



UTN.BA
UNIVERSIDAD TECNOLÓGICA NACIONAL
FACULTAD REGIONAL BUENOS AIRES

**Centro de
e-Learning**

Desarrollo en React JS

Centro de e-Learning SCEU UTN - BA.

Medrano 951 2do piso (1179) // Tel. +54 11 4867 7589 / Fax +54 11 4032 0148

www.sceu.frba.utn.edu.ar/e-learning



Módulo I

Nivelación, Javascript y configuración del entorno

Unidad I: Nivelación HTML, CSS y JS



Presentación:

React JS es una biblioteca para desarrollo web por lo cual debemos aprender conocimientos mínimos sobre los lenguajes que el navegador web interpreta.

HTML, es un lenguaje de etiquetas el cual dará la estructura para nuestras páginas web's.

CSS, veremos la sintaxis básica de CSS con el cual podremos dar estilos (diseño, colores, márgenes) a nuestras webs desarrolladas con HTML

JS, es el lenguaje de programación web por excelencia. React se desarrolló en base a este lenguaje.



Objetivos:

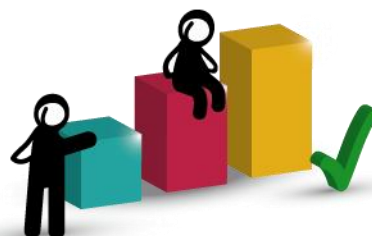
Que los participantes*:

- Aprendan a maquetar en HTML y la sintaxis básica de javascript
- Conozcan cómo dar diseño a sus páginas utilizando CSS
- Comprendan qué es Javascript, el DOM y la importancia de la estructura de una página web



Bloques temáticos*:

- ¿Qué es HTML?
- Un nuevo estándar para una nueva Web, HTML 5
- Cuáles son las novedades de HTML 5
- Un poco mas de HTML
- CSS (Hojas de estilo en cascada)
- Clases CSS
- Javascript
- Javascript – DOM



Consignas para el aprendizaje colaborativo

En esta Unidad los participantes se encontrarán con diferentes tipos de actividades que, en el marco de los fundamentos del MEC*, los referenciarán a tres comunidades de aprendizaje, que pondremos en funcionamiento en esta instancia de formación, a los efectos de aprovecharlas pedagógicamente:

- Los foros proactivos asociados a cada una de las unidades.
- La Web 2.0.
- Los contextos de desempeño de los participantes.

Es importante que todos los participantes realicen algunas de las actividades sugeridas y compartan en los foros los resultados obtenidos.

Además, también se propondrán reflexiones, notas especiales y vinculaciones a bibliografía y sitios web.

El carácter constructivista y colaborativo del MEC nos exige que todas las actividades realizadas por los participantes sean compartidas en los foros.



Tomen nota*

Las actividades son opcionales y pueden realizarse en forma individual, pero siempre es deseable que se las realice en equipo, con la finalidad de estimular y favorecer el trabajo colaborativo y el aprendizaje entre pares. Tenga en cuenta que, si bien las actividades son opcionales, su realización es de vital importancia para el logro de los objetivos de aprendizaje de esta instancia de formación. Si su tiempo no le permite realizar todas las actividades, por lo menos realice alguna, es fundamental que lo haga. Si cada uno de los participantes realiza alguna, el foro, que es una instancia clave en este tipo de cursos, tendrá una actividad muy enriquecedora.

Asimismo, también tengan en cuenta cuando trabajen en la Web, que en ella hay de todo, cosas excelentes, muy buenas, buenas, regulares, malas y muy malas. Por eso, es necesario aplicar filtros críticos para que las investigaciones y búsquedas se encaminen a la excelencia. Si tienen dudas con alguno de los datos recolectados, no dejen de consultar al profesor-tutor. También aprovechen en el foro proactivo las opiniones de sus compañeros de curso y colegas.



¿Qué es HTML?



HyperText Markup Language (lenguaje de marcas de hipertexto), hace referencia al lenguaje de marcado para la elaboración de páginas web. Es un estándar que sirve de referencia del software que conecta con la elaboración de páginas web en sus diferentes versiones, define una estructura básica y un código (denominado código HTML) para la definición de contenido de una página web, como texto, imágenes, videos, juegos, entre otros. Es un estándar a cargo del World Wide Web Consortium (W3C) o

Consortio WWW, organización dedicada a la estandarización de casi todas las tecnologías ligadas a la web, sobre todo en lo referente a su escritura e interpretación. Se considera el lenguaje web más importante siendo su invención crucial en la aparición, desarrollo y expansión de la World Wide Web (WWW). Es el estándar que se ha impuesto en la visualización de páginas web y es el que todos los navegadores actuales han adoptado.

Un nuevo estándar para una nueva Web, HTML 5

HTML 5 no es simplemente una nueva versión del lenguaje de marcación HTML, sino una agrupación de diversas especificaciones concernientes al desarrollo web. Es decir, HTML 5 no se limita sólo a crear nuevas etiquetas, atributos y eliminar aquellas marcas que están en desuso o se utilizan inadecuadamente, sino que va mucho más allá.



Así pues, HTML 5 es una nueva versión de diversas especificaciones, entre las que se encuentran:

- HTML 4
- XHTML 1
- CSS Nivel 2
- DOM Nivel 2 (DOM = Document Object Model)

A la par, HTML 5 pretende proporcionar una plataforma con la que desarrollar aplicaciones web más parecidas a las aplicaciones de escritorio, donde su ejecución dentro de un navegador no implique falta de recursos o facilidades para resolver las necesidades reales de los desarrolladores.

Cuáles son las novedades de HTML 5

HTML 5 incluye novedades significativas en diversos ámbitos. Este nuevo estándar supone mejoras en áreas que hasta ahora quedaban fuera del lenguaje y para las que se necesitaba utilizar otras tecnologías.

- Estructura del cuerpo: La mayoría de las webs tienen un formato común, formado por elementos como cabecera, pie, navegadores, etc. HTML 5 permite agrupar todas estas partes de una web en nuevas etiquetas que representarán cada uno de las partes típicas de una página.
- Etiquetas para contenido específico: Hasta ahora se utilizaba una única etiqueta para incorporar diversos tipos de contenido enriquecido, como animaciones Flash o vídeo. Ahora se utilizarán etiquetas específicas para cada tipo de contenido en particular, como audio, vídeo, etc.
- Canvas: es un nuevo componente que permitirá dibujar, por medio de las funciones de un API, en la página todo tipo de formas, que podrán estar animadas y responder a interacción del usuario. Es algo así como las posibilidades que nos ofrece Flash, pero dentro de la especificación del HTML y sin la necesidad de tener instalado ningún plugin.
- Bases de datos locales: el navegador permitirá el uso de una base de datos local, con la que se podrá trabajar en una página web por medio del cliente y a través de un API. Es algo así como las Cookies, pero pensadas para almacenar grandes cantidades de información, lo que permitirá la creación de aplicaciones web que funcionen sin necesidad de estar conectados a Internet.
- Web Workers: son procesos que requieren bastante tiempo de procesamiento por parte del navegador, pero que se podrán realizar en un segundo plano, para que el usuario no tenga que esperar que se terminen para empezar a usar la página.
- Aplicaciones web Offline: Existirá otro API para el trabajo con aplicaciones web, que se podrán desarrollar de modo que funcionen también en local y sin estar conectados a Internet.
- Geolocalización: Las páginas web se podrán localizar geográficamente por medio de un API que permita la Geolocalización.



Un poco mas de HTML

Ahora que ya entendemos que es HTML y que es HTML 5, comencemos a ver un poco más de código, que es lo que todos queremos hacer.

Este es un ejemplo de código en HTML:

```
<!DOCTYPE html>
<html>
<head>
<title>Page Title</title>
</head>
<body>

<h1>This is a Heading</h1>
<p>This is a paragraph.</p>

</body>
</html>
```

Si te preguntas ¿De qué me estás hablando? Te respondo, de un lenguaje diferente que tal vez todavía no conoces. Entonces aprendamos hablar los dos en este lenguaje.



DOCTYPE

```
<!DOCTYPE html>
```

DOCTYPE no es una etiqueta, esta es una instrucción para indicar al navegador que versión de HTML vamos a utilizar. El DOCTYPE mostrado en nuestro ejemplo es del estándar HTML 5.

Por ejemplo este es el DOCTYPE del estándar HTML 4:

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01//ES" "http://www.w3.org/TR/html4/strict.dtd">
```

Entonces, todo código html debe comenzar con el DOCTYPE

HTML

Representa la raíz de un documento HTML o XHTML. Todos los demás elementos deben ser descendientes de este elemento.

En el podemos incluir el atributo lang, por ejemplo: `<html lang="es">`.

Este atributo sirve para que el navegador identifique el lenguaje en el que está desarrollado el sitio web.



HEAD

Representa una colección de metadatos acerca del documento, incluyendo enlaces a, o definiciones de, scripts y hojas de estilo.

Metadatos del documento

- `<title>`
 - Define el título del documento, el cual se muestra en la barra de título del navegador o en las pestañas de página. Solamente puede contener texto y cualquier otra etiqueta contenida no será interpretada.
- `<base>`
 - Define la URL base para las URLs relativas en la página.
- `<link>`
 - Usada para enlazar JavaScript y CSS externos con el documento HTML actual.

```
<link rel="stylesheet" type="text/css" href="theme.css">
```

- `<meta>`
 - Define los metadatos que no pueden ser definidos usando otro elemento HTML.
Por ejemplo

```
<meta charset="UTF-8">
```

Con esta etiqueta meta definimos la codificación que tendrá nuestro archivo, los mismos pueden ser:

- UTF-8
- ANSI

ANSI es el formato estándar de codificación de archivos utilizados en el Bloc de notas. Se utiliza más comúnmente en archivos que utilizan caracteres del idioma inglés, ya que requieren menos espacio y menos tiempo para

Centro de e-Learning SCEU UTN - BA.

Medrano 951 2do piso (1179) // Tel. +54 11 4867 7589 / Fax +54 11 4032 0148
www.sceu.frba.utn.edu.ar/e-learning



procesar. UTF-8 es capaz de procesar los idiomas que utilizan más caracteres a un ritmo más rápido y más eficiente. Esto es útil de usar para las lenguas asiáticas y de Oriente Medio, ya que requieren más caracteres y no son capaces de ser procesadas en una tasa eficiente de formato ANSI.

- `<style>`
 - Etiqueta de estilo usada para escribir CSS en línea.

```
<style>
  .clase{
    background-color: rgb(255,255,255);
  }
</style>
```

Aquí estamos definiendo estilos para unas clases CSS, dándole un color de fondo al elemento.

Scripting

- `<script>`
 - Define ya sea un script interno o un enlace hacia un script externo. El lenguaje de programación es JavaScript
- `<noscript>`
 - Define un contenido alternativo a mostrar cuando el navegador no soporta scripting. Este elemento es muy importante para la construcción de sitios accesibles, ya que por ejemplo los lectores de pantallas utilizados por los no videntes no reproducen el contenido javascript.



Secciones

- **<body>**
 - Representa el contenido principal de un documento HTML. Solo hay un elemento **<body>** en un documento.
- **<section> HTML 5**
 - Define una sección en un documento.
- **<nav> HTML 5**
 - Define una sección que solamente contiene enlaces de navegación. Comúnmente utilizado para los menús en los sitios.
- **<h1>,<h2>,<h3>,<h4>,<h5>,<h6>**
 - Los elemento de cabecera implementan seis niveles de cabeceras de documentos; **<h1>** es la de mayor y **<h6>** es la de menor importancia. Un elemento de cabecera describe brevemente el tema de la sección que introduce.
Utilizado para marcar las diferentes secciones de acuerdo a su importancia, que tiene una página web.
Por ejemplo en un diario digital el título de la noticia será **<h1>** su volanta será **<h2>** y así hasta llegar al **<h6>** (No es obligatorio utilizar los 6 encabezados)
- **<header> HTML 5**
 - Define la cabecera de una página o sección. Usualmente contiene un logotipo, el título del sitio Web y una tabla de navegación de contenidos.
- **<footer> HTML 5**
 - Define el pie de una página o sección. Usualmente contiene un mensaje de derechos de autoría, algunos enlaces a información legal o direcciones para dar información de retroalimentación.



Formularios

- `<form>`
 - Representa un formulario, consistiendo de controles que puede ser enviado a un servidor para procesamiento.
- `<label>`
 - Representa el título de un control de formulario.
- `<input>`
 - Representa un campo de datos escrito que permite al usuario o usuaria editar los datos.
- `<button>`
 - Representa un botón.
- `<select>`
 - Representa un control que permite la selección entre un conjunto de opciones.
- `<option>`
 - Representa una opción en un elemento `<select>`, o una sugerencia de un elemento `<datalist>`.
- `<textarea>`
 - Representa un control de edición de texto multilínea.



Un pequeño formulario

Un poco de código

```
<!DOCTYPE HTML>
<html>
  <head>
    <title>Contacto | UTN Desarrollo para móviles</title>
    <meta charset="UTF-8">
  </head>
  <body>

    <style>
      h1{
        color: rgb(200,200,200);
      }
    </style>

    <h1>Contacto</h1>
    <h2>Completa tus datos de contacto</h2>
    <form>
      <div cla
        <label>Nombre</label>
        <input name="nombre" id="nombre"/>
      </div>
      <div>
        <label>Apellido</label>
        <input name="apellido" id="apellido"/>
      </div>
      <div>
        <label>Email</label>
        <input name="email" id="email"/>
      </div>
      <div>
        <label>Curso</label>
        <select>
          <option>Curso para desarrollo de aplicaciones móviles</option>
          <option>Desarrollo Web</option>
          <option>Project Management</option>
        </select>
      </div>
    </form>
    <h2>Nuestros teléfonos de contacto</h2>

  </body>
</html>
```

Y ahora... el resultado

Centro de e-Learning SCEU UTN - BA.

Medrano 951 2do piso (1179) // Tel. +54 11 4867 7589 / Fax +54 11 4032 0148
www.sceu.frba.utn.edu.ar/e-learning



Contacto

Completa tus datos de contacto

Nombre

Apellido

Email

Curso ▼

Nuestros teléfonos de contacto

Centro de e-Learning SCEU UTN - BA.

Medrano 951 2do piso (1179) // Tel. +54 11 4867 7589 / Fax +54 11 4032 0148
www.sceu.frba.utn.edu.ar/e-learning



CSS (Hojas de estilo en cascada)

Hoja de estilo en cascada o CSS (siglas en inglés de cascading style sheets) es un lenguaje usado para definir y crear la presentación de un documento estructurado escrito en HTML o XML2 (y por extensión en XHTML).

La idea que se encuentra detrás del desarrollo de CSS es separar la estructura de un documento de su presentación.

La información de estilo puede ser definida en un documento separado o en el mismo documento HTML. En este último caso podrían definirse estilos generales con el elemento «style» o en cada etiqueta particular mediante el atributo «style».



Incluir una hoja de estilo en el HTML

Como se mencionó anteriormente para incluir una hoja de estilo en nuestro documento HTML debemos utilizar el elemento <link>.

```
<link rel="stylesheet" type="text/css" href="theme.css">
```

Dentro del archivo CSS externo incluiremos las sentencias correspondientes, como lo hacemos con un archivo css en línea.

Por ejemplo:



```
.img-responsive {  
  max-width:100%;  
  height:auto;  
}  
.table-responsive {  
  overflow-x:auto;  
}  
a:hover, a:active {  
  color:#6E9A06;  
}  
table.lamp, table.w3-table-all {  
  margin:16px 0;  
}  
/*OPPSETT AV TOP, TOPNAV, SIDEMENU, MAIN, RIGHT OG FOOTER:*/  
.top {  
  position:relative;  
  background-color:#ffffff;  
  height:68px;  
  padding-top:20px;  
  line-height:50px;  
  overflow:hidden;  
  z-index:1;  
}
```

Clases CSS

Podemos hacer referencia a un elemento html a través de una clase o de su id en CSS.

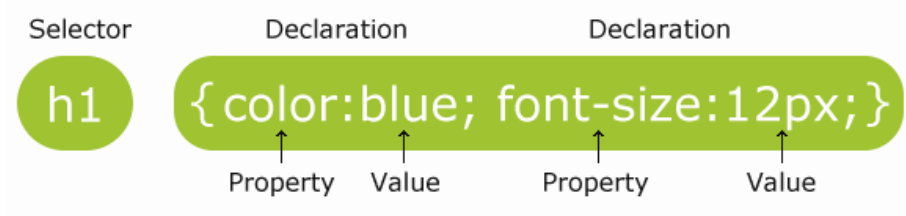
Para referencia al objeto a través de una clase lo hacemos a través de un. **(Punto)** en cambio por el id lo hacemos a través de **# (numeral)**.

```
.ejemplo_clase{  
  color: rgb(255,255,255);  
}  
#ejemplo_id{  
  color: rgb(255,255,255);  
}
```

También podemos hacer referencia a través del elemento directamente, por ejemplo aplicar un estilo a todos los inputs del documento.



```
input{  
  color: rgb(255,255,255);  
}
```



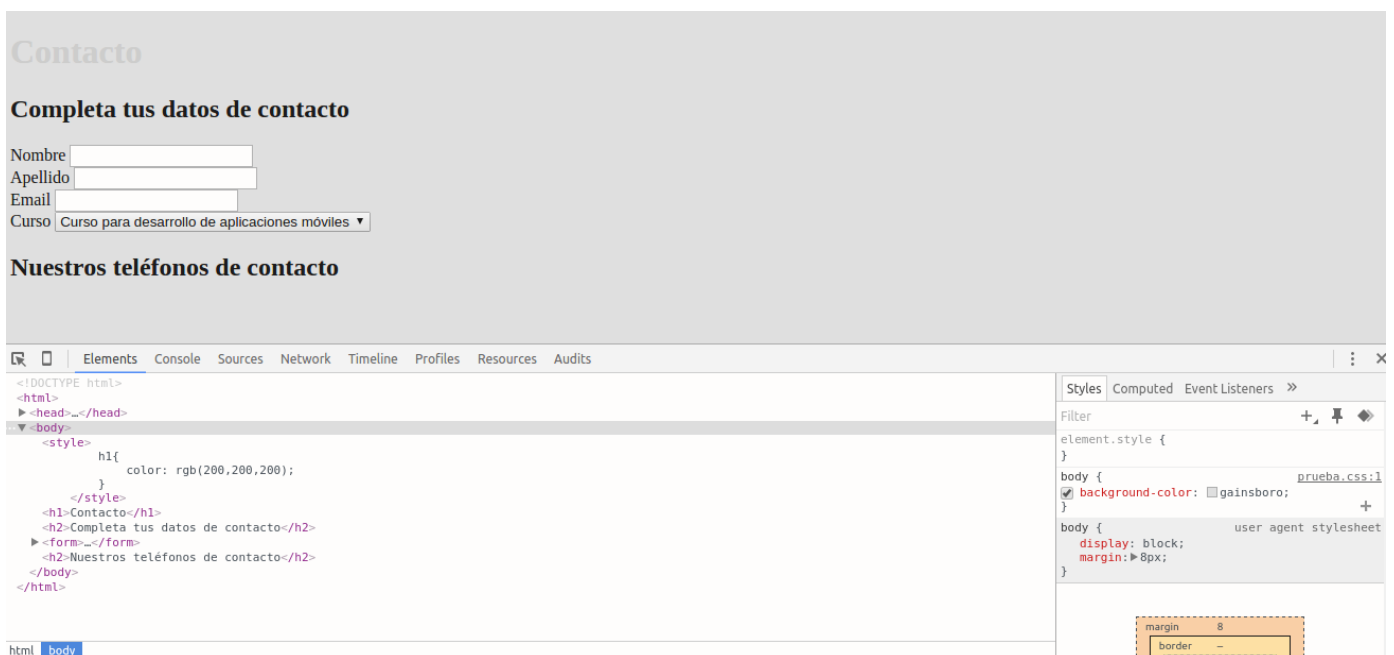
Si quieres ver las propiedades que brinda css puedes ingresar a <http://www.w3schools.com/css>

Utiliza el inspector de elementos

Para facilitar tu tarea a la hora de maquetar es importante que aprendas a utilizar el inspector de elementos.

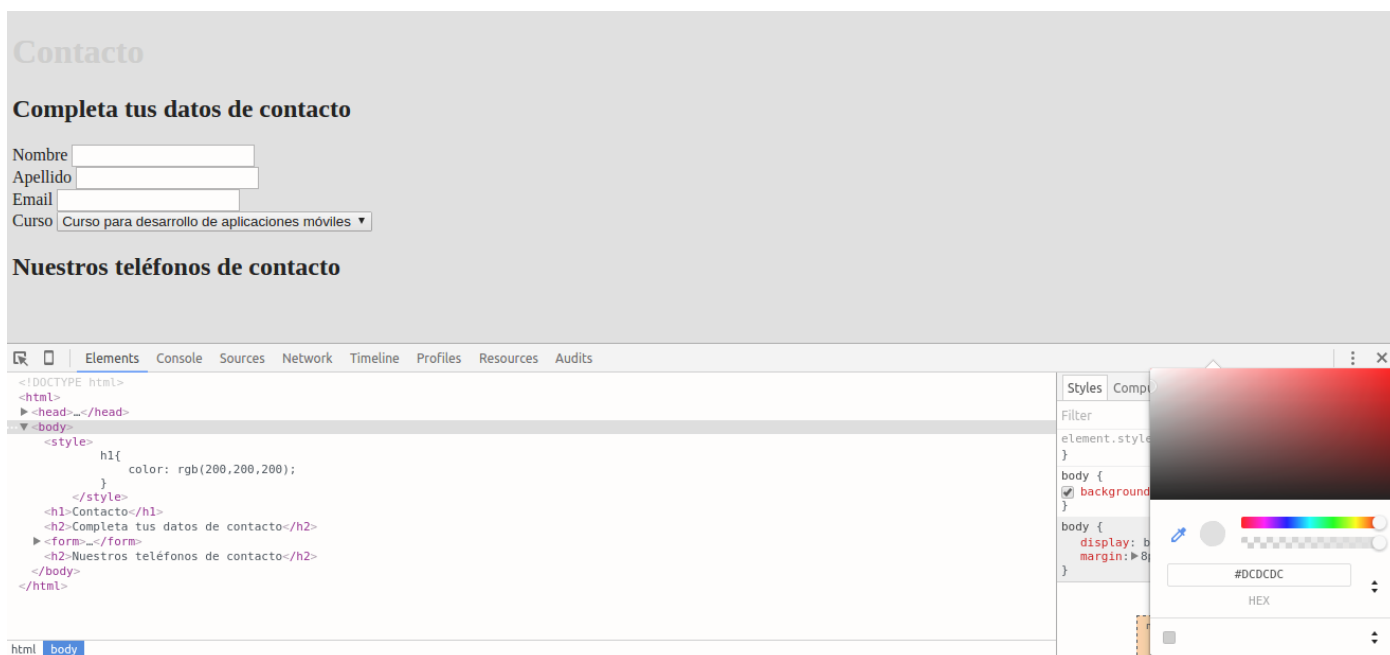
Para ello debes apretar F12 o hacer clic derecho sobre el navegador y luego inspeccionar elemento.

Por ejemplo con el inspector de elemento podrás modificar el background-color de un elemento y ver cómo queda sin necesidad de recargar la página





Debes hacer click sobre la columna derecha en el color que aparece luego del background-color.



Para ver cómo utilizar el inspector de elementos en firefox ingresa aquí:
https://developer.mozilla.org/es/docs/Tools/Page_Inspector

Sigamos con nuestro formulario de ejemplo

Este es nuestro formulario de ejemplo



Contacto

Completa tus datos de contacto

Nombre
Apellido
Email
Curso

Nuestros teléfonos de contacto

Muy bien ahora vamos a aplicarle un poco de estilos CSS. Y así quedó...

Contacto

Completa tus datos de contacto

Nombre
Apellido
Email
Curso

Nuestros teléfonos de contacto

Si tenes razón no es lo más lindo, pero que quede más lindo ahora depende de vos.



Javascript

JavaScript (abreviado comúnmente JS) es un lenguaje de programación interpretado.

Se utiliza principalmente en su forma del lado del cliente (client-side), implementado como parte de un navegador web permitiendo mejoras en la interfaz de usuario

Incluir el archivo javascript en nuestro HTML

```
<script type="text/javascript" src="prueba.js"></script>
```

Esto debe ser incluido en el head del html.

También podemos introducir nuestro código javascript dentro de la etiqueta `<script></script>`. Por ejemplo:

```
<script>
function mi_funcion() {
    document.getElementById("mi_funcion").innerHTML = "Ejemplo Mi funcion";
}
</script>
```



Tipos de ejecución

Ejecución directa

Es el método de ejecutar scripts más básico. En este caso se incluyen las instrucciones dentro de la etiqueta <SCRIPT>, cuando el navegador lee la página y encuentra un script va interpretando las líneas de código y las va ejecutando una después de otra.

Respuesta a un evento

Los eventos son acciones que realiza el usuario. Los programas como Javascript están preparados para atrapar determinadas acciones realizadas, en este caso sobre la página, y realizar acciones como respuesta.

Por ejemplo la pulsación de un botón, el movimiento del ratón o la selección de texto de la página.



Año	Nombre	Descripción
1997	ECMAScript 1	Primer edición
1998	ECMAScript 2	Editorial con algunos cambios
1999	ECMAScript 3	Añadió try/catch y expresiones regulares
	ECMAScript 4	Nunca se puso en producción
2009	ECMAScript 5	Añadió soporte a JSON
2011	ECMAScript 5.1	Cambios editoriales
2015	ECMAScript 6	Añadió clases y módulos
2016	ECMAScript 7	Añadió operador exponencial

Versiones

Centro de e-Learning SCEU UTN - BA.

Medrano 951 2do piso (1179) // Tel. +54 11 4867 7589 / Fax +54 11 4032 0148
www.sceu.frba.utn.edu.ar/e-learning



Javascript – Tag <script>

El código javascript es interpretado por los navegadores web como Chrome, Firefox e Internet Explorer o Edge. Los navegadores solo interpretan javascript (js), css y html.

Nuestro código js puede estar embebido (dentro) del código html para lo cual utilizaremos las siguientes etiquetas:

```
<script>
//Este es un comentario de una línea
/*Este comentario se puede extender
por varias líneas.
Las que quieras*/
</script>
```

También podemos desde nuestro HTML incluir un archivo con extensión .js de la siguiente manera:

```
<script src="/carrito_compra/assets/bootstrap/js/bootstrap.min.js"></script>
```

En este caso el archivo js no tendrá embebidas las etiquetas **<script>** sino que contendrá directamente el código programado bajo la sintaxis de javascript.

Propuesta:

- Crea un archivo .html y embebe código javascript dentro. Ejecute dicho archivo en el navegador (haciendo doble clic en el)
- Quita el código embebido del html, crea un archivo js con el mismo código js e incluirlo en el html.
- Abrió el navegador, apreté la tecla f12, hace clic en la pestaña consola y ejecuta el mismo código js. ¿Para qué crees que es útil esta herramienta?

El código js que puedes utilizar es el siguiente:

Centro de e-Learning SCEU UTN - BA.

Medrano 951 2do piso (1179) // Tel. +54 11 4867 7589 / Fax +54 11 4032 0148

www.sceu.frba.utn.edu.ar/e-learning



alert("Hola Mundo");



Javascript – DOM

Definición

DOM (Document Object Model o modelo de objetos del navegador) que nos sirve para acceder a cualquiera de los componentes que hay dentro de una página. Por medio del DOM podremos controlar al navegador en general y a los distintos elementos que se encuentran en la página.

Jerarquía del DOM:





UTN.BA
UNIVERSIDAD TECNOLÓGICA NACIONAL
FACULTAD REGIONAL BUENOS AIRES

**Centro de
e-Learning**

p. 30

Centro de e-Learning SCEU UTN - BA.

Medrano 951 2do piso (1179) // Tel. +54 11 4867 7589 / Fax +54 11 4032 0148
www.sceu.frba.utn.edu.ar/e-learning



Window

Todos los objetos comienzan en un objeto que se llama window. Este objeto ofrece una serie de métodos y propiedades para controlar la ventana del navegador. Con ellos podemos controlar el aspecto de la ventana, la barra de estado, abrir ventanas secundarias y otras cosas que veremos más adelante cuando expliquemos con detalle el objeto.

Además de ofrecer control, el objeto window da acceso a otros objetos como el documento (La página web que se está visualizando), el historial de páginas visitadas o los distintos frames de la ventana. De modo que para acceder a cualquier otro objeto de la jerarquía deberíamos empezar por el objeto window. Tanto es así que javascript entiende perfectamente que la jerarquía empieza en window aunque no lo señalemos.

```
<script>
window.document.bgColor = "red"
window.document.write("El texto a escribir")
</script>
```

Propiedades:

- **document:** Objeto que contiene el la página web que se está mostrando.
- **Frame:** Un objeto frame de una página web. Se accede por su nombre.
- **history:** Objeto historial de páginas visitadas.
- **location:** La URL del documento que se está visualizando. Podemos cambiar el valor de esta propiedad para movernos a otra página.
- **name:** Nombre de la ventana. Lo asignamos cuando abrimos una nueva ventana.

Para ver más: <http://www.desarrolloweb.com/articulos/826.php>



Métodos:

- **alert(texto):** Presenta una ventana de alerta donde se puede leer el texto que recibe por parámetro
- **blur():** Quitar el foco de la ventana actual
- **close():** Cierra la ventana.
- **forward():** Ir una página adelante en el historial de páginas visitadas.
- **open():** Abre una ventana secundaria del navegador.

Para ver más: <http://www.desarrolloweb.com/articulos/827.php>



Document

Se trata de las propiedades que son arrays, por ejemplo la propiedad `images` es un array con todas las imágenes de la página web. También encontramos arrays para guardar los enlaces de la página, los applets, los formularios y las anclas.

Cuando una página se carga, el navegador construye en memoria la jerarquía de objetos. De manera adicional, construye también estos arrays de objetos.

```
<script>
for (i=0;i<document.images.length;i++){
    document.write(document.images[i].src)
    document.write("<br>")
}
</script>
```

Propiedades:

- **bgColor**: El color de fondo del documento.
- **cookie**: Accede a una cookie del navegador
- **domain**: Nombre del dominio del servidor de la página.
- **fgColor**: El color del texto. Para ver los cambios hay que recargar la página.
- **ids**: Para acceder a estilos CSS.
- **title**: El título de la página.

Para ver más: <http://www.desarrolloweb.com/articulos/846.php>



Métodos:

- **close()**: Cierra el flujo del documento.
- **open()**: Abre el flujo del documento.
- **write()**: Escribe dentro de la página web. Podemos escribir etiquetas HTML y texto normal.
- **writeln()**: Escribe igual que el método write(), aunque coloca un salto de línea al final.

Para ver más: <http://www.desarrolloweb.com/articulos/861.php>



Bibliografía y Webgrafía utilizada y sugerida

Fedosejev, A. (2015). React.js Essentials (1 ed.). EEUU, Packt.

Amler, . (2016). ReactJS by Example (1 ed.). EEUU, Packt.

Stein, J. (2016). ReactJS Cookbook (1 ed.). EEUU, Packt.

<https://es.wikipedia.org/wiki/HTML>

<http://www.w3schools.com/html/>

http://www.w3schools.com/html/html5_intro.asp

<http://www.w3schools.com/css>

https://developer.mozilla.org/es/docs/Tools/Page_Inspector

<http://www.w3schools.com/Js/>



Lo que vimos:

En esta unidad aprendimos lo esencial para poder comenzar a trabajar con tecnologías web: HTML, CSS y cómo está compuesto el DOM de JS



Lo que viene:

En la próxima unidad aprendemos más sobre Javascript, su sintaxis básica, módulos y componentes con javascript ECMA script 6

