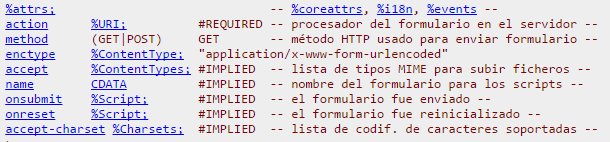
***Formularios en HTML***

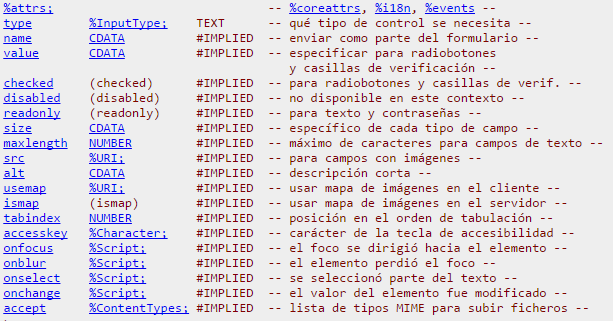
Se le considera así al lugar donde interactúa el usuario con el sitio web para proporcionar información llámese encuesta, captura de datos personales etc, y que esta a su vez se recopile en un servidor. Para lograr esto se obtiene ayuda de diversas herramientas.



El elemento Form, se podría definir como la etiqueta que representa la sección del documento html donde se iniciaran los controles pertinentes a la requisición de información. Con inicio y fin obligatorios, <form> </form>.



El elemento INPUT, útil para crear controles interactivos en los formularios para manejar contenido dinamico. Etiquetas Inicio <input>, final prohibida.



Como vimos en la parte de los controles se pueden crear varios atraves de este elemento como son : text, password (cambia caracteres por asteriscos pero no para enviar la información), checkbox, radiobuttons, botones (3 sub,reset,push), file select se puede usar VALUE para un ejemplo de nombre de fichero, hidden, Image crea un botón de envio grafico a través de un SRC.

<FORM action="http://algunsitio.com/prog/usuarionuevo" method="post">

<P>

Nombre: <INPUT type="text" name="nombre"><BR>

Apellido: <INPUT type="text" name="apellido"><BR>

email: <INPUT type="text" name="email"><BR>

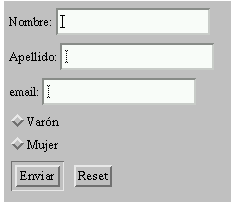
<INPUT type="radio" name="sexo" value="Varón"> Varón<BR>

<INPUT type="radio" name="sexo" value="Mujer"> Mujer<BR>

<INPUT type="submit" value="Enviar"> <INPUT type="reset">

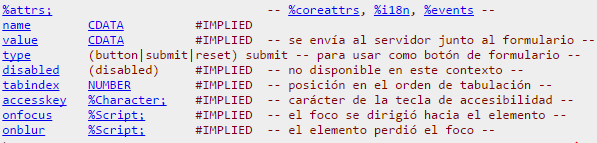
</P>

</FORM>

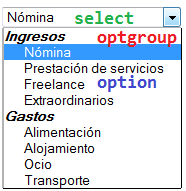


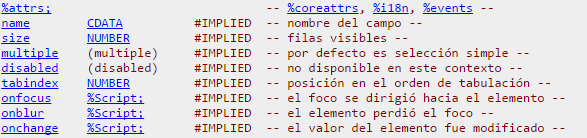
Nótese que el Label va de forma implícita a la línea de texto.

El elemento BUTTON crea botones de los tres tipos antes visto, ambas etiquetas obligatorias. <element> </element>



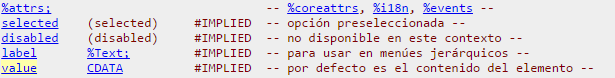
Es ilegal asociar un IMG con un USEMAP para un botón

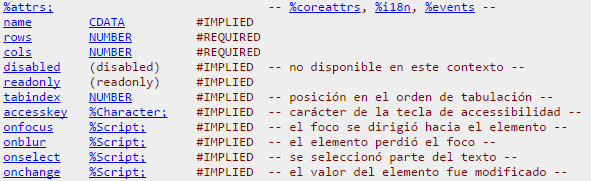
Los elementos SELECT OPTION Y OPTGROUP con etiquetas inicio-fin obligatorias



Opciones preseleccionadas, para esto es necesario tener ciertos atributos en los controles.

Para el elemento Option el atributo selected se encarga de la preseleccion.



El elemento TEXT AREA con etiquetas inicial y final obligadas, entrada de texto multilinea

<TEXTAREA name="eltexto" rows="20" cols="80">

El elemento ISINDEX texto en una sola línea, que requiere INPUT para crearlo, y llenarlo con prompt.

<ISINDEX prompt="Escriba la frase a buscar: ">

Los ROTULOS/ETIQUETAS, estos se les añaden a otros elementos para

identificarlos en la página web, y que el usuario proporcione la información pertinente a ello

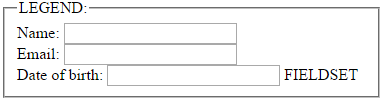


<LABEL for="nombre">Nombre: </LABEL>

<INPUT type="text" id="nombre"><BR>

Cuando el foco se dirige hacia el control asociado.

Estructura a los formularios; elementos FIELDSET & LEGEND son elementos que van de la mano pues FieldSet se encarga de agrupar controles, elementos etc, y Legend les asigna un nombre, como por decir el nombre de una carpeta.



<FIELDSET>

<LEGEND>Información Personal</LEGEND>

Apellido: <INPUT name="personal\_apellido" type="text" tabindex="1">

Nombre: <INPUT name="personal\_nombre" type="text" tabindex="2">

Dirección: <INPUT name="personal\_dirección" type="text" tabindex="3">

*...más información personal...*

</FIELDSET>

Para dirigir existen algunos métodos para direccionar el foco;

-Designar el elemento con un dispositivo apuntador.

-Navegar de un elemento a otro con el teclado.

-Seleccionar un elemento por medio de una tecla de acceso (a veces llamada "hot-key").

Navegación con tabulador, orden en que se alterna el foco a los distintos componentes del formulario, no es un atributo para todos, solo para algunos como son: A, AREA, BUTTON, INPUT, OBJECT, SELECT y TEXTAREA.

<BUTTON type="button" name="obtener-base-de-datos"

tabindex="1" onclick="obtener-base-de-datos">

Teclas de Acceso; son los accesos rápidos por medio del teclado a los elementos y se pueden asignar asi:

<LABEL for="nombre-usuario" accesskey="U">

Controles deshabilitados y de solo lectura; estos atributos se utilizan en caso de que no se requiera acceso a esta información por parte del usuario o en caso de ser irrelevante para el mismo.

En caso de estar deshabilitado; no se permite el foco hacia el elemento, por ende el orden de tabulación no se permite y no puede tener éxito y por ende no puede ser enviada la información que contenga, el atributo disabled solo soporta: BUTTON, INPUT, OPTGROUP, OPTION, SELECT y TEXTAREA.

Por el lado de solo lectura (readonly) tenemos que el foco si se puede dirigir al elemento pero solo para lectura, no hace nada mas, si puede estar en la lista de navegación con tabulador y puede que tengan o no éxito, el atributo solo aplica para INPUT y TEXTAREA.

Envio de Formularios, method del elemento FORM especifica el método HTTP usado para enviar el formulario al agente procesador. Este atributo puede tener dos valores:

get: Con el método HTTP "get", el conjunto de datos del formulario se agrega al URI especificado por el atributo action (con un signo de interrogación ("?") como separador) y este nuevo URI se envía al agente procesador.

post: Con el método HTTP "post", el conjunto de datos del formulario se incluye en el cuerpo del formulario y se envía al agente procesador.

Control de éxito; y procesamiento:

Paso uno: Identificar los controles con éxito

Paso dos: Construir el conjunto de datos del formulario

Un conjunto de datos del formulario es una secuencia de parejas nombre de control/valor actual construida a partir de los elementos con éxito.

Paso tres: Codificar el conjunto de datos del formulario

El conjunto de datos del formulario se codifica a continuación de acuerdo con el tipo de contenido especificado por el atributo enctype del elemento FORM.

Paso cuatro: Enviar el conjunto de datos del formulario codificado

Finalmente, los datos codificados se envían al agente procesador designado por el atributo action usando el protocolo especificado por el atributo method.

Códigos de estado HTTP

1XX -> Respuestas informativas

• 100 - Continue

El navegador puede continuar realizando su petición (se utiliza para indicar que la primera parte de la petición del navegador se ha recibido correctamente)

• 101 - Switching Protocols

El servidor acepta el cambio de protocolo propuesto por el navegador (puede ser por ejemplo un cambio de HTTP 1.0 a HTTP 1.1)

• 102 - Processing (WebDAV - RFC 2518)

El servidor está procesando la petición del navegador, pero todavía no ha terminado (esto evita que el navegador piense que la petición se ha perdido cuando no recibe ninguna respuesta).2

• 103 - Checkpoint

Se va a reanudar una petición POST o PUT que fue abortada previamente.

2XX ->Peticiones correctas

• 200 - OK

Respuesta estándar para peticiones correctas.

• 201 - Created

La petición ha sido completada y ha resultado en la creación de un nuevo recurso.

• 202 - Accepted

La petición ha sido aceptada para procesamiento, pero este no ha sido completado. La petición eventualmente pudiere no ser satisfecha, ya que podría ser no permitida o prohibida cuando el procesamiento tenga lugar.

• 203 - Non-Authoritative Information (desde HTTP/1.1)

La petición se ha completado con éxito, pero su contenido no se ha obtenido de la fuente originalmente solicitada sino de otro servidor

• 204 - No Content

La petición se ha completado con éxito, pero su respuesta no tiene ningún contenido (la respuesta sí que puede incluir información en sus cabeceras HTTP)

• 205 - Reset Content

La petición se ha completado con éxito, pero su respuesta no tiene contenidos y además, el navegador tiene que inicializar la página desde la que se realizó la petición (este código es útil por ejemplo para páginas con formularios cuyo contenido debe borrarse después de que el usuario lo envíe)

• 206 - Partial Content

La petición servirá parcialmente el contenido solicitado. Esta característica es utilizada por herramientas de descarga como wget para continuar la transferencia de descargas anteriormente interrumpidas, o para dividir una descarga y procesar las partes simultáneamente.

• 207 - Multi-Status (Multi-Status, WebDAV)

El cuerpo del mensaje que sigue es un mensaje XML y puede contener algún número de códigos de respuesta separados, dependiendo de cuántas sub-peticiones sean hechas.

• 208 - Already Reported (WebDAV)

El listado de elementos DAV ya se notificó previamente, por lo que no se van a volver a listar

3XX ->Redirecciones

El cliente tiene que tomar una acción adicional para completar la petición.

• 300 - Multiple Choices

Indica opciones múltiples para el URI que el cliente podría seguir. Esto podría ser utilizado, por ejemplo, para presentar distintas opciones de formato para video, listar archivos con distintas extensiones o word sense disambiguation.

• 301 - Moved Permanently

Esta y todas las peticiones futuras deberían ser dirigidas a la URI dada.

• 302 - Found

Este es el código de redirección más popular, pero también un ejemplo de las prácticas de la industria contradiciendo el estándar. La especificación HTTP/1.0 (RFC 1945) requería que el cliente realizara una redirección temporal (la frase descriptiva original fue "Moved Temporarily"), pero los navegadores populares lo implementaron como 303 See Other. Por tanto, HTTP/1.1 añadió códigos de estado 303 y 307 para eliminar la ambigüedad entre ambos comportamientos. Sin embargo, la mayoría de aplicaciones web y bibliotecas de desarrollo aún utilizan el código de respuesta 302 como si fuera el 303.

• 303 - See Other (desde HTTP/1.1)

La respuesta a la petición puede ser encontrada bajo otra URI utilizando el método GET.

• 304 - Not Modified

Indica que la petición a la URL no ha sido modificada desde que fue requerida por última vez. Típicamente, el cliente HTTP provee un encabezado como If-Modified-Since para indicar una fecha y hora contra la cual el servidor pueda comparar. El uso de este encabezado ahorra ancho de banda y reprocesamiento tanto del servidor como del cliente.

• 305 - Use Proxy (desde HTTP/1.1)

Muchos clientes HTTP (como Mozilla3 e Internet Explorer) no se apegan al estándar al procesar respuestas con este código, principalmente por motivos de seguridad.

• 306 - Switch Proxy

Este código se utilizaba en las versiones antiguas de HTTP, pero ya no se usa (aunque está reservado para usos futuros)

• 307 - Temporary Redirect (desde HTTP/1.1)

Se trata de una redirección que debería haber sido hecha con otra URI, sin embargo, aún puede ser procesada con la URI proporcionada. En contraste con el código 303, el método de la petición no debería ser cambiado cuando el cliente repita la solicitud. Por ejemplo, una solicitud POST tiene que ser repetida utilizando otra petición POST.

• 308 - Permanent Redirect

El recurso solicitado por el navegador se encuentra en otro lugar y este cambio es permanente. A diferencia del código 301, no se permite cambiar el método HTTP para la nueva petición (así, por ejemplo, si envías un formulario a un recurso que ha cambiado de lugar, todo seguirá funcionando bien)

4XX -> Errores del cliente

La solicitud contiene sintaxis incorrecta o no puede procesarse.

• 400 - Bad Request

La solicitud contiene sintaxis errónea y no debería repetirse.

• 401 - Unauthorized

Similar al 403 Forbidden, pero específicamente para su uso cuando la autentificación es posible, pero ha fallado o aún no ha sido provista. Vea autenticación HTTP básica y Digest access authentication.

• 402 - Payment Required

La intención original era que este código pudiese ser usado como parte de alguna forma o esquema de Dinero electrónico o micropagos, pero eso no sucedió, y este código nunca se utilizó.

• 403 - Forbidden

La solicitud fue legal, pero el servidor rehúsa responderla dado que el cliente no tiene los privilegios para hacerla. En contraste a una respuesta 401 No autorizado, la autenticación no haría la diferencia.

• 404 - Not Found

Recurso no encontrado. Se utiliza cuando el servidor web no encuentra la página o recurso solicitado.

• 405 - Method Not Allowed

Una petición fue hecha a una URI utilizando un método de solicitud no soportado por dicha URI; por ejemplo, cuando se utiliza GET en un formulario que requiere que los datos sean presentados vía POST, o utilizando PUT en un recurso de solo lectura.

• 406 - Not Acceptable

El servidor no es capaz de devolver los datos en ninguno de los formatos aceptados por el cliente, indicados por éste en la cabecera "Accept" de la petición.

• 407 - Proxy Authentication Required

Se requiere autenticación del proxy

• 408 - Request Timeout

El cliente falló al continuar la petición - excepto durante la ejecución de videos Adobe Flash cuando solo significa que el usuario cerró la ventana de video o se movió a otro.

• 409 - Conflict

Indica que la solicitud no pudo ser procesada debido a un conflicto con el estado actual del recurso que esta identifica.

• 410 - Gone

Indica que el recurso solicitado ya no está disponible y no lo estará de nuevo. Debería ser utilizado cuando un recurso ha sido quitado de forma permanente.

Si un cliente recibe este código no debería volver a solicitar el recurso en el futuro. Por ejemplo, un buscador lo eliminará de sus índices y lo hará más rápidamente que utilizando un código 404.

• 411 - Length Required

El servidor rechaza la petición del navegador porque no incluye la cabecera Content-Length adecuada.2

• 412 - Precondition Failed

El servidor no es capaz de cumplir con algunas de las condiciones impuestas por el navegador en su petición.2

• 413 - Request Entity Too Large

La petición del navegador es demasiado grande y por ese motivo el servidor no la procesa2

• 414 - Request-URI Too Long

La URI de la petición del navegador es demasiado grande y por ese motivo el servidor no la procesa (esta condición se produce en muy raras ocasiones y casi siempre porque el navegador envía como GET una petición que debería ser POST).2

• 415 - Unsupported Media Type

La petición del navegador tiene un formato que no entiende el servidor y por eso no se procesa.2

• 416 - Requested Range Not Satisfiable

El cliente ha preguntado por una parte de un archivo, pero el servidor no puede proporcionar esa parte, por ejemplo, si el cliente preguntó por una parte de un archivo que está más allá de los límites del fin del archivo.

• 417 - Expectation Failed

La petición del navegador no se procesa porque el servidor no es capaz de cumplir con los requerimientos de la cabecera Expect de la petición.2

• 418 - I'm a teapot

Soy una tetera.

• 422 - Unprocessable Entity (WebDAV - RFC 4918)

La solicitud está bien formada, pero fue imposible seguirla debido a errores semánticos.

• 423 - Locked (WebDAV - RFC 4918)

El recurso al que se está teniendo acceso está bloqueado.

• 424 - Failed Dependency (WebDAV) (RFC 4918)

La solicitud falló debido a una falla en la solicitud previa.

• 425 - Unassigned

Definido en los drafts de WebDav Advanced Collections, pero no está presente en "Web Distributed Authoring and Versioning (WebDAV) Ordered Collections Protocol" (RFC 3648).

• 426 - Upgrade Required (RFC 7231)

El cliente debería cambiarse a TLS/1.0.

• 428 - Precondition Required

El servidor requiere que la petición del navegador sea condicional (este tipo de peticiones evitan los problemas producidos al modificar con PUT un recurso que ha sido modificado por otra parte).2

• 429 - Too Many Requests

Hay muchas conexiones desde esta dirección de internet.

• 431 Request Header Fileds Too Large)

El servidor no puede procesar la petición porque una de las cabeceras de la petición es demasiado grande. Este error también se produce cuando la suma del tamaño de todas las peticiones es demasiado grande.2

• 449

Una extensión de Microsoft: La petición debería ser reintentada después de hacer la acción apropiada.

• 451 - Unavailable for Legal Reasons

El contenido ha sido eliminado como consecuencia de una orden judicial o sentencia emitida por un tribunal.

5XX ->Errores de servidor

El servidor falló al completar una solicitud aparentemente válida.

• 500 - Internal Server Error

Es un código comúnmente emitido por aplicaciones empotradas en servidores web, mismas que generan contenido dinámicamente, por ejemplo, aplicaciones montadas en IIS o Tomcat, cuando se encuentran con situaciones de error ajenas a la naturaleza del servidor web.

• 501 - Not Implemented

El servidor no soporta alguna funcionalidad necesaria para responder a la solicitud del navegador (como por ejemplo el método utilizado para la petición).2

• 502 - Bad Gateway

El servidor está actuando de proxy o gateway y ha recibido una respuesta inválida del otro servidor, por lo que no puede responder adecuadamente a la petición del navegador.2

• 503 - Service Unavailable

El servidor no puede responder a la petición del navegador porque está congestionado o está realizando tareas de mantenimiento.2

• 504 - Gateway Timeout

El servidor está actuando de proxy o gateway y no ha recibido a tiempo una respuesta del otro servidor, por lo que no puede responder adecuadamente a la petición del navegador.2

• 505 - HTTP Version Not Supported

El servidor no soporta o no quiere soportar la versión del protocolo HTTP utilizada en la petición del navegador.2

• 506 - Variant Also Negotiates (RFC 2295)

El servidor ha detectado una referencia circular al procesar la parte de la negociación del contenido de la petición.2

• 507 - Insufficient Storage (WebDAV - RFC 4918)

El servidor no puede crear o modificar el recurso solicitado porque no hay suficiente espacio de almacenamiento libre.2

• 508 - Loop Detected (WebDAV)

La petición no se puede procesar porque el servidor ha encontrado un bucle infinito al intentar procesarla.2

• 509 - Bandwidth Limit Exceeded

Límite de ancho de banda excedido. Este código de estatus, a pesar de ser utilizado por muchos servidores, no es oficial.

• 510 - Not Extended (RFC 2774)

La petición del navegador debe añadir más extensiones para que el servidor pueda procesarla.2

• 511 - Network Authentication Required

El navegador debe autenticarse para poder realizar peticiones (se utiliza por ejemplo con los portales cautivos que te obligan a autenticarte antes de empezar a navegar).2

• 512 - Not updated

Este error prácticamente es inexistente en la red, pero indica que el servidor está en una operación de actualizado y no puede tener conexión.