called with an event object that gathers information relevant to the event.
- Several listeners can be registered on a node for the same event type.
- When the propagation reaches one such node, listeners are invoked in the order of their registration.

Event

Event Object Properties

- Event object holds the context or details of the event

Property/ Method	IE5-8 Equivalent	Purpose
target	srcElement	It specifies the target of the event (most specific element).
type	type	It specifies the type of event fired.
cancelable	Not supported	It specifies whether you can cancel the default behaviour of an element.
preventDefault()	returnValue	It cancels the default behavior of the event (if possible).
stopPropogation()	cancelBubble	It stops any further bubbling/ capturing of the event.

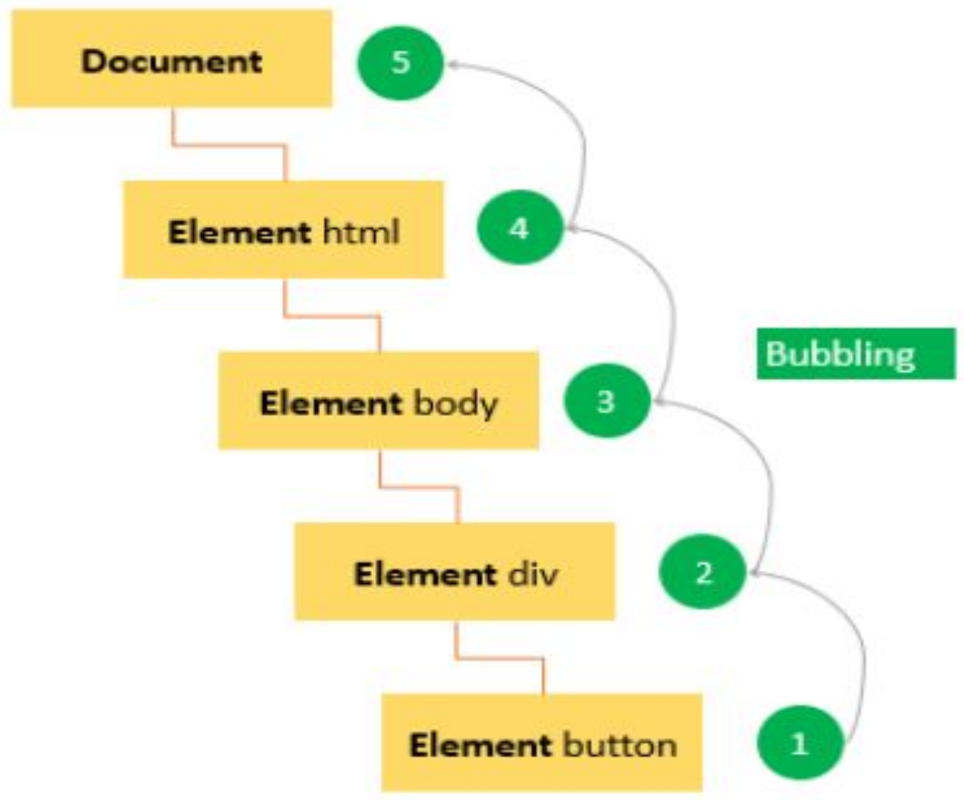
Event Flow

- There are two ways in which an event can function:
  - Event Bubbling
  - Event Capturing
- Why the event flow matters?

The flow of events only really matters when the code has event handlers on both an element and one of its ancestor/descendant elements. If the Boolean value is true, it sets the capturing phase but if false, it is the bubbling phase.

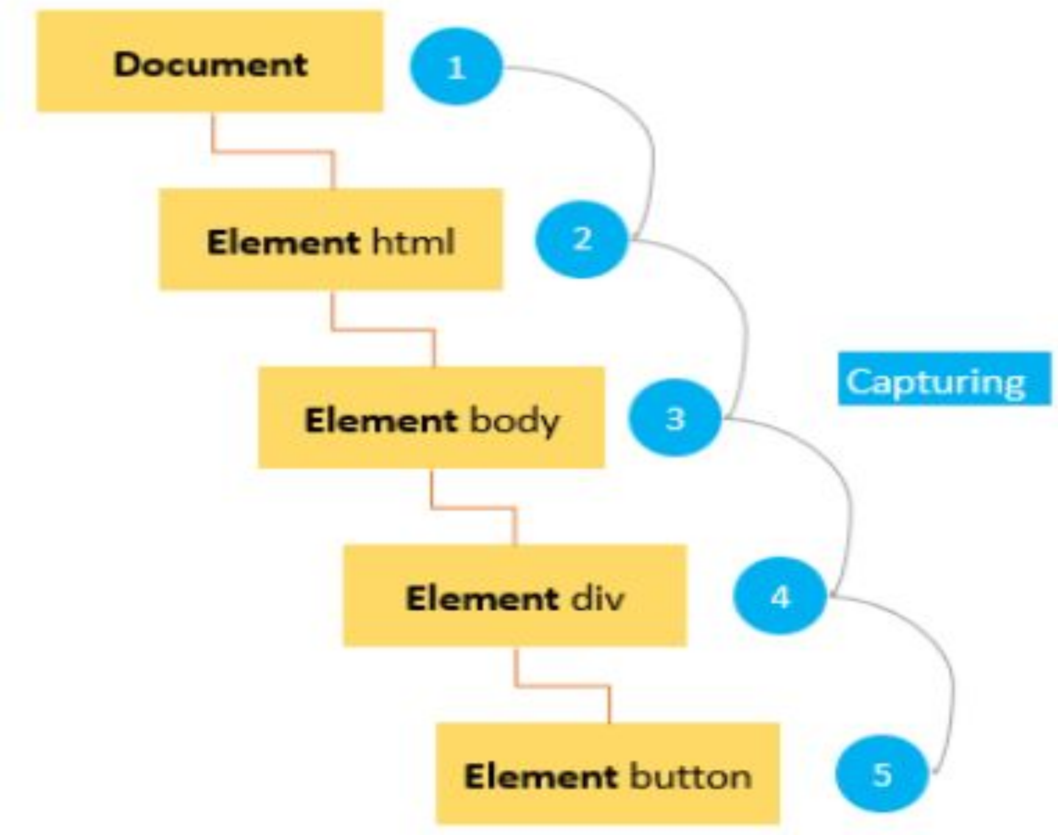
Event Flow

Event Bubbling



Event Flow

Event Capturing



PES UNIVERSITY

CELEBRATING 50 YEARS

THANK YOU

Vinay Joshi

Department of Computer Science and Engineering

vinayj@pes.edu

+91 80 2672 6622