

## HTML Y MATEMÁTICAS

Objetivos: Incorporar conceptos de Programación HTML a la clase de Matemática.

La creación de páginas web, y en general cualquier tarea ligada a la programación, tiene incorporado en forma implícita o explícita una gran cantidad de conceptos matemáticos. El objetivo de esta actividad es, junto con los alumnos, reconocer y aprovechar esos conceptos matemáticos para enriquecer el proceso de creación de páginas web y, al mismo tiempo, potenciar el aprendizaje matemático.

### CONCEPTOS BÁSICOS DE HTML

- HTML es un lenguaje que se utiliza para la creación de páginas web. Por página entenderemos el documento que aparece en el visualizador o navegador.
- HTML se compone de una serie de comandos, que son interpretados por el visualizador, o programa que utilizamos para navegar por internet. En última instancia es el visualizador el que ejecuta todas las órdenes contenidas en el código HTML, de forma que un visualizador puede estar capacitado para unas prestaciones, pero no para otras. Así, podremos especificar que una página tenga una imagen de fondo, o un texto parpadeando, pero si nuestro visualizador no está capacitado para esas funciones, no podremos verlas.

### ETIQUETAS EN HTML

- El lenguaje HTML se estructura utilizando etiquetas o comandos (a partir de ahora utilizaremos indistintamente uno de 3 términos para denominar a los elementos que se estructura HTML). La forma general de una marca es la de un comando HTML encerrado entre dos signos de menor y mayor como a continuación se muestra:

**<etiqueta [atributos]> .....[</etiqueta>]**

- Hay etiquetas que se aplican a todo el documento HTML, o sólo desde el punto en que son insertadas hasta el final del documento. Otras se aplican exclusivamente a un fragmento del texto. En ese caso, el final de la aplicación se especifica con la misma marca precedida de la barra inclinada hacia atrás (/).
- Las etiquetas pueden contener de forma opcional u obligatoria, lo que se denominan atributos o modificadores. Los atributos matizan el significado de la marca, y que se expresan de la siguiente forma:

**<marca atrib1="valor1" atrib2="valor2".....>**

- El valor de los atributos se expresa encerrados entre comillas.

## ETIQUETAS HTML, HEAD Y BODY

1. Cabecera: Se inicia mediante el comando `<HEAD>` y se termina con `</HEAD>`.
  - Dentro de la cabecera hay información del documento, que no se ve en la pantalla principal, y que precisa las características del documento, principalmente el título del documento. El título del documento se declara entre las etiquetas `<TITLE>` y `</TITLE>`. El título debe ser breve y descriptivo de su contenido, pues será lo que vean los demás cuando añadan nuestra página a "favoritos", además es lo que se ve en la pestaña de navegación.
2. Cuerpo: se inicia mediante el comando `<BODY>` y se termina con el comando `</BODY>`.
  - El cuerpo del documento incluye la información que efectivamente se verá en la página, con todas las etiquetas que permitan dar al texto el formato deseado e incluir los elementos multimediales (imágenes, videos, documentos, que el programador desee).
3. Etiquetas h1 y h2
  - HTML dispone de seis cabeceras, o tipos de letra predefinidos de distinto tamaño, que se utilizan para marcar los títulos o resaltes en las páginas. Se activan con el comando `<Hn>` y se desactivan con `</Hn>`, donde n es un número de 1 a 6. siendo el número indicativo del tamaño. El tamaño mayor es el correspondiente al número 1.

Código fuente	Resultado
<pre> &lt;html&gt; &lt;head&gt; &lt;/head&gt; &lt;body&gt; &lt;h1&gt; Este es un encabezado con h1 &lt;/h1&gt; &lt;h2&gt; Este está hecho con h2 &lt;/h2&gt; &lt;h3&gt; Así se ve con h3 &lt;/h3&gt; &lt;h4&gt; h4 es más pequeña &lt;/h4&gt; &lt;h5&gt; h5 disminuye más todavía&lt;/h5&gt; &lt;h6&gt; h6 es el más pequeño &lt;/h6&gt; &lt;/body&gt; &lt;/html&gt; </pre>	<p><b>Este es un encabezado con h1</b></p> <p><b>Este está hecho con h2</b></p> <p><b>Así se ve con h3</b></p> <p><b>h4 es más pequeña</b></p> <p><b>h5 disminuye más todavía</b></p> <p><b>h6 es el más pequeño</b></p>

## ETIQUETAS P, FONT Y BR

- La marca `<P>` sirve para separar párrafos en HTML. Además de introducir un salto de línea, fuerza un segundo salto de línea para dejar una línea en blanco entre un párrafo y el siguiente. De esta manera los párrafos quedan más espaciados y el texto se vuelve más legible.
- En esta marca la etiqueta de fin es opcional.
- Por defecto, el párrafo de un documento HTML está justificado a la izquierda. Pero podemos modificar la alineación horizontal de un párrafo mediante el atributo **ALIGN** de la marca `<P>`, que puede tomar los valores siguientes.

- **LEFT:** el párrafo es justificado a la izquierda. Valor por defecto.
- **CENTER:** el párrafo es centrado.
- **RIGHT:** el párrafo es justificado a la derecha.

Este párrafo está centrado, según los márgenes de la página o del elemento en que se inserte el texto

Este párrafo está a la izquierda, si no agregas el atributo align="left", igual el texto queda a la izquierda

Este párrafo está a la derecha

### Código

```
<p align="center"> Este párrafo está centrado, según los márgenes de la página o del elemento en que se inserte el texto </p>  
<p align="left"> Este párrafo está a la izquierda, si no agregas el atributo align="left", igual el texto queda a la izquierda </p>  
<p align="right"> Este párrafo está a la derecha </p>
```

- La etiqueta **<br>** Inserta un Salto de Línea, continuando con el mismo párrafo. Esta etiqueta no requiere ser cerrada.
- La etiqueta **<Font>** permite cambiar las características de la letra a utilizar en el documento html. Se puede mezclar con las etiquetas H1, H2, P y otras.
- Dentro de sus atributos se puede encontrar
  - **Size:** define el tamaño de la letra. Se usan números de 1 a 7, el valor por defecto, si no se usa este atributo es 3. (1 y 2 son más pequeños, de 4 a 7 son más grandes)
  - **Color:** indica el color de la letra a utilizar.
- Por ejemplo

```
<p> Este es texto normal, <font color="red" size="4"> este es texto en tamaño 4, con color rojo </font>  
<font color="blue" size="7"> y este es tamaño 7 con color azul</font></p>
```

Este es texto normal, este es texto en tamaño 4, con color rojo y este es tamaño 7 con color azul

## Y LAS MATEMÁTICAS, ¿DÓNDE ESTÁN?

Uno de los conceptos más importantes de la matemática moderna, que cruza casi todas las áreas de la matemática es el concepto de FUNCIÓN y VARIABLE.

Conceptualmente podemos interpretar una función como una regla de transformación que toma una variable 'x' y nos devuelve una variable 'y' cuyo valor depende del valor de 'x' (El uso de x e y es convencional, pueden ser otras letras).

Existen funciones de una o varias variables.

Por ejemplo,  $f(x) = 2x + 3$  toma un número cualquiera, lo multiplica por 2 y luego a este resultado le suma 3. Así  $f(5) = 2 \times 5 + 3 = 13$ . Una función de varias variables podría ser  $f(a,b) = 3a + 4b$ , que toma dos números, el primero lo multiplica por 3 y el segundo por 4 y luego suma los resultados.

Así  $f(5, 7) = 3 \times 5 + 4 \times 7 = 15 + 28 = 43$

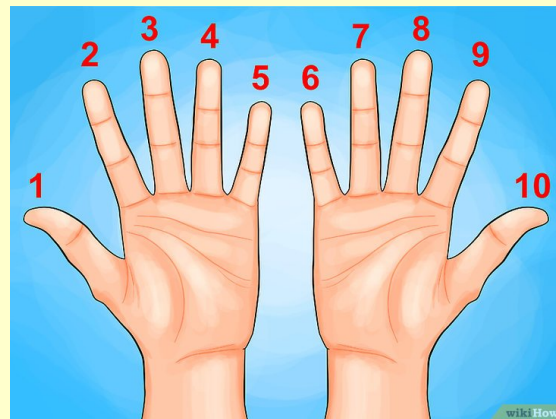
En HTML, cada vez que usamos atributos para modificar el aspecto de algún elemento estamos usando el concepto de función, donde las variables son los atributos usados con sus respectivos valores y la función nos devuelve un resultado en pantalla, cuyo aspecto dependerá de los valores asignados a las variables.

Por ejemplo `<font color="RED" size="4">` es una función que toma dos variables color y size y nos devuelve el texto con el formato definido por esas dos variables.

Mención especial tiene el tratamiento de los colores en HTML que podría cubrir muchas páginas de análisis.

Antes de analizar este tópico, revisaremos brevemente nuestro sistema de numeración.

Históricamente, al tener el ser humano 10 dedos en total en las manos, se hizo natural realizar los conteos en grupos de 10 elementos, ya que al completar las 10 unidades se hacía necesario hacer una marca para indicar cuantos grupos de 10 se habían completado, así surgen las decenas (10 unidades), y luego, por extensión, las centenas (10 decenas), las unidades de mil (10 centenas), etc. Esto además se ve reflejado en los 10 dígitos que utilizamos (0, 1, 2, 3, 4, 5, 6, 7, 8, 9). De hecho, la palabra DIGITO en latín significa DEDO (digitus).



Pero no todos los pueblos contaron en grupos de 10, algunos contaban en grupos de 16, ya que usaban ambas manos para llevar el registro con los pulgares: contaban en grupos de 4 y cada vez que completaban un grupo en la mano derecha, marcaban uno en la mano izquierda, pudiendo llegar de esta forma al 16 (4 grupos de 4)

En la actualidad los sistemas de numeración predominantes son el sistema decimal y el sistema BINARIO, que utiliza solamente dos dígitos el 0 y el 1, originado en el conteo de parejas de animales en los pueblos primitivos.

En el sistema BINARIO usamos como referencia las potencias de 2 ( $2^0=1$ ,  $2^1=2$ ,  $2^2=4$ ,  $2^3=8$ ,  $2^4=16$ ,  $2^5=32$ ,  $2^6=64$ ,  $2^7= 128$ , etc.), del mismo modo que en el sistema decimal usamos como referencia las potencias de 10 ( $10^0 = 1$ ,  $10^2=100$ ,  $10^3=1000$ ,  $10^4= 10000$ , etc.).

Así podemos establecer las siguientes equivalencias

Número Decimal	Número Binario	Lógica del número binario
1	1	1 elemento libre (no agrupado)
3	11	1 grupo de 2 1 elementos libres ( $2 + 1 = 3$ )
5	101	1 grupo de 4 0 grupos de 2 1 elemento libre ( $4 + 1 = 5$ )
14	1110	1 grupo de 8 1 grupo de 4 1 grupo de 2 0 elementos libres ( $8+4+2=14$ )
26	11010	1 grupo de 16 1 grupo de 8 0 grupo de 4 1 grupo de 2 0 elementos libres ( $16+8+2=26$ )
83	1010011	1 grupo de 64 0 grupo de 32 1 grupo de 16 0 grupo de 8 0 grupo de 4 1 grupo de 1 1 elemento libre ( $64 + 16 + 2+1= 83$ )

El uso de los números binarios se explica por el funcionamiento físico de los computadores y equipos electrónicos digitales, en cada terminal de un circuito o microchip sólo se puede identificar 2 estados básicos: hay electricidad o no hay electricidad, de esta forma si la hay se codifica con un número 1 y si no la hay se codifica con un 0. Esto también explica que los pendrives evolucionen en su capacidad en potencias de 2... los hubo de 128 Mbytes, 256 Mb, 512 Mb, 1 Gigabyte (1024 Mb), 2 Gb, 4 Gb y, actualmente, de 8 Gb, 16 Gb, 32 Gb, 64 GB, 128 Gb, 256 Gb, 512 GB, 1Terabyte (1024 Bytes) y 2 Tb.

El sistema de colores de las páginas web utiliza los números binarios para indicar la cantidad de Rojo, Verde y Azul que conforma cada color, permitiendo variaciones desde 0 hasta 255 en cada color, por ejemplo el color Marrón tiene 128 unidades de rojo, 64 unidades de Verde y nada de azul, el color Morado tiene 87 unidades de rojo, 35 unidades de Verde y 100 unidades de azul, el Azul Agua tiene 37 unidades de rojo, 109 unidades de Verde y 123 unidades de azul, el Negro tiene 0 unidades de cada color y el blanco tiene 255 unidades de cada color.

Para facilitar la codificación no se utilizan los números binarios, sino su derivado, los números HEXADECIMALES (con base 16), los cuales necesitan 16 símbolos para representarse. ¿Cómo solucionamos el problema de la falta de símbolos? Recurriendo a las letras del abecedario.

Decimal	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
Hexadecimal	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F

El Sistema Hexadecimal usa como referencia las potencias de 16 ( $16^0=1$ ,  $16^1=16$ ,  $16^2=256$ ,  $16^3= 4096$ , etc.)

Así para representar los colores indicados como ejemplo usaremos las siguientes equivalencias:

Color		Decimal	Hexadecimal	Código
Marrón	Rojo	128	<b>80</b> 8 veces 16 (128) 0 unidades	804000
	Verde	64	<b>40</b> 4 veces 16 (64) 0 unidades	
	Azul	0	<b>00</b> 0 veces 16 0 unidades	

Morado	Rojo	87	<b>57</b> 5 veces 16 (80) 7 unidades	572364
	Verde	35	<b>23</b> 2 veces 16 (32) 3 unidades	
	Azul	100	<b>64</b> 6 veces 16 (96) 4 unidades	
Azul Agua	Rojo	37	<b>25</b> 2 veces 16 (80) 5 unidades	256D7B
	Verde	109	<b>6D</b> 6 veces 16 (96) 13 unidades (13 = D)	
	Azul	123	<b>7B</b> 7 veces 16 (116) 11 unidades (11 = B)	
Negro	Rojo	0	<b>00</b> 0 veces 16 0 unidades	000000
	Verde	0	<b>00</b> 0 veces 16 0 unidades	
	Azul	0	<b>00</b> 0 veces 16 0 unidades	
Blanco	Rojo	255	<b>FF</b> 15 veces 16 (240) 15 unidades (15 = F)	FFFFFF
	Verde	255	<b>FF</b> 15 veces 16 (240) 15 unidades (15 = F)	
	Azul	255	<b>FF</b> 15 veces 16 (240) 15 unidades (15 = F)	

Afortunadamente existen muchas páginas web que permiten elegir el color en forma directa y nos entrega el código respectivo, por ejemplo,

<https://encycolorpedia.es/>

<https://www.colorhexa.com/>

Contenido relacionado:

Sistemas de Numeración (5to Básico)

Problema: ¿Cuántos colores diferentes se puede definir por este método? ¿Cuántos colores serían posibles si usáramos del mismo modo sólo 2 cifras para cada color, pero en sistema DECIMAL?

Contenido relacionado:

Combinatoria: Principio Multiplicativo

## MÁS HTML

### ETIQUETAS OL, UL Y LI

- Se usan para generar listas numeradas **<ol>** o listas con viñetas **<ul>**.
- Cada ítem de la lista se define con la etiqueta **<li>** y se finaliza con **</li>**

<code>&lt;ol&gt;</code>	
<code>&lt;li&gt; Chile &lt;/li&gt;</code>	1. Chile
<code>&lt;li&gt; Argentina &lt;/li&gt;</code>	2. Argentina
<code>&lt;li&gt; Brasil &lt;/li&gt;</code>	3. Brasil
<code>&lt;/ol&gt;</code>	
<code>&lt;ul&gt;</code>	
<code>&lt;li&gt; Santiago &lt;/li&gt;</code>	◆ Santiago
<code>&lt;li&gt; Valparaiso &lt;/li&gt;</code>	◆ Valparaiso
<code>&lt;li&gt; Iquique &lt;/li&gt;</code>	◆ Iquique
<code>&lt;li&gt; Concepción &lt;/li&gt;</code>	◆ Concepción
<code>&lt;/ul&gt;</code>	

### LAS ETIQUETAS TABLE, TH, TBODY, TR Y TD

- Una tabla es una manera muy compacta y clara de mostrar la información.
- Una tabla en HTML se entiende como un conjunto de filas (fila=horizontal), apiladas una sobre otra. Cada fila contiene a su vez un conjunto de celdas, puestas una al lado de la otra.



- Una tabla se declara usando el comando **<TABLE> ... </TABLE>**. Dentro de la tabla, se usa **<TR> ... </TR>** para indicar una fila y dentro de una fila, **<TD> ... </TD>** para delimitar el contenido de una celda.
- Todos estos comandos aceptan numerosos modificadores. Se pueden, entre otras cosas, unir celdas, especificar la alineación del contenido de la celda, y mucho más.
- **TH** permite definir los encabezados de la tabla y **TBODY** se refiere al cuerpo de la tabla. Ambas requieren comando de apertura y cierre.
  - Puedes ver ejemplos en detalle en los siguientes links:  
<https://www.uv.es/jac/guia/tablaeje.htm>  
<http://www.htmlquick.com/es/tutorials/tables.html>

## LA ETIQUETA IMG

- Permite insertar imágenes en el documento.

****

- Donde ruta del archivo es la ubicación de la imagen en el disco local del sitio o la url en caso de que sean imágenes desde internet.
- Por ejemplo

****

Inserta la imagen del archivo foto1.jpg que está grabada en la carpeta **images** que está en la misma carpeta que el archivo donde estamos insertando la imagen. Si es necesario subir una o más carpetas para llegar a la ubicación de la imagen se usa ../

****

Inserta una imagen desde el sitio innovacien.org. Se debe escribir la dirección completa (URL) de la imagen.

- Acepta los parámetros width (ancho) y height (alto) para establecer el tamaño de la imagen en pantalla.

## LA ETIQUETA A

- Esta es **LA** etiqueta más importante dentro de HTML, ya que es la que hace la diferencia entre trabajar con un simple TEXTO a trabajar con un HIPERTEXTO, es

decir un texto dinámico que no es necesario leer en forma secuencial, sino que te puede llevar mediante VINCULOS a otras partes del mismo sitio e incluso a otros sitios diferentes.

- La estructura de la etiqueta es `<a href="destino"> texto visible </a>`
- Por ejemplo:

`<a href="http://innovacien.org"> Visita el sitio de innovacien </a>`

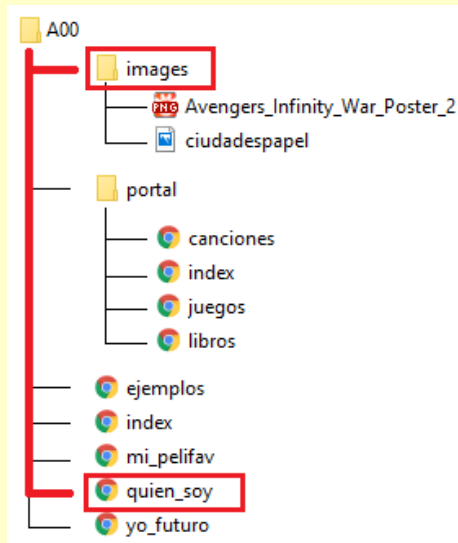
se verá como ...

[Visita el sitio de innovacien](http://innovacien.org)

y al seleccionar el link te llevará directamente a la página de innovacien (innovacien.org).

## REFERENCIANDO ARCHIVOS LOCALES

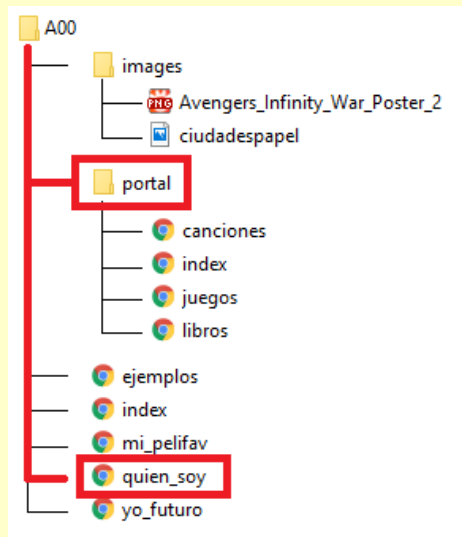
- Al igual que con las imágenes, en el caso de vincular páginas del mismo sitio se debe copiar la ruta respecto a la página origen, en cambio al vincular una página externa hay que copiar TODA la dirección (URL)
- Ejemplo: si tengo la siguiente estructura de archivos en mi sitio, veremos cómo acceder a diferentes archivos usando las referencias locales...



- Si queremos acceder a la imagen `ciudadespapel.jpg` desde el archivo `quien_soy.html` debemos entrar a la carpeta `images` y ahí encontraremos el archivo, no es necesario buscar más ya que el archivo `quien_soy.html` y la carpeta `images` están en la misma carpeta.

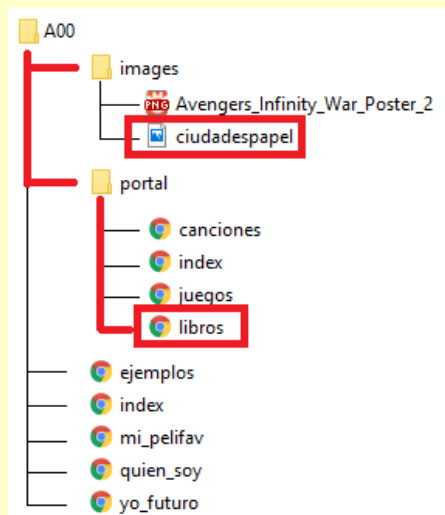
Usaremos `<IMG SRC="images/ciudadespapel.jpg">`

- Recuerda que para las etiquetas no importa si escribes con mayúscula o minúscula, pero los nombres de archivos o carpetas deben escribirse tal como son.



- Si queremos vincular el archivo **libros.html** desde el archivo **quien\_soy.html** deberemos indicar sólo la carpeta **portal**, ya que está en la misma carpeta que el archivo de origen.

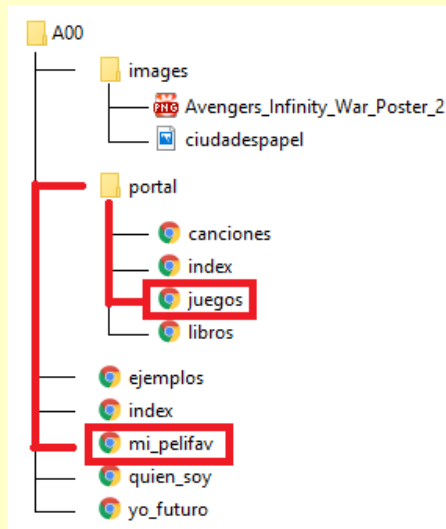
Usaremos `<a href="portal/libros.html"> libros </a>`



- Si queremos incluir la imagen **ciudadespapel.jpg** en el archivo **libros.html** deberemos subir una carpeta con el comando `../` y luego entrar a la carpeta `images`.

Usaremos ``

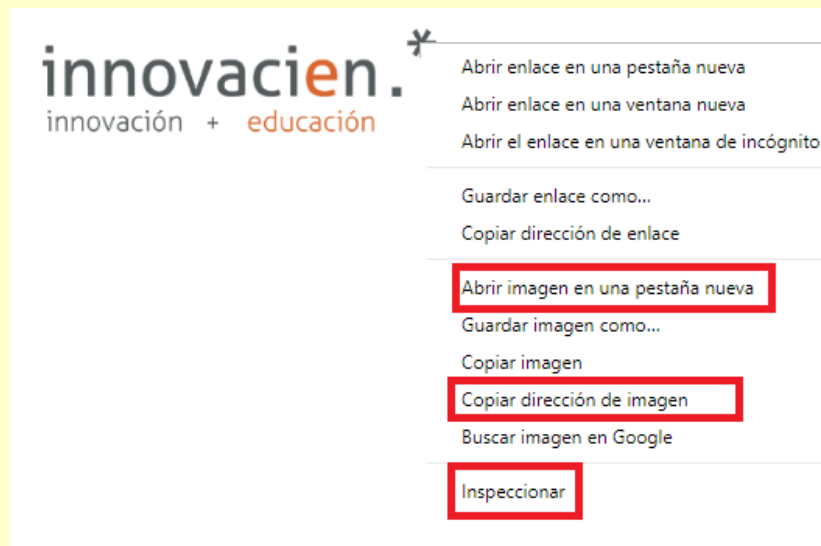
- Por último para vincular el archivo `mi_pelifav.html` desde `juegos.html` también deberemos subir una carpeta.



Usaremos `<a href=\"../mi_pelifav.html> Mi película </a>`

- **Todas las referencias son respecto a la ubicación del archivo origen, es decir, del archivo donde queremos insertar la imagen o el vínculo.**

## REFERENCIANDO ARCHIVOS EXTERNOS



- Para referenciar cualquier objeto que está en otro servidor se debe usar la **dirección completa del objeto** (imagen, página, documento, etc), es decir, todo lo que aparece escrito en la barra de direcciones del navegador, desde **http://** hasta el final.

- En el caso de las imágenes se puede usar “**copiar dirección de imagen**”, “**abrir imagen en una pestaña nueva**” y en algunos casos y con un poco más de experiencia “**Inspeccionar**” que se obtienen al presionar Botón derecho” sobre la imagen deseada.
- Todas las opciones corresponden a Chrome, si Ud usa otro navegador, puede haber ligeros cambios en la redacción de las opciones.

## VOLVAMOS A LAS MATEMÁTICAS...

### PROPORCIONES Y PORCENTAJES

El tratamiento del tamaño de las imágenes en Html se realiza en base a dos propiedades básicas, WIDTH y HEIGHT (Ancho y Altura), las cuales pueden recibir como atributos el ancho y la altura tanto en PÍXELES (Puntos de la pantalla) como en PORCENTAJES DEL TAMAÑO ORIGINAL.

Por ejemplo, si una imagen tiene tamaño original en pixeles de  $1280 \times 480$  y necesitamos ubicarla en la página en un espacio no superior a 400 pixeles de ancho, podemos usar el atributo `WIDTH="400PX"`. No es necesario definir ancho y alto a la vez, ya que el navegador realiza en forma automática el escalamiento proporcional de la imagen.

Problema: ¿Cuál será la altura de la imagen escalada?

Usando proporciones  $1280:480 = 400:x$

Aplicando regla de 3 simple tenemos  $x = \frac{400 \times 480}{1280} = 150$

Entonces la altura de la imagen será 150 pixeles.

¿Por qué debo saber calcular esto si el navegador escala solo al ingresar sólo el ancho?

Simple... si el espacio disponible es de  $400 \times 100$  podrás definir antes de programar tu página el tamaño óptimo, en este caso  $267 \times 100$  pixeles.

Usando proporciones  $1280:480 = x:100$

Aplicando regla de 3 simple tenemos  $x = \frac{1280 \times 100}{480} = 266,\bar{6} \cong 267$

Las medidas también pueden ingresarse en forma porcentual respecto al tamaño original, por ejemplo, queremos reducir todas las imágenes al 40%, en nuestra imagen ejemplo tendremos

40% de 1280 = 512

40% de 480 = 120

Los porcentajes también nos ayudan a definir los tamaños de las secciones de nuestra página web, ya sea que utilicemos "tablas" o "div".

Por ejemplo, al definir dos tablas con las medidas

```
<table width="80%" border="1" bordercolor="red">
<tr><td>
<table width="60%" border="1" bordercolor="blue">
<tr>
<td width="40%"> 40% </td>
<td width="35%"> 35% </td>
```

```
<td> El resto 25% </td>  
</tr>  
</table>  
</td></tr>  
</table>
```

40%	35%	El resto 25%
-----	-----	-----------------

La primera tabla (color rojo) ocupa el 80% del espacio disponible. La segunda tabla ocupa el 60% del espacio ocupado por la Tabla 1, ya que se encuentra dentro de ella.

Las celdas de la tabla 2 ocupan respectivamente el 40%, el 35% y el 25% del espacio ocupado por la tabla 2, es decir, en la hoja principal ocupan el 32% (40% de 80%), 28% (35% de 80%) y 20% (25% de 80%)

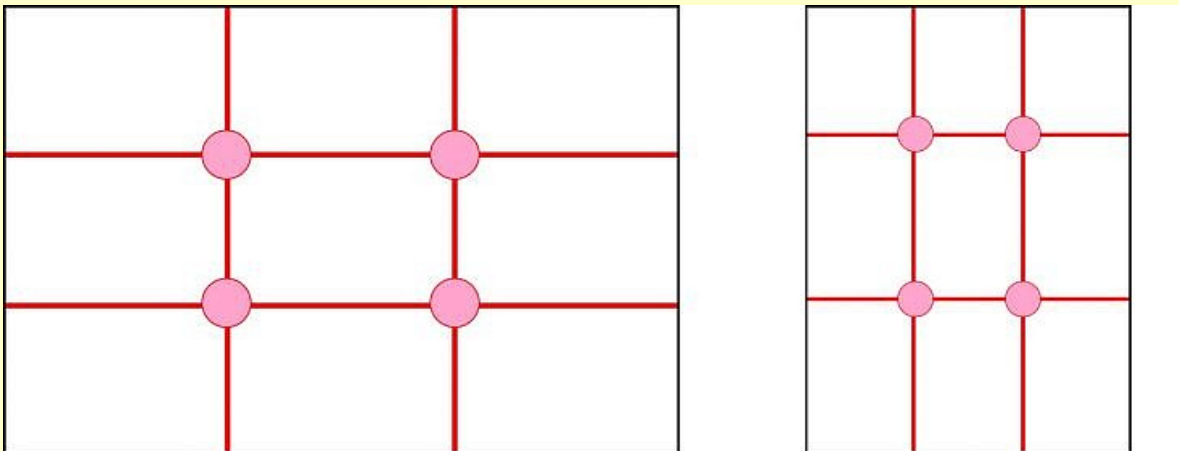
Problema: Si la página principal tiene 1280 píxeles de ancho, ¿Cuál es el ancho de cada celda, en píxeles?

## REGLA DE LOS TERCIOS

La regla de los tercios es una de las reglas de composición más famosas dentro del mundo de la fotografía. Es uno de los primeros recursos compositivos que uno descubre y probablemente sea de los primeros en aprenderse tanto por su sencillez como por lo efectivo que es en sus resultados.

Con la regla de los tercios conseguirás dotar a tus fotografías de una sensación de profundidad y lograrás que se aprecie un mayor equilibrio, guiando al ojo del espectador directamente al punto de mayor interés.

Mira una de tus fotografías y mentalmente divídela en tres tercios imaginarios, tanto horizontales como verticales. La imagen te quedará dividida en 9 partes iguales, y los cuatro puntos de intersección de esas líneas son los que van a fijar los puntos adecuados para situar el centro, o centros, de interés de nuestra foto. De esa forma estaremos rompiendo con la común costumbre de situar a nuestro sujeto o elemento protagonista justo en el centro del encuadre.

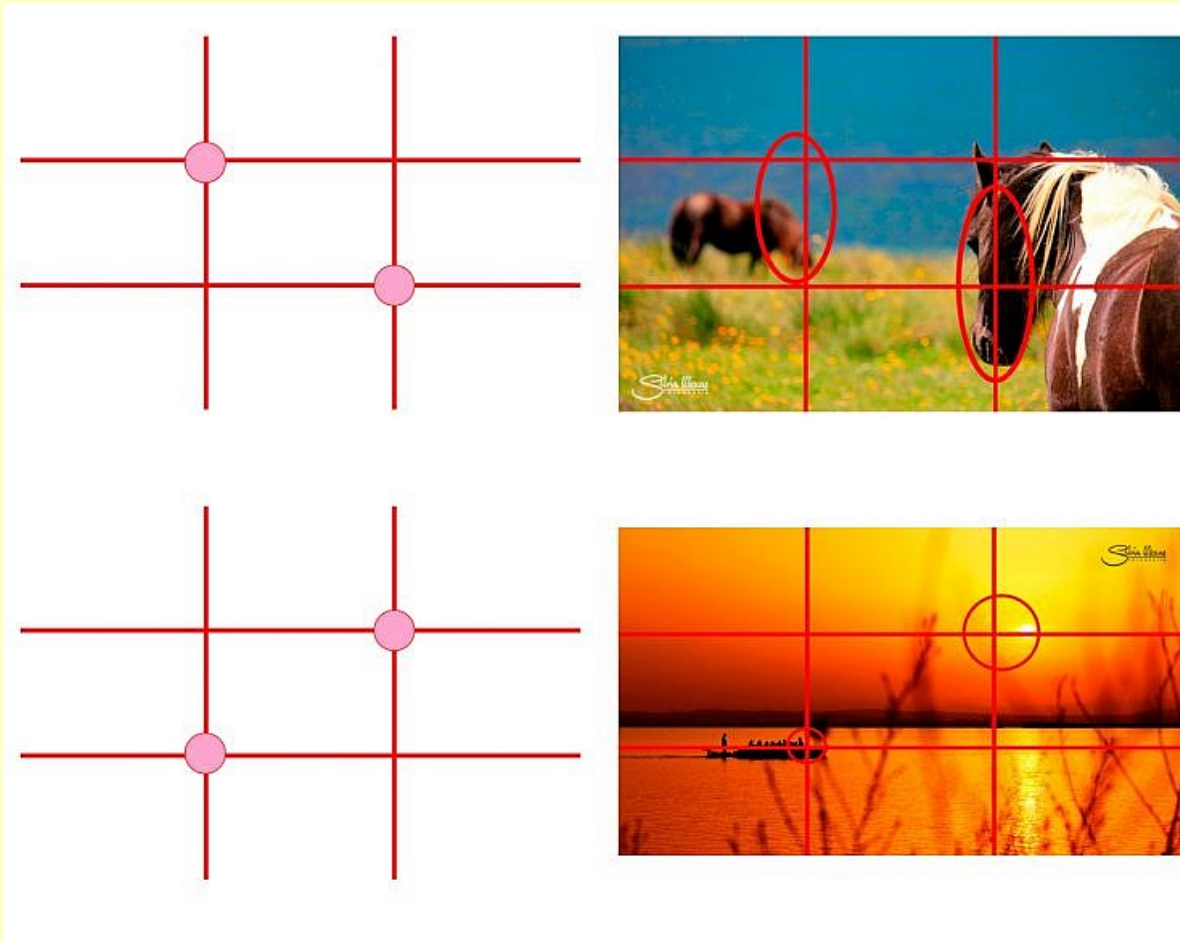


Cada uno de estos cuatro puntos de intersección se denominan puntos fuertes. Cuando estemos haciendo una foto, si solo existe un único punto de interés, es preferible situarlo en uno de los cuatro puntos de intersección mencionados en lugar de hacerlo directamente en el centro de la foto. Esto suele generar mayor atracción en el espectador que cuando el centro de interés está en el mismo centro de la fotografía.

### ¿Y Si Tengo Varios Centros de Interés?

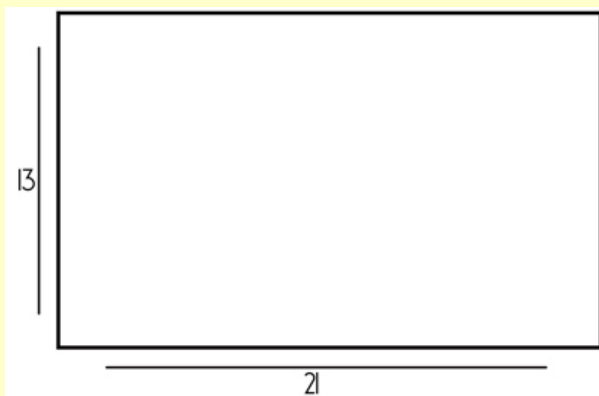
En caso de que en la foto hubiera dos centros de interés, es recomendable buscar dos de estos puntos, y siempre que sea posible opuestos, formando una diagonal en la foto. Veámoslo mejor con un ejemplo.



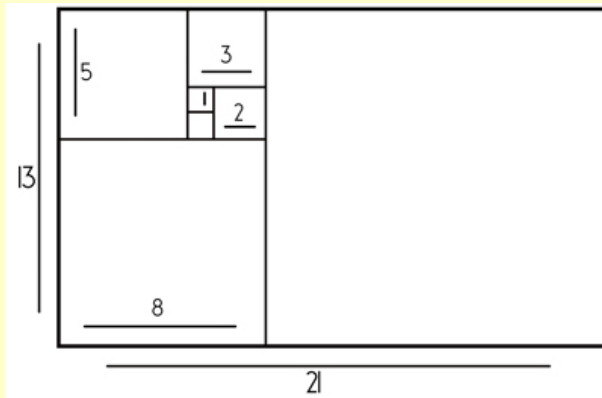


## LA PROPORCIÓN ÁUREA EXPLICADA CON IMÁGENES

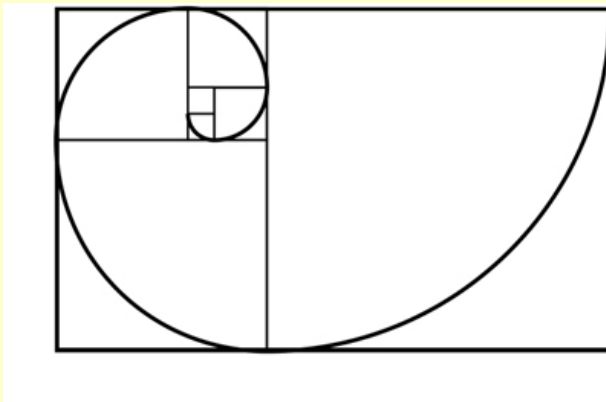
Bien, pues apliquemos la Proporción Aurea al mundo visual. Creemos un rectángulo cuyos lados midan dos de los números de la sucesión de Fibonacci (1 1 2 3 5 8 13 21 ... - cada número es la suma de los dos anteriores- )



Y ahora vamos a dividirlo siguiendo la sucesión numérica:



Si dibujamos una línea que una todos estos pequeños recuadros, quedaría algo parecido a esto:



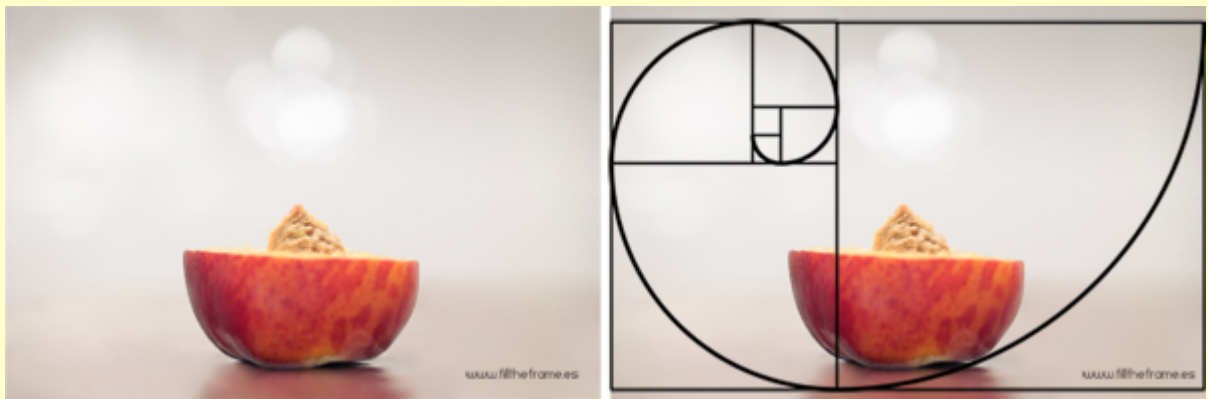
La espiral resultante (conocida como Espiral de Oro o Espiral Aurea) está permanentemente presente en la naturaleza: en las semillas de un girasol, en las conchas marinas... Componer una imagen siguiendo esta espiral nos resulta agradable visualmente porque las proporciones que se obtienen nos parecen naturales.



Es importante tener en cuenta que las fotografías no acostumbran a tener unas proporciones áureas (y si se trata de cámaras de medio formato, cuyos sensores suelen ser cuadrados, todavía menos) así que la espiral de Fibonacci debe ser sólo una guía que te ayude a componer y nunca una regla intocable que te cierre puertas creativas. También

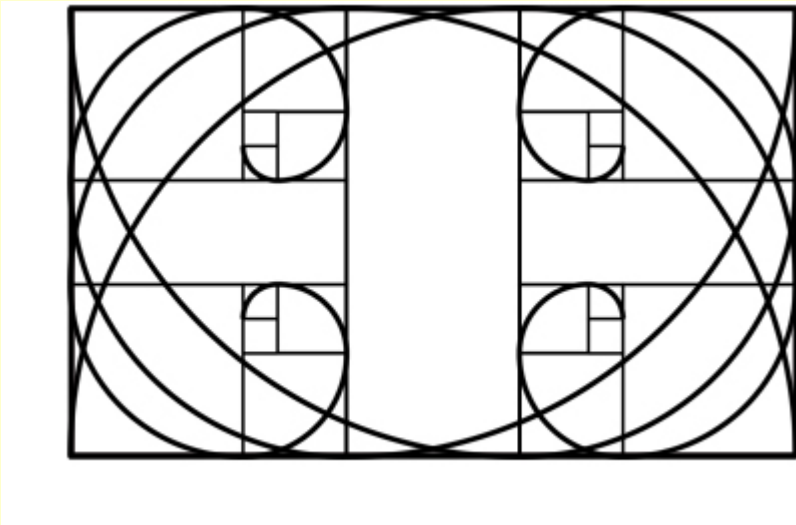
es cierto que muchas veces, componemos según la espiral de oro sin ser conscientes de ello, simplemente porque la composición que hemos creado nos ha parecido atractiva visualmente. La imagen anterior es un ejemplo de ello, pues la imagen se tomó y luego, para la redacción de este artículo, se le añadió la espiral, momento en el que me di cuenta de que había seguido la proporción áurea en su composición, sin ni siquiera percatarme de ello.

Siguiendo la proporción áurea puedes tener una idea de dónde situar el horizonte o los puntos más importantes de tu fotografía. Lo importante es ser consciente de que no es una ley que se deba cumplir a rajatabla y de que en absoluto asegura la calidad de la imagen final. A veces puede salir una fotografía más atractiva visualmente rompiendo esta regla que siguiéndola, todo es cuestión de probar. Un claro ejemplo de ello son las imágenes simétricas.

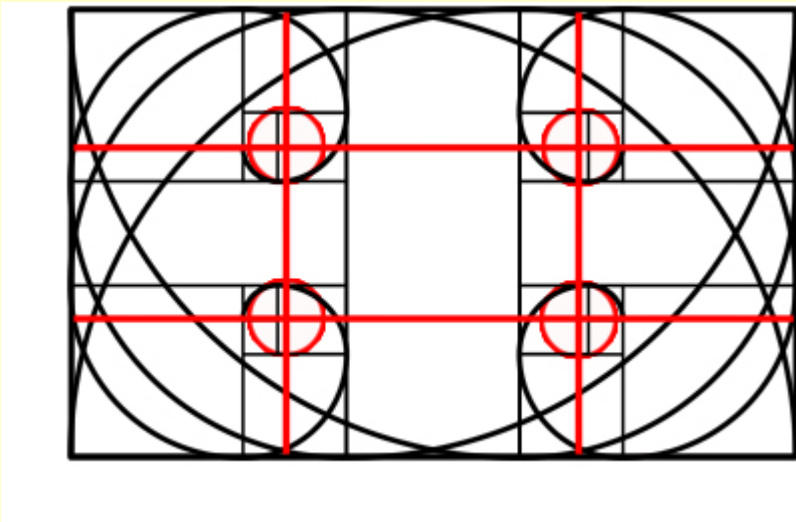


### La Relación entre la Regla de los Tercios y la Proporción Áurea

Ya habrás imaginado que todo esto te lo estoy contando con un fin y que, seguramente, ese fin sea llegar al origen de la Regla de los Tercios. Bien, pues ha llegado ese momento. Seguimos con el cuadro que hemos utilizado anteriormente. Lo que hacemos ahora es colocar cuatro espirales en el mismo rectángulo. Colocándolas de manera que se inicie una espiral en cada una de las cuatro esquinas del recuadro:



Qué, ¿te suena lo que ves? Vamos a marcar en rojo el centro de las espirales:



Señoras y señores, ¡aquí tenemos nuestra querida Regla de los Tercios! Así que ha quedado demostrado que las matemáticas nos ayudan a componer fotográficamente. Con este dibujo, además, se ven de manera muy gráfica y evidente, las zonas con más interés visual: las esquinas. Como se puede apreciar en el esquema, el centro de la imagen es la zona "menos interesante" de un encuadre (hablamos en general; como ya hemos descrito anteriormente, existen fotografías que rompen por completo esta concepción y, aún así, son muy atractivas visualmente.)

Como has visto, la regla de los tercios es una versión de la proporción áurea; en general, resulta más sencillo componer una fotografía con la regla de los tercios en mente (o superpuesta en la pantalla de la cámara a modo de guía) que con la Espiral de Oro.

## SIMETRÍA ASIMETRÍA

Tomado desde <https://www.fotonostra.com/grafico/equilibriosimetrico.htm>

Equilibrio simétrico: es cuando al dividir una composición en dos partes iguales, existe igualdad de peso en ambos lados. No se encuentran elementos que sobresalgan más que el resto en importancia y peso.

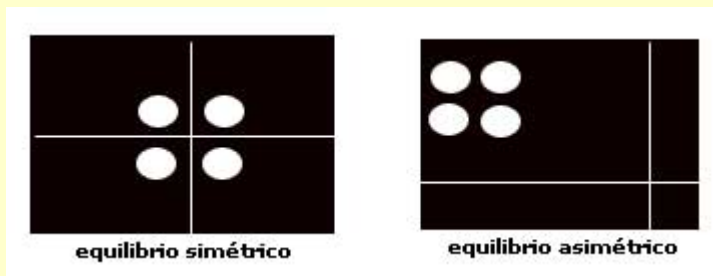
Un ejemplo más inmediato de simetría en el mundo orgánico es la mariposa, cuyas alas poseen una simetría axial bilateral en la que el eje es el cuerpo del insecto. Sus dibujos están dispuestos simétricamente respecto al eje. Esta regularidad constituye, a nivel de percepción, un factor estético de armonía.

La creación de un diseño simétrico nos transmite una sensación de orden.

Un equilibrio es asimétrico, cuando al dividir una composición en dos partes iguales no existen las mismas dimensiones en tamaño, color y peso, pero existe un equilibrio entre los elementos.

En el equilibrio asimétrico al ser desiguales los pesos a un lado y otro del eje, el efecto es variado.

La asimetría nos transmite agitación, tensión, dinamismo, alegría y vitalidad; en este tipo de equilibrio una masa grande cerca del centro se equilibra por otra pequeña alejada del aquel.



Estos principios se aplican al diseño en general y al diseño de páginas web en particular.

## USO DE CSS

Como vimos anteriormente, las etiquetas de HTML tienen ATRIBUTOS que le permiten al programador cambiar ciertas propiedades, ya sea respecto a la información contenida o al diseño de la presentación en pantalla.

Por ejemplo

**<H1 align="center"> Hola Mundo </H1>** permitirá insertar un encabezado centrado con el texto Hola Mundo. En este caso ALIGN (alineación) es un atributo al que se le ha asignado el valor "center" (centrado).

**<IMG src="imagen1.jpg" width="100px">** insertará una imagen desde el archivo imagen1.jpg con ancho 100 pixeles. En este caso SRC (archivo fuente) es un atributo al que se le ha asignado el valor "imagen1.jpg" (nombre del archivo) y WIDTH (ancho) es otro atributo, al que se le ha asignado el valor "100px" (100 pixeles)

De este modo es posible asignar propiedades a casi todas las etiquetas de un documento html, cambiando colores, tipos de letra, alineaciones y un sinnúmero de propiedades, según el tipo de etiqueta que se trate.

Pero si tenemos, por ejemplo, muchos párrafos en lugares diferentes de un documento (e incluso en diferentes documentos) y queremos dar a todos el mismo formato, con html deberíamos modificar individualmente cada etiqueta, lo cual se traduce en un trabajo tedioso y repetitivo.

***¿No sería mejor poder cambiar en un solo lugar las propiedades de todos los párrafos a la vez, o poder definir el color y tipo de letra de todos los títulos de un sitio con sólo una modificación?***

CSS viene a responder a esta inquietud, permitiendo separar el contenido de un documento de su diseño, al tiempo que amplía las posibilidades de diseño propias de html.

El código CSS se puede insertar en la etiqueta HEAD de un documento o bien dejar en un documento adicional y vincularlo con una o más páginas de un mismo sitio.

## ESTRUCTURA DE INSTRUCCIONES CSS

Las instrucciones de CSS tienen la siguiente estructura

```

OBJETO(S) {
  ATRIBUTO      : VALOR O VALORES;
  ATRIBUTO2     : VALOR O VALORES;
  :
  ATRIBUTO_n   : VALOR O VALORES;
}
  
```

Donde OBJETO puede ser una etiqueta, un grupo de etiquetas, una clase, u identificador, cada una en forma simple o condicionada, como veremos a continuación.

## TIPOS DE OBJETOS

- ETIQUETAS:

Se puede elegir TODAS las etiquetas

```
* {
}
```

Se puede elegir sólo una etiqueta

```
H1 {
}
```

Se puede elegir varias etiquetas a la vez

```
H1, H2, H3 {
}
```

En este caso se aplican las propiedades a todas las etiquetas H1, H2 y H3

Se puede elegir ALGUNAS etiquetas, que cumplan una condición

```
P SPAN {
}
```

En este caso se aplican las propiedades a todas las etiquetas <SPAN> que se encuentren DENTRO de una etiqueta <P>. Observe que NO LLEVA COMA.

Ejemplo

```
<p> <span> Texto1 ... </span> </p>
<h1> <span> Texto2 ... </span> </h1>
```

Sólo se aplicará la propiedad al Texto1, ya que el Texto2 se encuentra dentro de la etiqueta H1, no dentro de P.

La etiqueta SPAN es un separador para definir propiedades de texto.

- CLASES: Corresponde a categorías asignadas por el diseñador a etiquetas o grupos de etiquetas, por ejemplo

```
<p class="destacado"> texto </p> o <td class="destacado"> texto celda </td>
```

Asigna al texto la clase "destacado", de forma de diferenciarlo de otros textos que también tienen la etiqueta <p> o <td>. Una misma clase se puede usar en varios lugares dentro de un documento html.

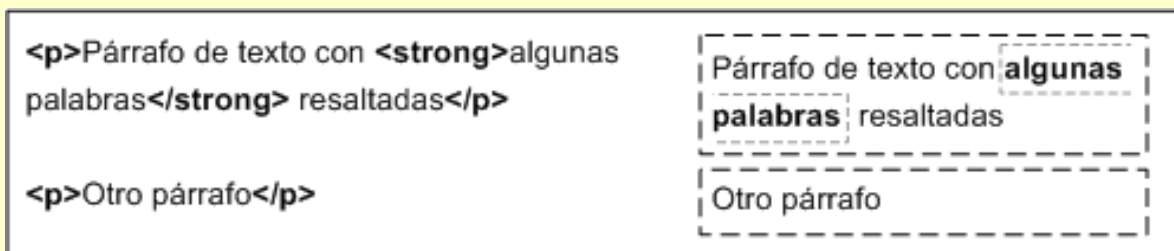
Para aplicarle propiedades en css se usa

```
.destacado {
}
```

También se pueden combinar del mismo modo que las etiquetas. Por ejemplo **p.destacado** se refiere sólo a los párrafos con clase destacado, dejando de lado otras etiquetas que pudieran tener la misma clase.

CSS define para cada objeto HTML, una caja o contenedor que “contiene” el objeto y sobre el cual se puede definir propiedades de diagramación (márgenes, bordes, espaciados, etc.)

Por ejemplo



Si bien estas “cajas” en principio no son visibles, el diseñador puede cambiar sus propiedades, cambiando la forma como se relacionan con los otros objetos.

## PROPIEDADES DE CAJAS

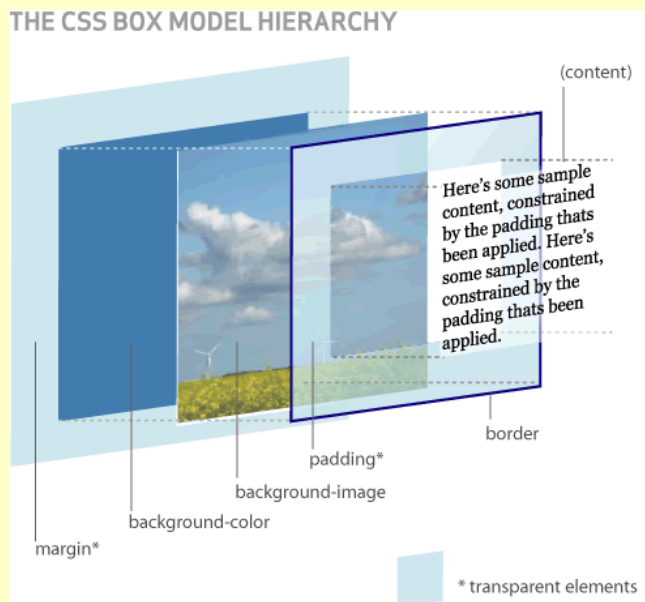
Las partes que componen cada caja y su orden de visualización desde el punto de vista del usuario son las siguientes:

Contenido (content): se trata del contenido HTML del elemento (las palabras de un párrafo, una imagen, el texto de una lista de elementos, etc.)

Relleno (padding): espacio libre opcional existente entre el contenido y el borde.

Borde (border): línea que encierra completamente el contenido y su relleno.

Imagen de fondo (background image): imagen que se muestra por detrás del contenido y el espacio de relleno.





Color de fondo (background color): color que se muestra por detrás del contenido y el espacio de relleno.

Margen (margin): separación opcional existente entre la caja y el resto de cajas adyacentes.

WIDTH: Define el ancho de los objetos.

HEIGHT: Define la altura de los objetos.

MARGIN, MARGIN-TOP, MARGIN-RIGHT, MARGIN-BOTTOM, MARGIN-LEFT : define los márgenes TOP (superior), RIGHT (derecho), BOTTOM (inferior), LEFT (izquierdo), MARGIN define todos los márgenes a la vez. Los márgenes se refieren a la distancia de la caja con los otros objetos.

PADDING, PADDING-TOP, PADDING-RIGHT, PADDING-BOTTOM, PADDING-LEFT: Define la distancia INTERIOR entre los bordes de la caja y su contenido.

BORDER-TOP-WIDTH, BORDER-RIGHT-WIDTH, BORDER-BOTTOM-WIDTH, BORDER-LEFT-WIDTH: Define el ancho del borde.

BORDER-TOP-COLOR, BORDER-RIGHT-COLOR, BORDER-BOTTOM-COLOR, BORDER-LEFT-COLOR: Define el color de los bordes.

BORDER-TOP-STYLE, BORDER-RIGHT-STYLE, BORDER-BOTTOM-STYLE, BORDER-LEFT-STYLE: define el Estilo de los bordes, según los siguientes valores posibles, cuyo resultado depende de los navegadores.

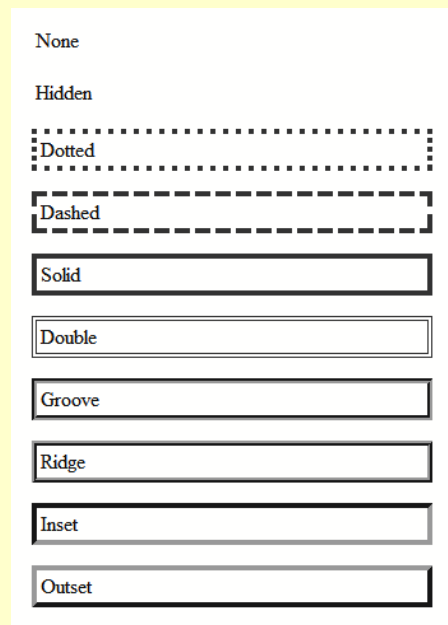
BACKGROUND-COLOR: Define el color de fondo.

BACKGROUND-IMAGE: Define una imagen de fondo.

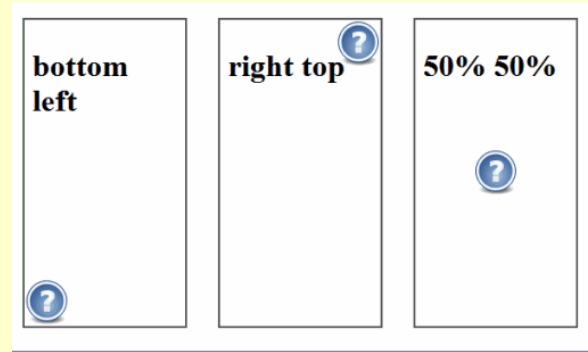
BACKGROUND-REPEAT: Repite la imagen de fondo dentro de una caja, valores posibles: repeat | repeat-x | repeat-y | no-repeat.

BACKGROUND-POSITION: define la posición de la imagen dentro de la caja.

Ejemplo de Imágenes en Background



```
#caja1 {  
  background-image:  
  url("images/help.png");  
  background-repeat: no-repeat;  
  background-position: bottom left;  
}  
#caja2 {  
  background-image:  
  url("images/help.png");  
  background-repeat: no-repeat;  
  background-position: right top;  
}  
#caja3 {  
  background-image: url("images/help.png");  
  background-repeat: no-repeat;  
  background-position: 50% 50%;  
}
```



```
<div id="caja1"><h1>bottom left</h1></div>  
<div id="caja2"><h1>right top</h1></div>  
<div id="caja3"><h1>50% 50%</h1></div>
```

## PROPIEDADES DE TEXTO

color: Define el color del texto

font-family: define las familias de fuente a utilizar en el texto, por ejemplo

**font-family: Arial, Helvetica, sans-serif;**

**font-family: "Times New Roman", Times, serif;**

**font-family: "Courier New", Courier, monospace;**

**font-family: Georgia, "Times New Roman", Times, serif;**

**font-family: Verdana, Arial, Helvetica, sans-serif;**

font-size: define el tamaño de la fuente

font-weight: define el grosor de la fuente (normal | bold | bolder | lighter | 100 | 200 | 300  
| 400 | 500 | 600 | 700 | 800 | 900 )

font-style: Define el estilo de la fuente (normal | italic | oblique)

## EL LENGUAJE JAVASCRIPT

JavaScript es un lenguaje de programación orientado a dar a los sitios web características de interacción con el visitante, permitiendo modificar características de diseño y contenido a partir de las acciones realizadas por el visitante de la página.

Las instrucciones de JavaScript se insertan en la página web, utilizando la etiqueta <script>, según si escribiremos el código directamente en la página o si lo leeremos desde un archivo externo.

Si lo leemos desde un archivo externo (recomendado cuando tenemos una tarea o función que nos sirve para varias páginas).

```
<script src="main.js"></script>
```

En este caso estamos cargando el archivo main.js, que contiene código javascript.

Si queremos insertar directamente el código en nuestra página usamos

```
<script language="javascript" type="text/javascript">  
Codigo JavaScript  
</script>
```

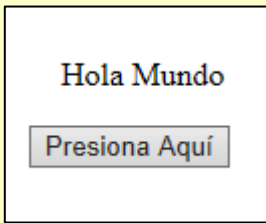
## CAMBIANDO PROPIEDADES CSS USANDO JAVASCRIPT.

Al definir cada objeto HTML, podemos darle un identificador "id", usando este identificador podemos acceder al objeto html usando la instrucción document.getElementById("id") y el parámetro .style o .value, podemos cambiar propiedades del objeto.

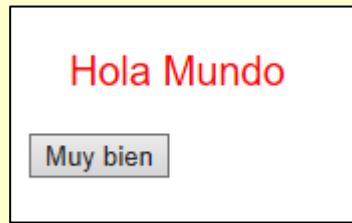
Por ejemplo, si definimos un párrafo con el id "encabezado", podemos cambiar el tamaño, color y fuente del texto...

```
<p id="encabezado"> Hola Mundo </p>  
<script>  
function texto(){  
document.getElementById("encabezado").style.color="red";  
document.getElementById("encabezado").style.fontFamily="Arial";  
document.getElementById("encabezado").style.fontSize="15pt";  
document.getElementById("boton1").value="Muy bien";  
}  
</script>  
<input id="boton1" type="button" value="Presiona Aquí" onclick="texto()">
```

Antes



Después



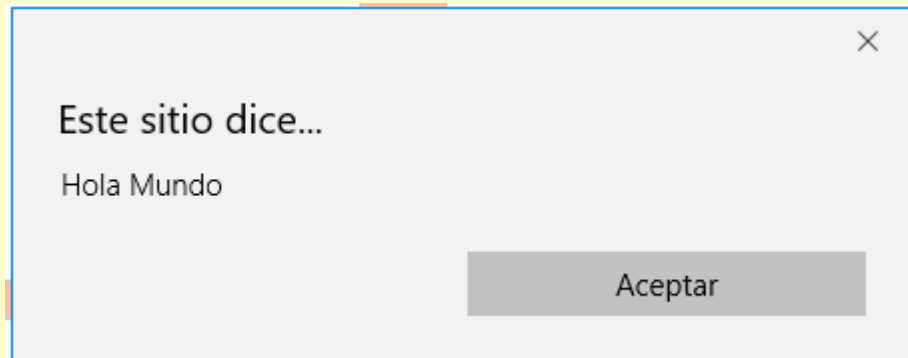
Del mismo modo podemos cambiar los atributos de cualquier objeto html. Para cambiar contenido de párrafos, encabezados u otros elementos HTML, usamos la propiedad **innerHTML**.

## INTERACTUANDO CON EL USUARIO USANDO JAVASCRIPT

- Mensajes de Alerta al usuario:

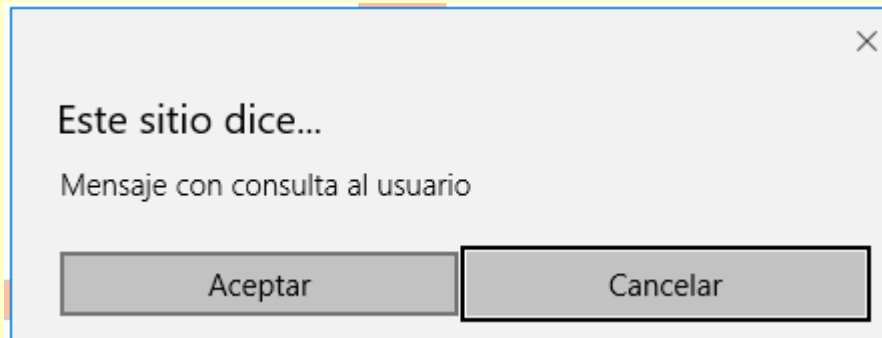
```
alert("mensaje");
```

Ejemplo : `alert("Hola Mundo");`



- Mensajes de alerta con decisiones:

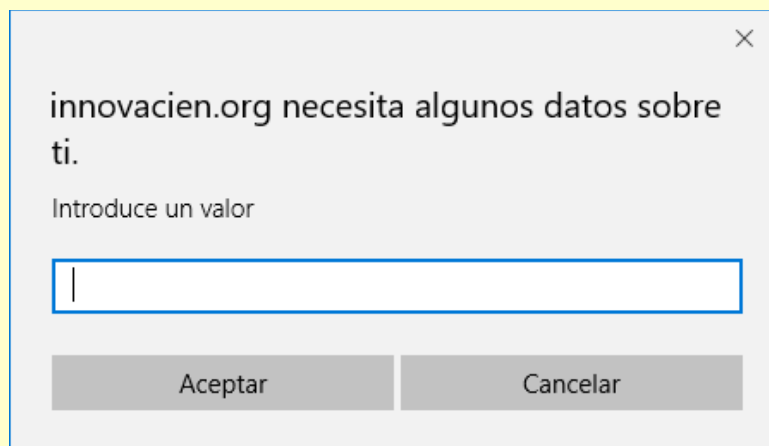
```
var respuesta=confirm("Mensaje con consulta al usuario");  
    if (respuesta){  
        INSTRUCCIONES SI SE ACEPTA  
    }  
    else  
    INSTRUCCIONES SI SE CANCELA  
}
```



Según el botón que presione el usuario serán las acciones a seguir.

- Mensajes con consulta de datos

```
var valor_ingresado = prompt("Introduce un valor");
```



Pide un valor al usuario, que puede ser de tipo numérico o texto y lo guarda en la variable "valor\_ingresado", después se puede operar con esta variable en forma habitual.

## ENTORNO GRÁFICO

HTML permite definir un lienzo gráfico el cual puede ser modificado utilizando una serie de funciones javascript. De esta forma podemos trabajar en un entorno gráfico que nos permita dibujar líneas, figuras geométricas predefinidas, desplegar textos con variados efectos. Pero lo principal es que podemos interactuar en forma dinámica con estos elementos, generando una experiencia en el usuario que no es posible con una página estática.

Al combinar en forma conveniente los elementos podemos generar, por ejemplo, juegos, aplicaciones web orientadas a la entretención, al aprendizaje o a la solución de algún problema específico. En pocas palabras, podemos utilizar la tecnología para la solución directa de nuestros problemas, adecuándose a nuestras necesidades, no siendo meros consumidores, sino potenciales creadores de estas soluciones.

## PROYECTO: JUEGO EDUCATIVO

Tema:

Números: Múltiplos de 2, 3, 4, 5, 6, 10 (5to Básico)

(El tema es referencial, puede ser adaptado fácilmente por el docente)

Objetivo del Juego: Discriminar aquellos elementos que cumplen una condición dada.

Instrucciones: Debes recibir en la canasta todas aquellas fichas que cumplan con la condición pedida.

### EL CÓDIGO HTML

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml"> <!-- Define que estamos en un documento html -->
<head> <!-- encabezado de la página, define opciones generales de formato y contenido -->
<meta http-equiv="Content-Type" content="text/html; charset=iso-8859-1" /> <!-- Set de caracteres (símbolos
posibles de usar en la página, en este caso, alfabeto occidental) -->
<title>Juego Educativo</title> <!-- Titulo de la página (Lo que se ve en la parte superior en la pestaña del navegador -->
</head>
<body> <!-- Cuerpo de la página, acá se ingresa el contenido -->
<h1 align="center"> Jugando con los Números </h1>
<!-- Encabezados. El código &uacute; representa la u acentuada (lo mismo para &aacute; = á , &eacute;=é, etc)-->
<H2 id="nomb_jugador" align="center"> </h2>
<H3 id="text_preg" align="center"> </h2>
<div class="fondo">
<!-- Cada DIV define una caja en la pantalla (región independiente para ubicar información) caja : fondo-->
<div class="instr"> <!-- Caja Instrucciones -->
<p> Observa la pregunta que aparece sobre el tablero de juego. </br> Debes guardar dentro de tu canasta <b>
Solo las figuras que cumplen la condici&uacute;n dada en la pregunta. </b></br></br>
<!-- p: párrafo - br: salto de línea dentro de un párrafo -->
Tienes inicialmente 3 vidas.
</br></br>
Mientras m&aacute;s figuras captures, mayor ser&aacute; tu puntaje</p>
<input class="tipo_preg" type="radio" id="tipo_preg" name="tipo_preg" value="1"
```

```

onclick="mostrar('numerica')"> Numérica </br> <!-- input radio : Casillas de seleccion única -->
<input class="tipo_preg" type="radio" id="tipo_preg" name="tipo_preg" value="2"
onclick="mostrar('conceptual')">Conceptual </br> <!-- al hacer click llaman la funcion mostrar, que define cual DIV se
muestra en pantalla -->
<div class="jugando" id="jugando">
<input class="ptje" id="ptje" type="text" value="0" readonly /> <!-- Cuadros de texto, no editables -->
<input class="vidas" id="vidas" type="text" value="3" readonly />
<input class="btn" type="button" value="Iniciar Juego" onclick="jugar()"/> <!-- botón, al hacer click llama la función
"jugar" -->
</div>
<div class="conceptual" id="conceptual">
<select id="sel_concep" name="sel_concep"> <!-- Menu desplegable, para elegir una opción -->
<option value= "1" selected> Palabras Agudas </option>
<option value="2"> Palabras Graves </option>
<option value="3"> Capitales de América </option>
<option value="4">Capitales de Europa </option>
<option value="5"> Regiones de Chile </option>
</select>
<input class="btn" id="btn_acep" type="button" value="Aceptar" onclick="mostrar('jugando')"/>
<!-- Boton para confirmar selección -->
</div>
<div class="numerica" id="numerica">
<select id="sel_math" name="sel_math">
<option value="11" selected> Pares </option>
<option value= "12"> Impares </option>
<option value= "13"> Primos </option>
<option value= "14"> Divisores del 100 </option>
<option value= "15"> Múltiplos de 3 </option>
</select>
<input class="btn" id="btn_acep" type="button" value="Aceptar" onclick="mostrar('jugando')"/>

</div>
</div>
<div class="game">
<p class="pregunta"> <p>
<p>
<canvas id="mi_canvas" class="t_game" ></canvas> <!-- Define el lienzo de trabajo gráfico para la página web -->
</p>

```



```

</div>
</div>
</body>
</html>

```

## EL DISEÑO CON CSS

En el código HTML se incluyeron varias definiciones de clases, esto es para poder configurar su diseño utilizando CSS.

```

<style> /* Inicia CSS */

.fondo { /* clase fondo */
display: table; /* Define el fondo como varias columnas, permite que las cajas se ordenen en forma horizontal */
width: 100%;
height:500px;
}

.fondo div {
display: table-cell;
}

.instr { /* clase instr */
position:relative; /* Posición respecto a la caja donde está contenida */
padding:20px; /* Borde interior de la caja */
background-color: orange; /* Color de fondo de la caja */
width: 20%; /* Ancho de la caja */
vertical-align:top; /* Alineación vertical de la caja */
height:100%; /* Altura de la caja */
}

.game { /* clase game */
position:relative;
padding:20px;
background-color: skyblue;
width: 80%;
padding: 15px;
height:100%
}

.btn { /* clase btn */
position:absolute;
bottom:10px; /* Distancia desde el borde inferior de la caja */
left:35%;
}

```

```

.t_game{                               /* clase t_game */
width:90%;
height:80%;
border:6px red solid;                 /* borde de la caja */
align:center;
}
.ptje {                                /* clase ptje */
left:25%;
text-align:right;                     /* Alineación del texto */
}
.jugando{                              /* clase jugando */
position:absolute;
bottom:100px;
visibility:hidden;                   /* Caja Oculta */
height:200px;
}
.conceptual{                           /* clase conceptual */
position:absolute;
bottom:100px;
visibility:hidden;
height:200px;
}
.numerica {                            /* clase numerica */
position:absolute;
bottom:100px;
visibility:hidden;
height:200px;
}
</style>

```

## LAS FUNCIONES JAVASCRIPT

Las funciones JavaScript nos permiten dar la interactividad a nuestra página, ya sea modificando el formato visual (mostrar u ocultar cajas) como interactuar con el usuario a todo nivel.

```

<script type="text/javascript"> //Inicia Javascript
function mostrar (id_div){ //Muestra u oculta cajas en la pantalla
document.getElementById("numerica").style.visibility="hidden";
document.getElementById("conceptual").style.visibility="hidden";
document.getElementById("jugando").style.visibility="hidden";

document.getElementById(id_div).style.visibility="visible";
}

```

```

function jugar(){ //Función principal del juego
var correcta=true; //inicializa variable "Correcta" y le asigna valor "true" (verdadero)
var txt_opcion; //define variable txt_opcion

//Todas las funciones siguientes están definidas dentro de la función jugar.

function genera_pregunta(){ //Permite asignar las respuestas que entregará el juego, para que el alumno decida si son
correctas o incorrectas

var t_preg=document.getElementById("tipo_preg"); //carga en un listado las opciones (numérica / conceptual
var n_opciones, n_correctas; //define las variables que guardarán el nro de opciones de esta pregunta y el nro
de opciones correctas asignadas.
var i
    for (i=0;i<tipo_preg.length;i++){
        if (tipo_preg[i].checked) //lee el tipo de pregunta elegida por el usuario en OPTION (0 / 1)
            break;
    }
var nro_preg=0;
if (i==0){
nro_preg=document.getElementById("sel_math").value;
}else{
nro_preg=document.getElementById("sel_concep").value;
} //lee la opcion elegida por el usuario en SELECT
var opciones; //Listado donde guardaremos las opciones de respuesta (correctas e incorrectas)
switch(nro_preg){ //Definición individual de cada pregunta, con sus opciones
case "1": //Ejemplo pregunta conceptual
opciones=["canción","japonés","acatar","acceder","borrar","corazón","reloj","auto",
"casa","pauta","camino","sala","árbol"];
n_opciones=13;
n_correctas=7;
break;
case "11": //Ejemplo pregunta numérica
n_opciones=50;
n_correctas=25;
for (i=0;i<25;i++){
opciones[i]=2*i;
opciones[n_correctas+i]=2*i+1;
}
break;
}
var n=Math.floor(Math.random() * (n_opciones)); //elige una opcion al azar
txt_opcion=opciones[n]; //carga la opcion elegida, para mostrarla en pantalla
if (n<n_correctas){ //define si la opcion elegida es correcta o incorrecta
correcta=true;
}
}

```

```

else
{
correcta=false;
}
}
//toma el canvas creado
var tablero=document.getElementById("mi_canvas"); //define el contexto : un lienzo 2D
var ctx = tablero.getContext("2d");

//Marca el punto central del canvas

var x = tablero.width/2;
var y = tablero.height/3;

//Define Radio de la Bola
var ballRadius = 10;

//Define el tamaño del paso tras cada click a las flechas
var dx = 2*(Math.floor(Math.random()*4))-3;;
var dy = -2;

//inicializa puntaje en cero
var puntaje=0;
genera_pregunta();
var vidas=3;

//Define el tamaño de la canasta y su centro inicial
var altura_caja = 15;
var ancho_caja = 75;
var izq_caja = (tablero.width-ancho_caja)/2;

//Indica que inicialmente las teclas de flecha no estan presionadas
var press_derecha = false;
var press_izquierda = false;

//Le dice al sistema que debe estar atento a los eventos Presionar tecla y soltar tecla
document.addEventListener("keydown", keyDownHandler , false);
document.addEventListener("keyup", keyUpHandler, false);

// Lee que tecla se ha presionado y compara con el código ASCII de las flechas derecha e izquierda
function keyDownHandler(e) {

```

```

    if(e.keyCode == 39) {
        press_derecha = true;
    }
    else if(e.keyCode == 37) {
        press_izquierda = true;
    }
}

```

// Lee cuando se suelta la tecla ...

```

function keyUpHandler(e) {
    if(e.keyCode == 39) {
        press_derecha = false;
    }
    else if(e.keyCode == 37) {
        press_izquierda = false;
    }
}

```

//Dibujar un círculo, como un arco del radio definido y 2Pi radianes (equivalente a 360 grados)

```

function drawBall() {
    ctx.beginPath();
    ctx.arc(x, y, ballRadius, 0, Math.PI*2);
    ctx.fillStyle = "lightblue"; //Color de la pelota
    ctx.fill(); //Rellenar la pelota
    ctx.closePath();
    ctx.beginPath();
    ctx.font="7px Arial";
    ctx.fillStyle = "black";
    ctx.fillText(txt_opcion,x-3,y+3);
    ctx.closePath();
}

```

//Dibujar la paleta, como una caja rectangular con las medidas definidas anteriormente

```

function drawPaddle() {

    ctx.beginPath();
    ctx.rect(izq_caja+3, tablero.height-altura_caja, ancho_caja-6, altura_caja);
    ctx.fillStyle = "blue"; //Color de la Paleta
    ctx.fill();
    ctx.closePath();
    ctx.beginPath();
    ctx.rect(izq_caja+6, tablero.height-altura_caja, ancho_caja-12, altura_caja-3);
}

```

```

ctx.fillStyle = "red";
  ctx.fill();
  ctx.closePath();
}
function reinicia_vars(){ //reinicia las variables del juego
x = tablero.width/2;
y = tablero.height/3;
dx = 2*(Math.floor(Math.random()*4))-3;
dy = -2;
altura_caja = 15;
ancho_caja = 75;
press_derecha = false;
press_izquierda = false;
genera_pregunta();
}

function draw() {
ctx.clearRect(0, 0, tablero.width, tablero.height); //Limpia el lienzo
  drawBall(); //Dibuja la pelota
  drawPaddle(); //Dibuja la Paleta

  if(x + dx > tablero.width-ballRadius || x + dx < ballRadius) {
//verifica si la pelota choco con los bordes laterales, rebota
    dx = -dx;
  }
  if(y + dy < ballRadius) { //verifica si la pelota choca arriba, rebota
    dy = -dy;
  }
  else if(y + dy > tablero.height-ballRadius)
  {

//si paleta
if(x > izq_caja && x < izq_caja + ancho_caja)
{
//Verifica si la pelota choco en la paleta.

//          si correcta

if (correcta)
{

//alert("Correcto");
//nueva_respuesta
//mas_puntaje

puntaje+=50;
document.getElementById("ptje").value=puntaje;

```

```

//reinicia variables
reinicia_vars();
}

// sino
else{

// menos vidas
// alert("UPS");

vidas-=1;
document.getElementById("vidas").value=vidas;

// si no quedan vidas
if (vidas==0)
{

// game over
var respuesta=confirm("GAME OVER \n Jugar Nuevamente");           //preguntar
if (respuesta)                                                     // sigo ... reinicio variables y empiezo de nuevo
{
vidas=3;
document.getElementById("vidas").value=vidas;
puntaje=0;
document.getElementById("ptje").value=puntaje;
alert ("INICIAR NUEVO JUEGO");
delay(3000);
reinicia_vars();
}
else                                                                 // no sigo ... fin
{
clearInterval(juego);
}
}
else                                                                 // si quedan vidas

{

// nueva bolita

reinicia_vars();
}
}
}
else                                                                 // si no es paleta (fondo)
{
if (correcta)

```

```

{
    // menos vidas
    // alert("UPS");

vidas-=1;
document.getElementById("vidas").value=vidas;

    // si no quedan vidas

if (vidas==0)
{
    // game over
var respuesta=confirm("GAME OVER \n Jugar Nuevamente"); // preguntar
if (respuesta) // sigo ... reinicio variables y empiezo de nuevo
{
vidas=3;
document.getElementById("vidas").value=vidas;
puntaje=0;
document.getElementById("ptje").value=puntaje;
alert ("INICIAR NUEVO JUEGO");
delay(3000);
reinicia_vars();
}
else // no sigo ... fin
{
clearInterval(juego);
}
}
Else // si quedan vidas

{ // nueva bolita
reinicia_vars();
}
}
else
{
reinicia_vars()
}
}
}

//Controla el movimiento de la paleta

if(press_derecha && izq_caja < tablero.width-ancho_caja) {
    izq_caja += 7;
}
else if(press_izquierda && izq_caja > 0) {
    izq_caja -= 7;
}

```



```
    }  
    x += dx;  
    y += dy;  
  }  
  juego=setInterval(draw, 30);  
}
```

```
</script>
```

```
<script>  
//var nombre=prompt("Ingresa tu nombre ");  
//document.getElementById("nomb_jugador").innerHTML=nombre;  
</script>
```