

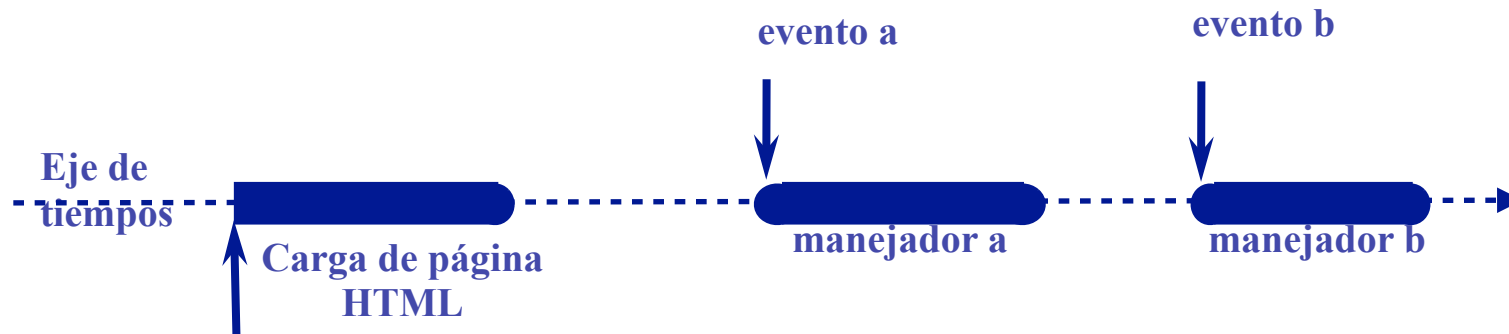


JavaScript:

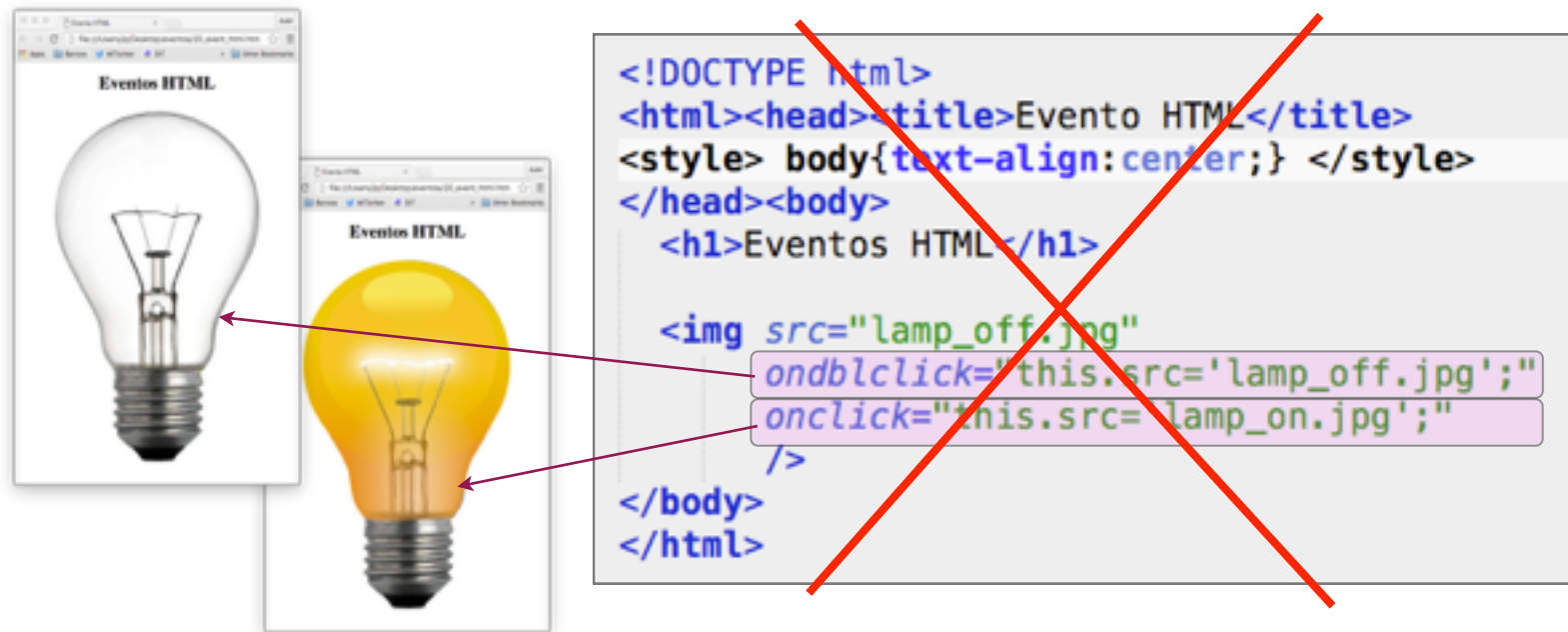
Eventos DOM y jQuery

Eventos y Manejadores

- ◆ JavaScript utiliza eventos para interaccionar con el entorno
 - Hay eventos de muchos tipos
 - ◆ Temporizadores, clicks en boton, tocar en pantalla, pulsar tecla, ...
- ◆ Un manejador (callback) de evento
 - Es una función que se ejecuta al ocurrir el evento
- ◆ El script inicial debe configurar los manejadores (callbacks)
 - a ejecutar cuando ocurra cada evento que deba ser atendido



Eventos en HTML



- ◆ En programas grandes es recomendable **separar HTML, CSS y JavaScript**
 - Debe estar en **ficheros diferentes** (o al menos en **partes claramente separadas**)
 - ♦ De esta forma el **cuerpo (body)** solo contiene **HTML** y la **cabecera (head)** incluye **CSS** y **JavaScript**
- ◆ La forma habitual de definir **eventos directamente en JavaScript** es con
 - **objetoDOM.addEventListener(evento, manejador)**
 - ♦ También existe un método **removeEventListener(..)** para **eliminar el evento**
 - <https://developer.mozilla.org/en-US/docs/Web/API/EventTarget/addEventListener>
 - Tradicionalmente el manejador se asignaba a una propiedad con el nombre del atributo HTML
 - ♦ **objeto.evento = manejador**
- ◆ jQuery usa la función **on()** para definir eventos y **off()** para eliminarlos
 - **objetojQuery.on(evento, manejador)**
 - ♦ <http://api.jquery.com/category/events/event-handler-attachment/>

Eventos en JavaScript



```
<!DOCTYPE html>
<html><head><title>Evento</title><meta charset="UTF-8">
<style> body{text-align:center;} </style>

<script type="text/javascript">

    function inicializar() {
        var img = document.getElementById('i1');

        img.addEventListener("dblclick", function () {img.src='lamp_off.jpg'});
        img.addEventListener("click", function () {img.src='lamp_on.jpg'});
    }
</script>
</head> <!-- El arbol DOM debe estar construido al ocurrir onload -->
<body onload="inicializar()">
    <h1>Eventos</h1>
    
</body></html>
```

- ◆ Los **eventos** se definen asociados a **onload** para que el **árbol DOM** esté ya está **construido**
 - El manejador del **evento onload** hay que invocarlo o definirlo o en HTML o en sino en un script al final
- ◆ La norma de JavaScript incluye **muchos eventos** diferentes
 - Se pueden ver en <https://developer.mozilla.org/en-US/docs/Web/Events>
 - ◆ Los **nombres de los eventos** son **diferentes** del de los atributos de eventos
 - La forma tradicional (objeto.evento = manejador) esta en desuso y no la ilustramos
- ◆ El **manejador** del evento es una **función** (objeto de la clase Función)
 - Puede pasarse directamente como un literal de función con el código del manejador (como aquí)
 - ◆ o cualquier otro objeto Function, como por ejemplo el nombre de una función definida en otro lugar
 - OJO! debe ser la función (su código) y no su invocación

Eventos en jQuery



```
<!DOCTYPE html>
<html><head><title>Evento jQuery</title><meta charset="UTF-8">
<style> body{text-align:center;} </style>

<script type="text/javascript" src="jquery-2.1.4.min.js" > </script>
<script type="text/javascript">
    $(function(){
        var img = $('#i1');
        img.on('dblclick', function(){img.attr('src', 'lamp_off.jpg')});
        img.on('click', function(){img.attr('src', 'lamp_on.jpg')});
    });
</script>
</head>
<body>
    <h1>Eventos</h1>
    
</body>
</html>
```

◆ jQuery también permite definir eventos en objetos jQuery con el método **on()**

■ **objetojQuery.on(evento, manejador)**

- ◆ jQuery conserva métodos asociados a eventos individuales de versiones anteriores, pero está recomendado usar solo **on()** y **off()**
- <http://api.jquery.com/category/events/>

◆ Los **nombres y tipos de eventos** utilizados por los métodos **on(..)** y **off()**

- son los mismos que los utilizados con el método **addEventListener(..)**
 - ◆ se pueden ver en <https://developer.mozilla.org/en-US/docs/Web/Events>

◆ El **manejador** del evento es una **función** (objeto de la clase Función) ejecutado al ocurrir el evento

- Puede pasarse directamente como un literal de función con el código del manejador (como aquí)
 - ◆ o cualquier otro objeto Function, como por ejemplo el nombre de una función definida en otro lugar
 - OJO! debe ser el nombre de la función (su código) y no su invocación

Calculadora jQuery

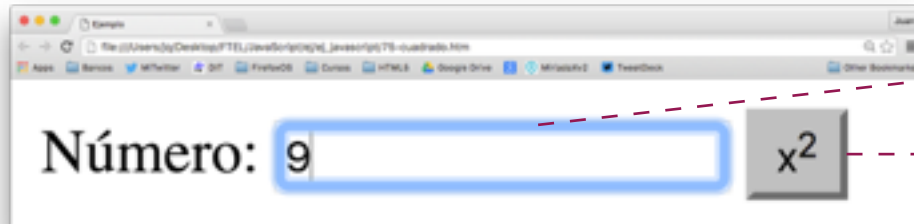
Obtener objeto jQuery (DOM) del cajetín: `$("#n1")`

Obtener objeto jQuery (DOM) del botón: `$("#b1")`

◆ jQuery simplifica la calculadora

◆ Modificaciones

- Debemos **importar** la **librería jQuery**
- Definir eventos en **función ready**
 - ◆ con **método on(..)**
 - con **árbol DOM ya construido**
- **Obtener** objetos jQuery con `$("#...")`
- **Obtener texto** de cajetín con `val()`
- **Asignar texto** en cajetín con `val(texto)`



```

<!DOCTYPE html>
<html><head><title>Calculadora</title>
    <meta char set="utf-8">
    <script type="text/javascript"
        src="jquery-2.1.4.min.js">
    </script>

    <script type="text/javascript">
    $(function() {
        $("#n1").on("click",
            function(){ $("#n1").val(""); }
        );

        $("#b1").on("click",
            function() {
                var num = $("#n1");
                num.val(num.val() * num.val());
            }
        );
    });
    </script>
</head>

<body>
    Número:
    <input type="text" id="n1">

    <button id="b1"> x<sup>2</sup> </button>
</body>
</html>

```

Importar librería jQuery

Evento click

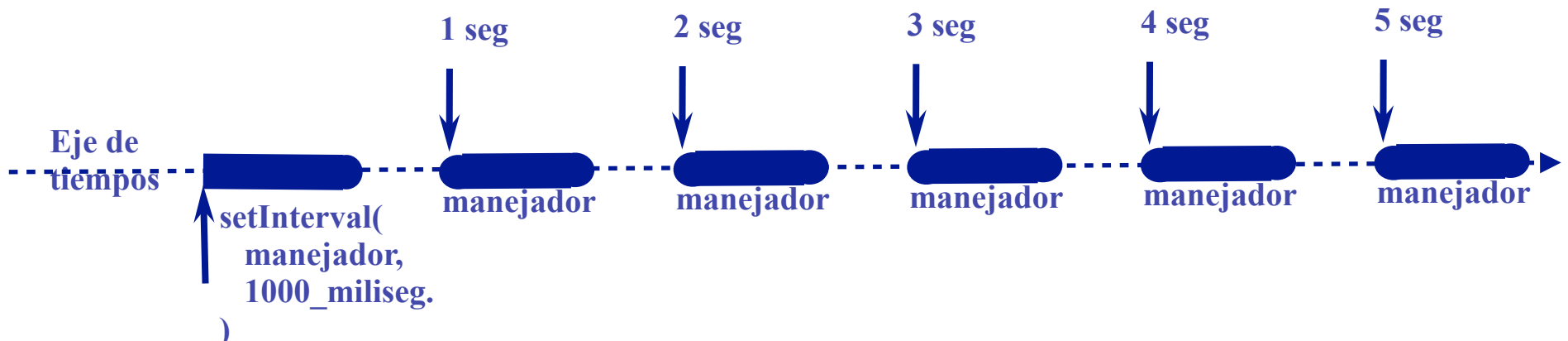
Vaciar el cajetín

Evento click

Calcular resultado obteniendo el string tecleado en cajetín con `num.val()` y guardando el resultado con `num.val(..resultado..)`.

Eventos periódicos con setInterval(...)

- ◆ JavaScript tiene una función **setInterval (..)**
 - para programar eventos periódicos
- ◆ **setInterval (manejador, periodo_en_milisegundos)**
 - tiene 2 parámetros
 - ◆ **manejador**: función que se ejecuta al ocurrir el evento
 - ◆ **periodo_en_milisegundos**: tiempo entre eventos periódicos



Reloj

Importar librería jQuery

```
<!DOCTYPE html>
<html><head><title>Reloj</title>
  <meta charset="UTF-8">
  <script type="text/javascript"
    src="jquery-2.1.4.min.js" >
  </script>
```

Mostrar
fecha en
bloque
<div>

```
<script type="text/javascript">

function mostrar_fecha( ) {
  $("#fecha").html(new Date( ));
}
```

```
$(function(){
  // Define evento periodico, ocurre
  // cada segundo (1000 miliseg)
  setInterval(mostrar_fecha, 1000);

  // muestra fecha al cargar
  mostrar_fecha();
});
```

Define un
evento que
actualiza la
hora cada
segundo

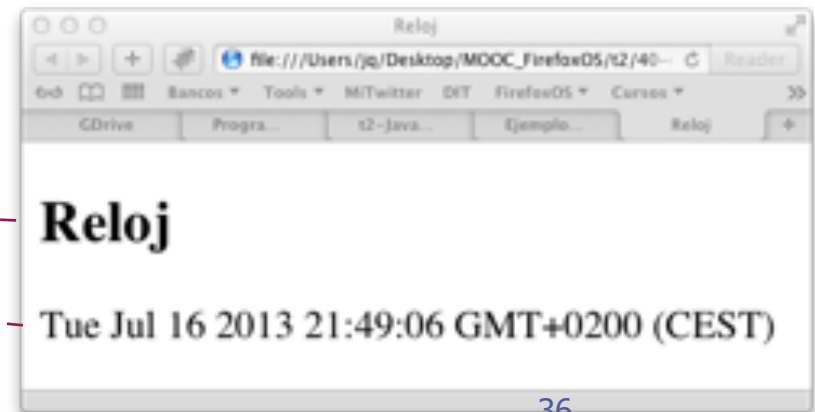
Muestra la hora al
cargar la página Web

```
</script>
</head>
<body>

  <h2>Reloj</h2>

  <div id="fecha"><div>
</body>
</html>
```

- ◆ El reloj utiliza un evento periódico
 - para actualizar cada segundo
 - ♦ la fecha y la hora mostrada en el bloque <div>
- ◆ El evento periódico se programa con
 - **setInterval(..manejador.., ..periodo..)**
 - ♦ El manejador es una función
 - ♦ El periodo se da en milisegundos
 - con árbol DOM ya construido
 - **setInterval(mostrar_fecha, 1000)**
 - ♦ Ejecuta la función mostrar_fecha()
 - cada segundo (1000 milisegundos)
- ◆ Más información en
 - https://developer.mozilla.org/en-US/Add-ons/Code_snippets/Timers





Final del tema
Muchas gracias!

