

# Métodos en JQuery

Módulo 05

# Método text()

El método **text()** permite modificar los textos de un elemento, por ejemplo de la siguiente forma

```
<p> Mi párafo <strong> importante </strong> </p>
```

```
<script>
var mensaje = $('p').text()
//guardamos el contenido del parrafo
//con el método text
//en la variable mensaje

alert(mensaje)

//lo alertamos
</script>
</body>
```

# Método html()

**La diferencia entre este método y el anterior, es que el método html() conserva el formato.**

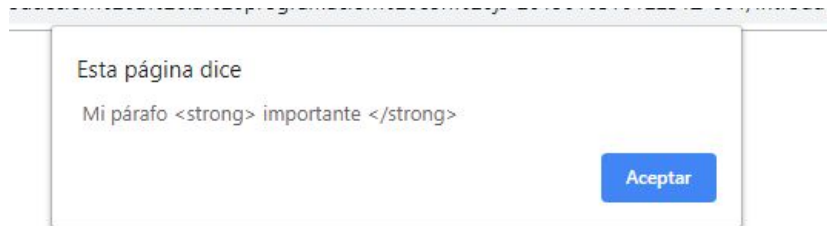
Si hiciéramos lo mismo con el texto anterior, el problema sería que en el mensaje se mostraría también que ese párrafo **tiene un strong dentro**.

Por eso si lo que queremos es simplemente mostrar un contenido en una alerta, como la alerta es una ventana que no soporta formato lo ideal es hacerlo con el **método text()** sino pasaría lo siguiente,

```
<script>
var mensaje = $('p').html()
alert(mensaje)
</script>
```

# Método html()

En el navegador se vería el código anterior así,



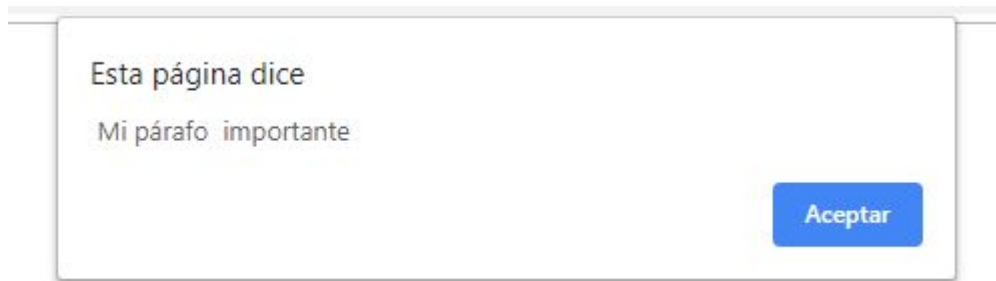
Si lo hubiésemos hecho con el **método text()** sería

```
<script src="https://ajax.googleapis.com/ajax/libs/jquery/3.3.1/jquery.min.js"></script>

<script>
var mensaje = $('p').text()
alert(mensaje)
</script>
```

# Método html()

En el navegador se verá el código anterior así,



Por eso lo ideal era trabajar con el **método text()**, veamos un ejemplo **con ID cambiando el formato con html()**,

```
<script>
$('p').html('mi párrafo que ahora tiene <em> italica </em>')
</script>
```

# Funciones y Eventos

# Funciones y Eventos

Los métodos y funciones desde **jQuery** se trabajan sin el prefijo `on`, y son mucho más simples de generar, por ejemplo la sintaxis es la siguiente,

```
<script src="https://ajax.googleapis.com/ajax/libs/jquery/3.3.1/jquery.min.js"></script>

<script>
$(document).ready(inicio)

function inicio(){

    $('selector1').evento(nombreFuncion1)
    $('selector2').evento(nombreFuncion2)
    $('selector3').evento(nombreFuncion3)}

function nombreFuncion1(){

}

function nombreFuncion2(){

}

function nombreFuncion3(){

}

</script>
```

# Funciones y Eventos

Un **caso real** sería de la siguiente forma, como vemos en la **imagen de la derecha**.

**Es decir los eventos son los mismos sólo que no los trabajamos con el prefijo on, y también el llamado a la función difiere.**

```
<script src="https://ajax.googleapis.com/ajax/libs/jquery/3.3.1/jquery.min.js">
</script>
<script>
$.ready(ini)

function ini(){
$('p').click(alertar)
$('p').dblclick(alertar2)
$('h2').mouseover(alertar3)
$('h2').mouseout(alertar4)
}

function alertar1(){alert('Me hiciste click')}

function alertar2(){alert('Me hiciste doble click')}

function alertar3(){alert('Pasaste el mouse por arriba del h2')}

function alertar4(){alert('Sacaste el cursor del mouse del h2')}
</script>

</head>

<body>

<p> Haceme click o doble click </p>

<h2> Pasa el mouse por arriba o saca el cursor </h2>
```



# Funciones y Eventos

Por ejemplo algo que también se simplifica es **pasar parámetros**, veamos un ejemplo,

```
<script>
function alertar(parametro){

    alert('Hola como estás' + parametro)

}

</script>

<p onclick="alertar('Juana')"> Haceme click </p>

<p onclick="alertar('Roberto')"> Haceme click </p>
|
```

# Funciones y val()

Ahora que ya trabajamos con **eventos**, si por ejemplo queremos obtener el valor de un elemento de formulario también será de la siguiente forma,

```
<select>

<option value="1"> Uno </option>
<option value="2"> Dos </option>
<option value="3"> Tres </option>

</select>

<script>
$('select').change(mostrarValor);

function mostrarValor(){

var valor = $('select').val()

alert(valor)

}

</script>
```

# Revisión

- Repase cómo implementar **Jquery**
- Entienda cómo trabajar con diferentes métodos como **html()**
- Comprenda la diferencia entre **html()** y **text()**
- Aprenda a implementar **funciones y parámetros** desde **Jquery**
- Diferencie **eventos** desde **Jquery**
- Combine **val()** y **funciones**
- Trabaje en el **Proyecto Integrador**
- Realice las preguntas necesarias al/la docente antes de continuar



# ¡Muchas gracias!

¡Sigamos trabajando !