**Curso online Latam**

**Seo (search engine optimization)**

Técnicas que te permiten posicionarte de primero en una búsqueda. Aplica que cada etiqueta tiene un peso específico

Es bueno poner palabras claves en el título (title, h1 y h2) y otras etiquetas., esas palabras claves deben estar esparcidas por toda la página y ponerlas como el usuario las buscaría.

Pero entre menos palabras posibles y más palabras claves, mejor.

La imagen tiene peso también en relación a su nombre y al atributo “alt”

**Loren ipsu**

Texto de prueba. En el Sublime se puede poner Loren tab y aparecerá el texto

/\* Es buena práctica no justificar el texto \*/

**Reglas css**

La última orden es la que manda

La regla más específica es la que manda, la que se pone en más detalle

**Span**

Permite modificar una palabra dentro de un texto grande. Se utiliza con style

<p> <span style = “color: Green;”>Scscsds </span> adasscas dfdsfsd </p>

**textarea**

Crea una caja de texto

**Explicación colores haxadecimal:**

Los valores hexagesimales permiten poner 256 variantes de cada color, se llaman hexagesimales porque permiten valores que van desde 0 hasta f, donde el valor 10 es representado con una a, el 11 con una b y finalmente el 16 con una f

FF es el valor 255, 00 es el valor 0

Los grises generalmente son cuando todos los colores son iguales:

Ej: #cccccc

Cuando un color es negro puede simplificarse así: #000

**Datos varios**

* En CSS cuando hacemos referencia en el selector, a varios elementos, el orden de estos si afecta,

p.class span etc.. busca analizar datos y extraer conocimiento.g, Big data er cual va a predominar, si un id o una clase, el id predomina por{

propiedad: valor;

}

/\* Esto quiere decir que estamos haciendo referencia a un párrafo de la clase ‘class’ que dentro tiene un ‘span’ \*/

\* Preguntar el orden de las cosas, que cosa está dentro de que?

* Cuando hablamos de las reglas de que lo específico predomina sobre lo general, y que lo que está más abajo predomina sobre lo que está más arriba (regla de la cascada) ¿Cuál de estas dos reglas predomina más en caso de encontrarnos en un caso en donde se encuentren ambas?

Lo específico predomina por sobre la regla de la cascada

/\* Siguiendo esta misma regla, si por ejemplo se tuviera que ver cual va a predominar, si un id o una clase, el id predomina porque es más particular que una clase \*/

* Recuerda que así a un elemento se le da más de una clase:

<h1 class = “book domestic”> Nombre del libro</h1> /\* tiene la clase book y la clase domestic \*/

* Preguntar para que sirve la llave SSH

\* Data sience: Incluye todos los conocimientos como minería de datos, Big data, Bussines intelligence etc.. busca analizar datos y extraer conocimiento.

**Listas**

**Ul (unordered list)**

En ccs puede tener el estilo de

Selector {

list-style-type: disc; /\* aparece un punto al lado de la lista \*/

list-style-type: none; /\* saca cualquier estilo \*/

list-style-type: circle; /\* pone las listas con un círculo vacío \*/

/\* En el inspector de elementos, uno puede ir viendo y cambiando el tipo de puntero de lista (por ejemplo el punto) \*/

}

**Float**

Permite poner las listas de forma horizontal

Selector{

Float: right; /\* pone la lista a la derecha \*/

Float: left; /\* pone la lista a la izquierda \*/

}

**Clear:**

Para que el título no quede pegado a la lista. Hace que el texto se salte una línea

Se pone inmediatamente después del término de la lista (</ul>)

<div style=”clear:both;”>

Por qué este div no se cierra?

**Padding para separar los elementos de la lista**

Se agrega padding a cada uno de los elementos de la lista

<ul>

<li style= “float:left; padding:50px;”>imagen</li>

<li style= “float:left; padding:50px;”>>Inicio</li>

<li style= “float:left; padding:50px;”>>Contacto</li>

</ul>

**Link de anchor: Links que te tele-transportan a un lugar específico de la página**

* Primero se hace un div que será la sección a donde quiero que me tele-transporte el link. A este div debo darle un id: (declaración)

<div id=”tele-trasporta”>

<h1>hola </h1>

</div>

Luego en la parte del link (llamado)

<a href=”#tele-trasporta”> </a> /\* Al href, le paso el nombre del id \*/

**Hacer un anclaje (anchor) hacia otra página:**

Cuando cree una nueva página (contacto por ejemplo) e hice el llamado al link de tele-transportación. Si lo pongo de la manera que está a continuación, estoy llamando a un link que no está declarado en mi página contacto (debido a que el div de declaración no está en esta página sino en index)

<a href=”#tele-trasporta”> </a> /\* Al href, le paso el nombre del id \*/

La solución a esto, es que cuando llame al link de tele-trasportación, además le de la dirección del index, para que se vaya automáticamente a esa página

<a href=”index.html#tele-trasporta”> </a> /\* En donde se está dando la dirección a index.html para que tome ese id pero de la página index \*/

**Extras**

* En el sublime cuando quiero hacer una tabulación contraria, se hace con shift-tab
* En el selector de elementos, en la parte del CSS, se cambia el selector de la parte de más arriba
* Cuando creas un nuevo elemento, se le puede dar toda la ruta con el nombre y te la guardará en esa dirección
* Cuando se crea un div en la parte de HTML, uno escribe div .nombre de la clase, y solo te hará toda la estructura
* Para mover en sublime una línea: ctl+comd+arriba

**Clases**

**Cuando se ocupan las clases**

Cuando tenemos un elemento, por ejemplo un <p> pero no vamos a tener más de un párrafo, los cuales queremos que tengan diferentes estilos, entonces creamos una clase

p.e En una <ul> que puede tener varios <li> con estilos diferentes, se puede hacer una case “navbar” (barra de navegación) para agruparlos

<ul class = “navbar”>

<li>gato</li>

<li>perro</li>

<li>pollo</li>

</ul>

En el CCS se llamaría así:

ul.navbar { /\*no se pone solo navbar porque puede haber más de un elemento con la clase navbar, así es más específico \*/

propiedad: valor;

}

/\* cuando hablamos de clase e id:

id=nombre

clase=tipo

\*/

**Pregunta:**

¿Cómo va el orden de los selectores? Tiene que ver con el orden que están escritos en el HTML o es un tema de una cosa va dentro de otra? Ver ejemplo más abajo

* HTML:

<ul class="navbar">

<li>

<img src= "../images/Leon\_chico.jpg" id="leon\_logo" /> </li>

<li>Inicio</li>

<li>

<a href="index.html#seccion-cursos">Cursos</a>

</li>

<li>

<a href="contacto.html">Contacto</a>

</li>

</ul>

* CSS:

ul.navbar li {

float:left;

padding: 50px;

}

**Escritura**

Forma de escribir cuando se llama a una clase y un selector

ul.navbar li {

propiedad: valor;

}

/\* En donde se pone ul de la clase navbar que afecta a los li (el orden no lo tengo claro) \*/

**Imágenes de fondo**

**Buscar fondo**

<https://www.toptal.com/designers/subtlepatterns/>

/\* página para buscar imágenes de fondo \*/

**Ingresar fondo CSS**

Se agrega la imagen a la carpeta de imágenes, luego en el CSS se agrega a el Body (si se quiere que el fondo esté en toda la página ) o en otro selector si se quiere que el fondo sea sectorizado

Body {

Background-image: url(../images/fondo.jpg);

}

Cuando a un elemento de una lista se le pone el float, tiene alto 0 porque los elementos dentro de el están flotando

A los div les pasa lo mismo: si sus elementos flotan, entonces el alto es 0

Backgound-size

Permite moldear el tamaño de un fondo, en caso que por ejemlo hayamos ingresado una imagen de fondo y esta se me haya repetido

Selector-fondo{

Background-size: 100% auto;

}

**Box sizing**

Cambia el cómo se calcula el tamaño de la caja, es decir que si se tiene 3 elementos que ocupan un 100% (uno de 30% otro de 30% y otro de 40% por ejemplo) de la línea y se agrega un borde de 1px por ejemplo, no va a haber más espacio en esa línea y por lo tanto el último elemento va a tener que posicionarse en una nueva línea

Lo que hace el box sizing es hacer parte del ancho total del elemento, al border, padding (en el ejemplo, que sea parte de ese 30% o 40%)

/\* sólo hará parte al border y al padding\*/

selector box-sizing: border-box;n mna porox-sizingsizing se hace de la siguiente manera:e de ese 30%st porque ya tiene fijado un mna por{

box-sizing: border-box;

}

Si se quiere que todo tu HTML tenga la propiedad box sizing se hace de la siguiente manera:

html{

box-sizing: border-box;

}

\*,\*:before, \*:after{

box-sizing: inherit;

}

/\* todos lo elementos se afectan con esta propiedad

/\* Pone la propiedad box-sizing a todo el HTML y a todos los elementos antes y después van a heredad la propiedad de box-sizing

**que significa esta sintaxis?**

**z-index**

Nos permite ver que elemento irán por sobre otros, decir si es el relative el que irá arriba o si a caso es el static

Selector {

Position: relative;

Left: 200px;

z-index -10; /\* como es negativo, hará que el otro elemento esté arriba \*/

}

**Pregunta**

Si el estar arriba y abajo es como un true/false entonces la cantidad -10 que significa?

**Nav**

Sirve para construir un menú

* **HTML**

<nav>

<ul>

<li>item1</li>

<li> item2</li>

<li> item3</li>

</ul>

</nav>

* **CSS**

Nav{

Width: 50%

Height: 100px;

Background-color: silver;

}

**Tablas**

**Hacer una tabla**

La tabla se escribe de izquierda a derecha es decir en rows=filas

<table>: comienza una tabla

<th> : table head que es el título de cada columna

<tr>: cuando se comienza una nueva fila

<td> lo que va dentro de una nueva fila

**Ejemplo:**

<table>

<tr>

<th>nombre</th>

<th>apellido</th>

<th>ciudad</th>

</tr>

<tr>

<td>javiera</td>

<td>riveros</td>

<td>santiago</td>

</tr>

</table>

**Estilar una tabla**

**En el CSS**

table tr td{

propiedad: valor;

}

**Datos varios**

\* Cada vez que quiero realizar una página, vemos las divisiones que tiene en relación a los cuadros o imágenes y cada uno de esos es un div

\* Al principio del CSS es bueno poner un body con margin 0, eso es un reseteo básico

Body{

Margin:0;

}

\* La mejor forma de centrar un elemento dentro de una caja (por ejemplo div) es con el padding. El padding se hace al elemento más grande, no al más pequeño

.claseDiv {

background-color: #a1243a;

text-align: center;

**padding: 50px, 0;**

}

Tambien se utiliza el margin:

selector{

margin: 0 auto;

}

\* Para poder agregar un documento que tiene un :

<head>

<link type =”text/css” rel=”stylesheet” href=”assets/css/fonts.css” >

</head>

\* Para utilizar dos elementos distintos dentro de una misma clase y darles estilo sin tener que repetir el código:

.nameDiv h1,

.NameDiv h2 {

margin: 0;

}

\* La imagen se comporta como texto por lo tanto se pueden centrar con el text-align

\* Cuando tenemos un fondo y queremos ponerle height, podemos poner

background{

min-height: valor;

}

/\* eso asegura que el fondo tenga un tamaño mínimo, es decir al menos eso, nunca menos \*/

\* Las imágenes puden ser block e inline

\* BOOLET: círculos lista

\* Recuerda que el “display: inline-block;” te sirve para poner los elementos al lado independiente del float

* Escribir como hacer navbar

**Text box**

<form action="/action\_page.php">

Firts name: <input type ="text"

name = "FirstName" value="Mickey"> <br>

Last name: <input type="text" name="LastName" value="Mouse"> <br>

<input type="submit" value="Submit">

</form>

**Tipos de unidades de medida**

**Absolute**: No toma en cuenta el contexto en donde se está aplicando, esto quiere decir que la pantalla se podrá achicar o expandir y el tamaño del elemento va a ser el mismo Ej: el px

**Relative** Son aquellas que dependen de otro elemento padre

Pueden haber de dos tipos:

* Relativas a la tipografía:

1. Tenemos el em: que depende del padre inmediato. Eso quiere decir que:

En el caso del em, esta es una unidad de medida que depende el padre (por ser relativa), por lo tanto si tenemos un elemento de 18 em y dentro de ese elemento tenemos otro de 5em, el valor total será la multiplicación de ambos

1. Rem: Es el mismo caso del anterior, solo que en vez de depender del padre inmediato, depende de la raíz

* Relativas al viewport

1. Vh: El cual hace referencia al tamaño de altura viewport. En donde se ajusta el tamaño de mi elemento en función al espacio de alto del viewport

selector {

height: 50vh; /\* Ese 50 funciona como si fuera un 50% de la página del viewport \*/

}

1. Vw: El cual hace referencia al tamaño de ancho del viewpor. En donde se ajusta el tamaño de mi elemento en función al espacio de ancho del viewport

selector {

width: 50vw;

}

**Porcentajes**

Trabaja en relación al elemento que lo contiene, pero contrario al em, no se multiplicará la cantidad de valores del porcentaje dependiendo de elementos anidados si no que dependerá inmediatamente del padre

**Tipos de diseño de páginas web**

**Fluido/Líquido:** Se utiliza unidades de medida como em y porcentajes en lugar de pixeles (los cuales son absolutos), para que los elementos se adapten según el ancho de la pantalla (debido a que dependen del padre), lo que produce problemas en dispositivos muy grandes (el contenido se estira mucho) o en muy pequeños (difíciles de leer)

Para poder solucionar esto, se puede establecer un min-width o un min-height, pero el problema es que se genera desordenes en el diseño, por ejemplo espacios en blanco en dispositivos grandes

Ej: Si uno cambia el tamaño de la página, se verá que la letra se mueve hacia abajo tratando de caber en la página pero que haciendo eso no queda bien el diseño

**Adaptativo**: Se basan en un punto de quiebre. Cuando la pantalla alcanza un cierto límite, se cambia a otro diseño por ejemplo, el diseño adaptativo detectará el tipo de pantalla que se está usando y se cambiará el tipo de diseño

Si embargo este típo de método involucra crear plantillas para cada tipo de dispositivo, lo que genera que a veces se muestre menos contenido del original

Ej: Si uno cambia el tamaño de la página, se verá que cuando el espacio disminuye pasado cierto punto, desaparecen ciertos elementos, adaptándose al entorno

**Responsivo** En este diseño, el contenido se adapta según cada pantalla/aparato, tomando en cuenta además al navegador

Esto quiere decir, que el usuario ve una diagramación distinta en cada dispositivo pero permite que la información se mantenga igual.

**Estático**: Aquellas páginas que no estaban optimizadas. Por ejemplo aquellas páginas que uno podía encontrar antes de toda la revolución de los celulares.

Ej: Si uno cambia el tamaño de la página, se deberá hacer scroll

**Table-cell**

Permite transformar el contenido como si fuera una tabla

Selector{

Display: table-cell;

}

**Media**

Permite hacer reglas de si o no

<style>

@media(min-width: 400px) and (max-width: 600px) {

body{

background-color: green;

}

}

</style>

**/\* Cuando se habla de px (en este caso 400px) esto recibe el nombre de “break-points” lo cual corresponde al modelo adaptativo \*/**

**Manipular el contenido con media**

@media(min-width: 400px) and (max-width: 600px) {

body::after{

content: "pequeño";

}

}

@media(min-width: 601px) and (max-width: 800px){

body::after{

content: "mediano";

}

}

@media(min-width: 801px) {

body::after{

content: "grande";

}

}

**Pseudo elementos/selectores:**

Es un elemento virtual que puede ser manipulado en su contenido y forma

Irán con un doble punto (::)

**Before- after:** Pseudo selectore que permiten agregar cierto contenido antes y después de un contenido original

**Movile first**

Dice que es bueno pensar como si se estuviera construyendo un sitio para un teléfono y después ahí partir para adelante

**Modelo Responsivo grillas**

**grillas**

<style>

.row {

clear: both;

}

.cell{

float: left;

width: 50%;

}

</style>

</head>

<body>

<div class="row">

<div class="cell">

Lorem ipsum dolor sit amet, consectetur adipisicing elit, sed do eiusmod

</div>

<div class="cell">

Lorem ipsum dolor sit amet, consectetur adipisicing elit, sed do eiusmod

</div>

</div>

**Modelo responsivo grilla avanzado**

Permite hacer una grilla y ocupar más de una columna para poner un mismo dato, es como en el Excel fusionar columnas

<style>

.row{

clear: both;

}

.col-md-1, .col-md-2, .col-md-3, .col-md-4, .col-md-5, .col-md-6, .col-md-7, .col-md-8,

.col-md-9, .col-md-1{

float: left;

}

.col-md-1{

width: 10%;

}

.col-md-2{

width: 20%;

}

.col-md-3{

width: 30%;

}

.col-md-4{

width: 40%;

}

.col-md-5{

width: 50%;

}

.col-md-6{

width: 60%;

}

.col-md-7{

width: 70%;

}

.col-md-8{

width: 80%;

}

.col-md-9{

width: 90%;

}

.col-md-10{

width: 100%;

}

</style>

</head>

<body>

<div class= "row">

<div class="col-md-1">

una columna

</div>

<div class="col-md-2">

yo tengo dos columnas, así que tengo más espacio

</div>

<div class="col-md-4">

Lorem ipsum dolor sit amet, consectetur adipisicing elit, sed do eiusmod

tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam,

quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo

consequat. Duis aute irure dolor in reprehenderit in voluptate velit esse

cillum dolore eu fugiat nulla pariatur. Excepteur sint occaecat cupidatat non

proident, sunt in culpa qui officia deserunt mollit anim id est laborum.

</div>

</div>

**/\* Siempre el contenido será primero el row y luego las columnas \*/**

**Offsets**

Cuando uno hace grillas, para que me marque la columna1, columna2, etc… se tiene que poner como si fuera un div, pero que pasa si yo me quiero saltar las columnas del principio y poner data sólo en la columna 4 y que me quede el espacio en blanco de la columna 1, 2, y 3? Esto se hace con los ofset, que crean un especie de margen

Mover una columna

CSS:

.col-md-1{

width: 10%;

}

.col-md-offset-1 {

margin-left: 10%;

}

HTML:

<div class= "row">

<div class = "col-md-4 col-md-offset-1">

Lorem ipsum dolor sit amet, consectetur adipisicing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat. Duis aute irure dolor in reprehenderit in voluptate velit ese cillum dolore eu fugiat nulla pariatur. Excepteur sint occaecat cupidatat non proident, sunt in culpa qui officia deserunt mollit anim id est laborum.

</div>

</div>

Mover dos columnas

CSS:

.col-md-2{

width: 20%;

}

.col-md-offset-2 {

margin-left: 12%;

}

HTML:

<div class= "row">

<div class = "col-md-4 col-md-offset-2">

Lorem ipsum dolor sit amet, consectetur adipisicing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat. Duis aute irure dolor in reprehenderit in voluptate velit ese cillum dolore eu fugiat nulla pariatur. Excepteur sint occaecat cupidatat non proident, sunt in culpa qui officia deserunt mollit anim id est laborum.

</div>

</div>

**Ver el modelo grilla con el inspector de elementos**

Inspector de elementos/network/tab CSS/Apretar el documento en donde está el css

* ponemos stop recording

**Grilla con break points**

**HTML**

<div class = "col-md-4 col-md-offset-2 .col-sm-5">

Lorem ipsum dolor sit amet, consectetur adipisicing elit, sed do eiusmod

</div>

**CSS**

.col-md-1, .col-md-2, …

.col-sm-1, .col-sm-2, …

{

float: left;

}

@media(max-width: 480px){

.col-sm-1{

width: 10%;

}

.col-sm-2{

width: 20%;

}

.

.

.

}

@media(min-width: 481px){

.col-md-1{

width: 10%;

}

.col-md-2{

width: 20%;

}

.

.

.

}

**Modificar el viewport**

<meta name="viewport" content="initial scale=1 width=device-width">

**Variables/custom properties en css**

:root { /\* el root es a lo que se dirige el rem, es la variable para el navegador\*/

--color: red;

}

body{

background-color: var(--color);

}

git b

**Hacer botones**

**HTML**

<div class=”hobbies”>

<form action="/action\_page.php">

<button type="submit" value="Enviar"><a href="http://www.google.com">Fotografía</a></button>

</form>

</div>

CSS

.hobbies button{

background-color: #ae5064;

border-radius: 0%

height: 35px;

font-size: 14px;

}

.hobbies button a{

color: white;

text-decoration: none;

}

**Varios**

* para que un link sea activo, se debe poner en la clase ‘class=”active”’
* Eliminar elemento desde el inspector: uno se posiciona en el elemento/botón derecho/ delete element (el truco es buscar el elemento que los engloba)

**Agregar jquery al código fuente:**

Uno ingresa a jquery.com/poner descargar/se descarga la última versión/luego el archivo se pone en la carpeta que estamos trabajando (generalmente se crea una nueva carpeta llamada ‘js’ que tendrá todos los archivos js, esto a la misma altura que images)/luego justo arriba de:

<!-- Latest compiled and minified JavaScript -->

Se escribirá lo siguiente:

<script type="text/javascript" src="js/jquery-3.2.1.min.js"></script> /\* en donde ‘jquery-3.2.1.min.js’ es el nombre del archivo que descargamos y ‘js/’ es el nombre de la ruta donde dejamos el archivo

**Cargar tu propio archivo css**

Cuando se hace el link para tu propio estilo, se debe necesariamente hacer después de la plantilla que nos descargamos y pusimos después de

<!-- Optional theme -->

Esto porque en CSS la última regla manda.

Quedaría así:

<!-- Optional theme -->

<link rel="stylesheet" href="superhero.css">

<!-- mi estilo -->

<link rel="stylesheet" href="miestilo.css">

<script type="text/javascript" src="js/jquery-3.2.1.min.js"></script>

**CDN**

Cuando uno está en desarrollo es mejor tener el archivo de forma local. Cuando uno está en producción sería mejor el CDN

El CDN es un conjunto de servidores que tienen la misma información que tu página web

**Datos sublime**

cmd-shif-d: Te duplica un texto

Cambiar varios texto se una vez: cmd-d

Siempre ocupar seccion, header, footer como tags

Html:5 para que aparezca la estructura