|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | http://alimentos2014.epn.edu.ec/logo_epn_android.jpg | | **ESCUELA POLITÉCNICA NACIONAL ESCUELA DE FORMACIÓN DE TECNÓLOGOS** | | | | | |  |
|  |  |  |  |  |  |  |  |
|  | **GUIA DE PRÁCTICAS  LABORATORIO TALLER 4  No. Práctica: 4** | | | | | |  |
|  |  |  |  |  |  |  |  |  |  |
|  | **CARRERA:** |  | **ASA**  \_\_\_ **ASI** \_X\_ **EM** \_\_\_\_ **ET** \_\_\_ | | | |  |  |  |
|  | **ASIGNATURA:** | | Arquitectura Web | | **CÓDIGO:** | TSI463 | **GRUPO:** | GR1 |  |
|  | **FECHA:** |  | (11/12/15) | |  |  |  |  |  |
|  | **APELLIDOS Y NOMBRES :** | | Sánchez Arteaga Fredy Vicente | | | | | |  |
|  |  |  |  |  |  |  |  |  |  |
|  | **CÉDULA DE IDENTIDAD:** | | 1725632552 | |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  | **1. PROPÓSITO DE LA PRÁCTICA:**   * Creación de estilos con CSS dentro de HTML. | | | | | | | |  |
|  |  |
|  |  |
|  |  |  |  |  |  |  |  |  |  |
|  | **2. OBJETIVO GENERAL:**   * Aplicación de estilos CSS dentro de una página web. | | | | | | | |  |
|  |  |
|  |  |
|  |  |  |  |  |  |  |  |  |  |
|  | **3. OBJETIVOS ESPECÍFICOS:**   * Comprender la forma en cómo se visualizan los elementos dentro de una página web. * Comprender la estructura de un color hexadecimal. * Comprender el concepto de selector. * Aplicar selectores dentro de HTML y en un archivo CSS. | | | | | | | |  |
|  |  |
|  |  |
|  |  |
|  |  |  |  |  |  |  |  |  |  |
|  | **4. DESCRIPCIÓN DE ACTIVIDADES Y PROCEDIMIENTO DE LA PRÁCTICA:**  Parte I:  1. En esta oportunidad empezaremos con la parte de los estilos para la página web, o CSS.  2. Anteriormente dentro de la web todo se desplegaba por medio del HTML por lo cual maneja atributos dentro de los tags para poder mostrar la página de manera atractiva al usuario, pero el HTML solo debería mostrar cómo está estructurada la página mas no como se ve, probaremos unos atributos para entender de qué hablamos.  3. Separaremos el contenido de la presentación para esto vamos a usar los CSS los cuales son simplemente código que aparece dentro del HTML, observaremos la diferencia entre una página con CSS y una sin CSS.  4. Escribiremos nuestro primer “selector”, este no es más que el nombre que se le da cuando escribimos el estilo para sepa a que tag va a afectar el mismo, revisaremos un ejemplo de esto, el selector más simple es el de tipo ya que tiene el mismo nombre que el tag que afecta.  5. Un selector tiene una sintaxis específica, la cual consta de el nombre del selector seguido de llaves y dentro de las llaves el nombre de la propiedad que vamos a modificar seguida de “:” y luego el valor que le daremos a la propiedad terminado por un “;”.  6. Un solo selector puede cambiar varias propiedades, para empezar a probar los selectores nos dirigimos a la parte superior donde está el <head> y dentro de este vamos a escribir el siguiente tag <style type=”text/css”></style>, dentro de este bloque vamos a escribir los selectores para darle estilo a la página.  7. Empezaremos con un selector como el siguiente p {} y dentro vamos a poner las siguientes propiedades “text-decoration: underline;”, “color: red;” y observaremos que sucede.  8. Seleccionaremos múltiples tags con un solo selector esta ves modificaremos el tag li con la propiedad “font-size: 24px”, con esto podemos observar cómo afecta a todos los elementos que tenemos con el tag <li>.  9. Qué pasa si solo queremos que el estilo afecte a una parte específica y no a todos los tags, para eso vamos a poner el tag principal al que queremos que afecte por ejemplo el selector queda de la siguiente manera ul li {font-size: 24px} de esta manera solo se verán afectados los elementos de una lista <ul> los elementos de la <ol> no se verán afectados, el selector debe escribir los nombres siempre de izquierda a derecha empezando por los padres.  10. Seleccionaremos tags basados en acciones o condiciones, un pseudo-selector es un modificador que puede ser añadido a un selector para seleccionar un tag cuando cierta condición ocurre, de la siguiente manera: a {text-decoration: none;} a: hover {text-decoration: underline; color: darkred;}, la modificación del estilo se realiza cuando se hace clic en el enlace.  11. Cambiaremos el estilo solo para el primer elemento de la lista <ol> para esto debemos escribir algo como ol li:first-child {color: red;}, de esta manera solo se pintara de rojo el primer elemento de la lista observemos.  12. Revisaremos un poco acerca de los colores dentro del HTML para lo cual necesitaremos saber un poco de los que son los Hexadecimales.  13. Una manera popular de escoger colores en HTML es por medio de los hexadecimales “#FFFF00”, entendamos de que se trata, los primeros 2 números representan la cantidad de color rojo, los dos siguientes la cantidad de verde y los dos últimos la cantidad de azul.  14. Mínimo se puede tener 0 de color y máximo 255 por cada color del RGB (red-green-blue), por lo tanto podemos tener 16777216 diferentes colores dependiendo de las cantidades que pongamos, para poder ver los colores y como se desplegaran en una página podemos ir al siguiente enlace.  15. Como un ejercicio ahora vamos a cambiar los colores de los tags <h1><h2><h3> junto a sus tamaños respectivos, los colores de los enlaces y le quitaremos el subrayado. Modificaremos el tamaño de los elementos de las listas y también de los párrafos para que todo tenga una dimensión adecuada.  16. Notemos algo interesante, cada elemento de la página se muestra con unos espacios, de donde salen estos y es posible modificarlos para que estén más alejados o más pegados, la respuesta es sí y para esto debemos entender algo llamado ¨the box¨ o la caja.  17. Cada ítem mostrado en el <body> esta contenido dentro de una caja invisible, o cajas de contenido, cada ítem padre tiene su propia caja y estos están a nivel de bloque, estos toman como largo todo el espacio de la página, tags que se muestran a nivel de bloque pueden ser <h1> <h2> <h3> <p> <ul> <li>.  18. Tenemos también tags que están a nivel de línea como son <a><img><input><label>, y es debido a que por lo general van dentro de otros tags.  19. Nosotros podemos pasar de nivel de bloque a nivel de línea de la siguiente manera: ul li { display: inline;}  20. Regresando a nuestro problema como cambiamos el espacio entre estas cajas invisibles, “The box model” es una forma de describir los bordes y el espacio que existe entre las cajas de cada tag.  21. Hay 4 partes fundamentales en el “box model”:  22. Área de contenido, es el área que contiene el texto, imágenes que nosotros ponemos.  23. Relleno (padding), es el espacio adicional fuera del área de contenido.  24. Borde, rodean al relleno.  25. Margen, rodean al borde.  26. Presionando f12 abrimos la ventana de la consola del navegador y ahí podemos seleccionar cualquier tag o elemento dentro de nuestra página y en la parte derecha inferior observar “The box model” allí podremos ver las 4 partes mencionadas anteriormente.  27. Aplicaremos ahora estos espacios dentro de nuestro sitio, para esto existen 2 formas: la primera consiste en listar las propiedades a manera de bloque como “padding/top: 6px; padding-right: 3px;” y la otra poner todo en línea “padding: 6px 3px 0 0;” lo cual es mucho más simple recordemos que el orden es importante (top, rigth, botton, left).  28. Para la parte del borde agregamos propiedades como “border: 6px solid black” (width, style, color), dentro del <h2>.  29. Además podemos solo poner una el borde en un lado u otro así: “border/bottom: 6px solid black” de esta manera tendremos solo una línea en la parte de abajo y en el resto nada.  30. Para el margen de la misma manera “margin: 6px 0 6px 0;”, este es usado para el espacio entre las cajas.  31. Cuando deberíamos usar el relleno (padding), este es usado para controlar el tamaño de la caja sin ajustar el tamaño del contenido de la misma “padding: 10px 0 10px 0;”.  32. Ahora revisemos nuestro trabajo, pero vamos a seguir notando que hay un espacio ahí que nosotros no hemos configurado, esto es debido a que los navegadores tienen su propia hoja de estilos para cuando no existe una puesta por los desarrolladores.  33. Para desactivar esto nos ubicamos en la parte superior de donde estamos escribiendo los selectors y escribimos lo siguiente: html, body, h1, h2, h3, p, ol, ul, li, a {padding: 0; border: 0; margin: 0;}, ahora si todas las propiedades por defecto del navegador están desactivadas y solo funcionaran las que nosotros pongamos.  34. Ahora que esta desactivado todo el estilo por defecto vamos a poner nuestras propias reglas, empezaremos por el body {padding: 20px 20px 20px 20px;}, esto empujara al resto de tags adentro.  35. Luego vamos por h1 {margin: 10px 0 15px 0;}, este es el espacio vertical entre los tags.  36. En el h2 {margin: 10px 0 20px 0;}, no te preocupes si tenías más cosas antes dentro de este simplemente modificamos una propiedad el resto se puede mantener.  37. En el h3 {margin: 15px 0 15px 0; border-bottom: 1px solid #CCCCCC; padding-bottom: 3px;}.  38. El padding también es usado para la indentación de los ítems así ul {padding: 0 0 0 50px;} y ol { padding: 0 0 0 50px;}.  39. Observemos hasta ahora como ha quedado la página.  Parte II  1. Ya que visualizamos nuestra página nos encontramos con un pequeño problema, la lista de los ingredientes se presenta conforme el menú que tenemos en la parte superior, como vamos a resolver esto.  2. Vamos a añadir un atributo de clase para acceder directamente a ese atributo <ul class= “nav”> y .nav {padding-left: 0}, podemos observar que el padding solo se aplica en el lugar donde llamamos a la clase.  3. Las clases pueden ser usadas junto con los tags para asegurar que los selectores no van a cambiar por ejemplo .nav a {color: #0000af; text-decoration: none;}, este selector va a buscar todos los tags a que sean hijos de cualquier tag que pertenezca a la clase a y pondrá el color a azul y quitara la línea que se dibuja sobre el texto.  4. El problema consistía en que no queremos que los ingredientes se presenten en una sola línea ul li {display: inline}, entonces vamos a cambiar este texto usando .nav li {display: inline}.  5. Cuando escribimos los selectores no importa el orden, pero cuando usamos clases sí importa por ejemplo si un atributo esta puesto en el .nav y otro en el ul, el del ul sobre escribirá al del nav y no tendrá efecto por lo tanto primero se debe poner el selector para el ul y luego el del .nav para que realice efecto.  6. Nuestra página tiene un vínculo hacia una página donde se van a mostrar las recetas, por lo tanto vamos a crear un archivo llamado recetas.html donde vamos a ubicar la receta junto a sus descripción con un enlace a cada receta individual, ahora tenemos un problema esta página no tiene ningún estilo por lo tanto tendríamos que copiar todo el estilo otra vez dentro de la nueva página, pero eso no es óptimo.  7. Para solventar el problema anterior vamos a crear un archivo llamado main.css, dentro de este vamos a copiar todo lo que está dentro de nuestro tag <style>.  8. Este archivo va a contener todos los estilos y solo lo vamos a enlazar en el <head> de cada uno de nuestros archivos.html.  9. Para enlazar al archivo.css debemos ponerlo dentro de un tag <style>, este debe estar ubicado dentro del <head>, este tag no contiene un tag de cierre.  10. Como es un tag vacío vamos a agregar varios atributos dentro para tener algo de la siguiente manera: <link type = “text/css” rel=”stylesheet” href=”main.css”>, el atributo “type” con el contenido de “text/css” le dice al navegador donde va a cargar el archivo css, el atributo “rel” es un alias para “relationship”, este le dice al HTML que debe usar un archivo enlazado para determinar estilos de página, el atributo “href” funciona justo como lo hace el <a>, es un path relativo así que sabemos que el archivo css está dentro de la misma carpeta que nuestros archivos HTML.  11. Ahora podemos ver como nuestra página está usando el estilo que nosotros creamos y la diferencia que esto representa.  12. No siempre usar el mismo estilo en todas las paginas las hace lucir mejores, vamos a modificar un poco nuestro estilo, vamos a quitar el borde dentro de <h3>, el “list-style-type” de <ul>, vamos a poner un borde alrededor de <li>, “padding-left” entre el borde de <li> y el “content área” y el color verde en <a>.  13. Para esto le vamos a dar al <ul> una clase específica para las recetas .recipes {list-style-type: none; padding: 0;}.  14. Después escribiremos algo más específico, .recipes li { border: 1px solid #b56663; padding-left: 15px;}.  15. Además .recipes h3 { border-bottom: 0;} y .recipes a{ color: #7facaa; text-decoration: none;}, revisemos los resultados.  16. Organizaremos nuestro “layaout”, esto no es más que una división imaginaria como cuadricula de los elementos en los cuales está divido un espacio en este caso nuestra página, lo vamos a dividir en “header”, “main content”.  17. Para realizar las divisiones anteriores vamos a usar el tag <div> este es una división, es un tag de tipo bloque, usaremos uno de estos por cada división de contenido que vamos a realizar, para diferenciar los espacios usaremos una clase, por lo tanto usaremos los nombres de las divisiones para crear clases para los <div>.  18. Dejaremos a los div manejar el “padding” .body { padding; 0 0 0 0;}, .header { padding: 10px 10px 10px 10px; }, .main-content { padding: 20px 20px 20px 20px; border: 1px solid #dddddd; margin: 30px 0 0 0; }, observemos los cambios.  19. Hagamos que el “header” se vea diferente para diferenciar los espacios, .header { padding: 10px 10px 10px 10px; background-color: #b56663; }, .header a{ color: #ffffff; }, h1{ color: #ffffff; }, .main-content { padding: 20px 20px 20px 20px; border: 1px solid #dddddd; margin: 30px 0 0 0; width: 500px;}, el atributo “width” y “height” pueden ser cambiados dentro del CSS, con esto le damos un tamaño especifico que va a usar, sino usara el 100% del espacio disponible.  20. Para centrar el contenido usamos .main-content { margin: 30px auto 0 auto;}, con esto le decimos que el margen en derecha e izquierda sea automático.  21. Para centrar el texto en cambio usaremos: h1{ text-align: center; }, .nav { text-align: center; }, hay dos formas de centrar el contenido como pudimos observar, todo depende de que es lo que se quiera centrar. | | | | | | | |  |
|  |  |
|  |  |
|  |  | | **ESCUELA POLITÉCNICA NACIONAL ESCUELA DE FORMACIÓN DE TECNÓLOGOS** | | | | | |  |
|  |  |  |  |  |  |  |  |
|  | **http://alimentos2014.epn.edu.ec/logo_epn_android.jpg5. TÉCNICAS E INSTRUMENTOS APLICADOS:**   * Cloud9 * Brackets * Navegador | | | | | | | |  |
|  |  |
|  |  |
|  |  |
|  |  |  |  |  |  |  |  |  |  |
|  | **6. RESULTADOS**  Parte I:  1. En esta oportunidad empezaremos con la parte de los estilos para la página web, o CSS.  **CSS.-** Es un lenguaje utilizado en la presentación de documentos HTML. Entonces podemos decir que el lenguaje CSS sirve para organizar la presentación y aspecto de una página web.  2. Anteriormente dentro de la web todo se desplegaba por medio del HTML por lo cual maneja atributos dentro de los tags para poder mostrar la página de manera atractiva al usuario, pero el HTML solo debería mostrar cómo está estructurada la página mas no como se ve, probaremos unos atributos para entender de qué hablamos.    3. Separaremos el contenido de la presentación para esto vamos a usar los CSS los cuales son simplemente código que aparece dentro del HTML, observaremos la diferencia entre una página con CSS y una sin CSS.    5. Un selector tiene una sintaxis específica, la cual consta de el nombre del selector seguido de llaves y dentro de las llaves el nombre de la propiedad que vamos a modificar seguida de “:” y luego el valor que le daremos a la propiedad terminado por un “;”.    6. Un solo selector puede cambiar varias propiedades, para empezar a probar los selectores nos dirigimos a la parte superior donde está el <head> y dentro de este vamos a escribir el siguiente tag <style type=”text/css”></style>, dentro de este bloque vamos a escribir los selectores para darle estilo a la página.    7. Empezaremos con un selector como el siguiente p {} y dentro vamos a poner las siguientes propiedades “text-decoration: underline;”, “color: red;” y observaremos que sucede.    8. Seleccionaremos múltiples tags con un solo selector esta ves modificaremos el tag li con la propiedad “font-size: 24px”, con esto podemos observar cómo afecta a todos los elementos que tenemos con el tag <li>.    9. Qué pasa si solo queremos que el estilo afecte a una parte específica y no a todos los tags, para eso vamos a poner el tag principal al que queremos que afecte por ejemplo el selector queda de la siguiente manera ul li {font-size: 24px} de esta manera solo se verán afectados los elementos de una lista <ul> los elementos de la <ol> no se verán afectados, el selector debe escribir los nombres siempre de izquierda a derecha empezando por los padres.    10. Seleccionaremos tags basados en acciones o condiciones, un pseudo-selector es un modificador que puede ser añadido a un selector para seleccionar un tag cuando cierta condición ocurre, de la siguiente manera: a {text-decoration: none;} a: hover {text-decoration: underline; color: darkred;}, la modificación del estilo se realiza cuando se hace clic en el enlace.  C:\Users\TU FACIL CREDITO\Pictures\Screenshots\Captura de pantalla (57).png  11. Cambiaremos el estilo solo para el primer elemento de la lista <ol> para esto debemos escribir algo como ol li:first-child {color: red;}, de esta manera solo se pintara de rojo el primer elemento de la lista observemos.    12. Revisaremos un poco acerca de los colores dentro del HTML para lo cual necesitaremos saber un poco de los que son los Hexadecimales.    **Los dos primeros números establecen la cantidad de rojo, los dos siguientes la cantidad de verde y los dos últimos la cantidad de azul.**  **Este color es el máximo rojo, verde máximo, y no azul= Amarrillo**  **En total tenemos 16’ 777,216 colores.**  13. Una manera popular de escoger colores en HTML es por medio de los hexadecimales “#FFFF00”, entendamos de que se trata, los primeros 2 números representan la cantidad de color rojo, los dos siguientes la cantidad de verde y los dos últimos la cantidad de azul.  15. Como un ejercicio ahora vamos a cambiar los colores de los tags <h1><h2><h3> junto a sus tamaños respectivos, los colores de los enlaces y le quitaremos el subrayado. Modificaremos el tamaño de los elementos de las listas y también de los párrafos para que todo tenga una dimensión adecuada.    16. Notemos algo interesante, cada elemento de la página se muestra con unos espacios, de donde salen estos y es posible modificarlos para que estén más alejados o más pegados, la respuesta es sí y para esto debemos entender algo llamado ¨the box¨ o la caja.  17. Cada ítem mostrado en el <body> esta contenido dentro de una caja invisible, o cajas de contenido, cada ítem padre tiene su propia caja y estos están a nivel de bloque, estos toman como largo todo el espacio de la página, tags que se muestran a nivel de bloque pueden ser <h1> <h2> <h3> <p> <ul> <li>.  C:\Users\TU FACIL CREDITO\Pictures\Screenshots\Captura de pantalla (58).png  18. Tenemos también tags que están a nivel de línea como son <a><img><input><label>, y es debido a que por lo general van dentro de otros tags.  19. Nosotros podemos pasar de nivel de bloque a nivel de línea de la siguiente manera: ul li { display: inline;}    20. Regresando a nuestro problema como cambiamos el espacio entre estas cajas invisibles, “The box model” es una forma de describir los bordes y el espacio que existe entre las cajas de cada tag.  21. Hay 4 partes fundamentales en el “box model”:  22. Área de contenido, es el área que contiene el texto, imágenes que nosotros ponemos.  23. Relleno (padding), es el espacio adicional fuera del área de contenido.  24. Borde, rodean al relleno.  25. Margen, rodean al borde.    26. Presionando f12 abrimos la ventana de la consola del navegador y ahí podemos seleccionar cualquier tag o elemento dentro de nuestra página y en la parte derecha inferior observar “The box model” allí podremos ver las 4 partes mencionadas anteriormente.    27. Aplicaremos ahora estos espacios dentro de nuestro sitio, para esto existen 2 formas: la primera consiste en listar las propiedades a manera de bloque como “padding/top: 6px; padding-right: 3px;” y la otra poner todo en línea “padding: 6px 3px 0 0;” lo cual es mucho más simple recordemos que el orden es importante (top, rigth, botton, left).    28. Para la parte del borde agregamos propiedades como “border: 6px solid black” (width, style, color), dentro del <h2>.    29. Además podemos solo poner una el borde en un lado u otro así: “border/bottom: 6px solid black” de esta manera tendremos solo una línea en la parte de abajo y en el resto nada.    30. Para el margen de la misma manera “margin: 6px 0 6px 0;”, este es usado para el espacio entre las cajas.    31. Cuando deberíamos usar el relleno (padding), este es usado para controlar el tamaño de la caja sin ajustar el tamaño del contenido de la misma “padding: 10px 0 10px 0;”.    32. Ahora revisemos nuestro trabajo, pero vamos a seguir notando que hay un espacio ahí que nosotros no hemos configurado, esto es debido a que los navegadores tienen su propia hoja de estilos para cuando no existe una puesta por los desarrolladores.    33. Para desactivar esto nos ubicamos en la parte superior de donde estamos escribiendo los selectores y escribimos lo siguiente: html, body, h1, h2, h3, p, ol, ul, li, a {padding: 0; border: 0; margin: 0;}, ahora si todas las propiedades por defecto del navegador están desactivadas y solo funcionaran las que nosotros pongamos.    34. Ahora que esta desactivado todo el estilo por defecto vamos a poner nuestras propias reglas, empezaremos por el body {padding: 20px 20px 20px 20px;}, esto empujara al resto de tags adentro.    35. Luego vamos por h1 {margin: 10px 0 15px 0;}, este es el espacio vertical entre los tags.    36. En el h2 {margin: 10px 0 20px 0;}, no te preocupes si tenías más cosas antes dentro de este simplemente modificamos una propiedad el resto se puede mantener.    37. En el h3 {margin: 15px 0 15px 0; border-bottom: 1px solid #CCCCCC; padding-bottom: 3px;}.    38. El padding también es usado para la indentación de los ítems así ul {padding: 0 0 0 50px;} y ol { padding: 0 0 0 50px;}.    39. Observemos hasta ahora como ha quedado la página.    Parte II  1. Ya que visualizamos nuestra página nos encontramos con un pequeño problema, la lista de los ingredientes se presenta conforme el menú que tenemos en la parte superior, como vamos a resolver esto.    2. Vamos a añadir un atributo de clase para acceder directamente a ese atributo <ul class= “nav”> y .nav {padding-left: 0}, podemos observar que el padding solo se aplica en el lugar donde llamamos a la clase.    3. Las clases pueden ser usadas junto con los tags para asegurar que los selectores no van a cambiar por ejemplo .nav a {color: #0000af; text-decoration: none;}, este selector va a buscar todos los tags a que sean hijos de cualquier tag que pertenezca a la clase a y pondrá el color a azul y quitara la línea que se dibuja sobre el texto.    4. El problema consistía en que no queremos que los ingredientes se presenten en una sola línea ul li {display: inline}, entonces vamos a cambiar este texto usando .nav li {display: inline}.    5. Cuando escribimos los selectores no importa el orden, pero cuando usamos clases sí importa por ejemplo si un atributo esta puesto en el .nav y otro en el ul, el del ul sobre escribirá al del nav y no tendrá efecto por lo tanto primero se debe poner el selector para el ul y luego el del .nav para que realice efecto.    6. Nuestra página tiene un vínculo hacia una página donde se van a mostrar las recetas, por lo tanto vamos a crear un archivo llamado recetas.html donde vamos a ubicar la receta junto a sus descripción con un enlace a cada receta individual, ahora tenemos un problema esta página no tiene ningún estilo por lo tanto tendríamos que copiar todo el estilo otra vez dentro de la nueva página, pero eso no es óptimo.    7. Para solventar el problema anterior vamos a crear un archivo llamado main.css, dentro de este vamos a copiar todo lo que está dentro de nuestro tag <style>.  8. Este archivo va a contener todos los estilos y solo lo vamos a enlazar en el <head> de cada uno de nuestros archivos.html.    9. Para enlazar al archivo.css debemos ponerlo dentro de un tag <style>, este debe estar ubicado dentro del <head>, este tag no contiene un tag de cierre.    10. Como es un tag vacío vamos a agregar varios atributos dentro para tener algo de la siguiente manera: <link type = “text/css” rel=”stylesheet” href=”main.css”>, el atributo “type” con el contenido de “text/css” le dice al navegador donde va a cargar el archivo css, el atributo “rel” es un alias para “relationship”, este le dice al HTML que debe usar un archivo enlazado para determinar estilos de página, el atributo “href” funciona justo como lo hace el <a>, es un path relativo así que sabemos que el archivo css está dentro de la misma carpeta que nuestros archivos HTML.    11. Ahora podemos ver como nuestra página está usando el estilo que nosotros creamos y la diferencia que esto representa.    12. No siempre usar el mismo estilo en todas las paginas las hace lucir mejores, vamos a modificar un poco nuestro estilo, vamos a quitar el borde dentro de <h3>, el “list-style-type” de <ul>, vamos a poner un borde alrededor de <li>, “padding-left” entre el borde de <li> y el “content área” y el color verde en <a>.    13. Para esto le vamos a dar al <ul> una clase específica para las recetas .recipes {list-style-type: none; padding: 0;}.    14. Después escribiremos algo más específico, .recipes li { border: 1px solid #b56663; padding-left: 15px;}.    15. Además .recipes h3 { border-bottom: 0;} y .recipes a{ color: #7facaa; text-decoration: none;}, revisemos los resultados.    16. Organizaremos nuestro “layaout”, esto no es más que una división imaginaria como cuadricula de los elementos en los cuales está divido un espacio en este caso nuestra página, lo vamos a dividir en “header”, “main content”.  17. Para realizar las divisiones anteriores vamos a usar el tag <div> este es una división, es un tag de tipo bloque, usaremos uno de estos por cada división de contenido que vamos a realizar, para diferenciar los espacios usaremos una clase, por lo tanto usaremos los nombres de las divisiones para crear clases para los <div>.    18. Dejaremos a los div manejar el “padding” .body { padding; 0 0 0 0;}, .header { padding: 10px 10px 10px 10px; }, .main-content { padding: 20px 20px 20px 20px; border: 1px solid #dddddd; margin: 30px 0 0 0; }, observemos los cambios.    19. Hagamos que el “header” se vea diferente para diferenciar los espacios, .header { padding: 10px 10px 10px 10px; background-color: #b56663; }, .header a{ color: #ffffff; }, h1{ color: #ffffff; }, .main-content { padding: 20px 20px 20px 20px; border: 1px solid #dddddd; margin: 30px 0 0 0; width: 500px;}, el atributo “width” y “height” pueden ser cambiados dentro del CSS, con esto le damos un tamaño especifico que va a usar, sino usara el 100% del espacio disponible.    20. Para centrar el contenido usamos .main-content { margin: 30px auto 0 auto;}, con esto le decimos que el margen en derecha e izquierda sea automático.    21. Para centrar el texto en cambio usaremos: h1{ text-align: center; }, .nav { text-align: center; }, hay dos formas de centrar el contenido como pudimos observar, todo depende de que es lo que se quiera centrar. | | | | | | | |  |
|  |  |
|  |  |
|  |  |
|  |  |  |  |  |  |  |  |  |  |
|  | **7. CONCLUSIONES**     * Al utilizar un archivo css para los estilos de presentación en una página web html es más fiable y reconocible cada una de las variantes así también todas las especificaciones dentro de la presentación de la página web. * Los estilos son aplicados de manera que se presenta una página parcialmente agradable a la vista. * La combinación entre css y html es necesaria para mantener un orden adecuado. | | | | | | | |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |  |  |  |  |  |  |  |  |
|  | **8. BIBLIOGRAFÍA REFERENCIAL:** | | | | | | | |  |
|  |  |
|  |  |
|  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  | **Fredy** | **Sánchez A.** |  |  |  |  |
|  | **FIRMA DEL ESTUDIANTE** | | | | | | | |  |