7 de Agosto, 2017

Día 1:

Profesoras: Estefanía y Belén

* Slackoverflow
* Github
* W3C

Temas de clases :

* HTML
* CSS
* BOOTSTRAP
* JS
* FRAMEWORKS JS
* (JQUERY, ANGULAR, REACT)
* JSON , AJAX

**Versionado**

\*Octocat (mascota de GitHub)

Repositorio: Carpetas que contienen el código fuente de un proyecto, y el resto de las carpetas. Cuando esta en mi maquina o en una sola maquina se llama repositorio local.

Dentro de un repositorio local: index.html / carpeta css / carpeta jpg / carpeta js / etc

Versión: conjunto de archivos a los que no quiero modificar mas. Se van a generar versiones a través de las diferentes modificaciones.

Git o bitbacket (repositorios remoto u online): Software de control de versiones. Sirve para guardar los códigos y compartirlos.

GitHub: red social de código (bueno mantenerla actualizada)

Gitclone: clonamos un repositorio online para poder trabajar de manera local.

Comandos GIT:

1. GIT pull
2. GIT status
3. GIT add
4. GIT commit
5. GIT push

8 de Agosto, 2017

Día 2:

Vinculación entre repo y nube

Descarga de cliente de git para poder hacer ese vínculo

Crear la clave o firma SCH

**Iniciación con Git**

Terminal

**git --version**

**ls (aparecen las carpetas)**

**cd carpeta para elegir**

**git clone URL (de github)**

**cd ../ (va a la carpeta anterior)**

**ls -la (muestra carpetas y archivos ocultos)**

Creamos nuevo repo en github

Nombre: prueba1

**Configuración de usuario git**

En terminal

**git config --global user.name “nombre”**

**git config --global user.email** [**nombre@mail.com**](mailto:nombre@mail.com)

**git config --list**

**Crear una carpeta y convertirla en un repo** con **git init** estando en la carpeta

Una vez que tenemos un repo podemos **vincularla con github**

**git remote add origin URL**

**git remote –v** para ver el fetch y puch o rutas

**Si agregamos un archivo cualquiera a la carpeta vinculada y apretamos git status**

El archivo aparece en rojo (no esta vinculado todavía)

Usando **git add nombre de archivo** y luego **git status** lo pasa a mostrar en verde como que **ya esta vinculado**

**También se puede subir todo por medio del git add --all**

Usando **git rm –cached nombre del archivo** nombre de archivo podemos **remover** el archivo si nos equivocamos.

**Git commit –m “comentario”** para **guardar**

**Para subir** al git hay que poner **git push origin master** y va a pedir usuario y contraseña

-## ¿Qué es y para qué sirve?

+![Git acaba con versiones finales](../assets/git/git.png)

+

+Git es un \*\*sistema de control de versiones\*\* distribuido, gratuito y de código abierto. Es decir, es un tipo de software pensando para hacer un seguimiento claro y ordenado de todos los cambios que vas haciendo en los archivos de un proyecto. Mejor todavía, te permite hacerlo en equipo, de forma colaborativa.

+

+Gracias al software de control de versiones, podés mantener tus archivos \*\*organizados, coordinados y protegidos\*\*, tanto de posibles eventos catastróficos como de los inevitables errores cometidos por tus compañeros de equipo (y bueno, también de los tuyos); si eso sucede es fácil solucionar el problema rescatando lo que haga falta de una versión anterior.

+

+Git nació en \*\*2005\*\*, después de que se venga abajo la relación entre la comunidad que desarrollaba el núcleo de Linux y la compañía responsable de BitKeeper, el sistema de control de versiones que venían usando hasta entonces. A estos muchachos (y en particular a Linus Torvalds, el creador de Linux) se les ocurrió que la mejor forma de reemplazarla era desarrollando una herramienta propia. Desde entonces Git ha logrado evolucionar manteniendo su \*\*diseño sencillo\*\* y su tremenda \*\*eficiencia\*\*, incluso para manejar proyectos muy grandes.

+

+### Lista comandos Git:

+

+ git init

+Indicamos que el directorio donde nos encontramos comenzará a ser versionado bajo Git.

+

+ git status

+Vemos el estado actual de nuestra carpeta versionada así como archivos que no están versionados, archivos modificados, archivos eliminados, etc. Los mensajes que podremos ver con este comando pueden ser los siguientes:

+Untracked files: nos mostrará todos los archivos que no han sido agregados al "Stage"

+Changes to be committed: nos indica qué archivos serán agregados a nuestro commit.

+

+ git add

+Agrega los archivos que no están bajo versionamiento (los Untracked files que vimos anteriormente).

+

+ git commit -m "mensaje"

+Confirma los cambios hechos con "git add" en una nueva versión del proyecto; con el parámetro -m le indicamos un mensaje para describir la razón del commit.

+

+ git remote add origin url\_del\_repositorio

+Hasta ahora todos los comandos visto han sido puro trabajo local, con el comando git remote add agregamos nuestro repositorio a un servidor remoto; para hacerlo debemos tener ya creado nuestro repositorio en un servidor y este nos brindará la url.

+

+ git push -u origin master

+Empujamos todo lo que tengamos en nuestros commits al repositorio remoto.

+

+ git pull

+Trae los cambios que haya trabajado otra persona en el mismo repositorio, es recomendable siempre que trabajemos un repositorio que no hayamos iniciado nosotros hacer siempre un git pull primero antes de comenzar a trabajar en los archivos.

+

+ git remote -v

+Nos muestra la url remota del repositorio en el que nos encontremos.

+

+ git clone url\_del\_repositorio nombre\_local

+Realiza una copia en nuestro directorio local de un repositorio existente.

+

+ git branch -all

+En Git se trabaja con branches (ramas en español), originalmente se crea la master por defecto, a raíz de esto nosotros podemos crear nuestras propias ramas, siendo claro una copia de la master en primera instancia. Con este comando podemos ver todas las ramas que tenga el proyecto.

+

+### Extras:

+

+Presentación hecha en Google -> [Git-Cheat Sheet](../assets/git/ppt/git-cheatSheet.pdf)

+

+### Referencias:

+

+La biblia -> [Progit](https://git-scm.com/book/es/v1)

+Link para favoritos-1 -> [Git - la guía sencilla](http://rogerdudler.github.io/git-guide/index.es.html)

+Link para favoritos-2 -> [Introducción a Github](http://www.cristalab.com/tutoriales/introduccion-a-github-en-linux-ubuntu-c106086l/)

Vamos a revisar lo que ha aprendido hasta ahora:

1. Puede añadir títulos de diferentes tamaños utilizando los diferentes elementos partidas: <h1>a través <h6>.
2. Los párrafos se añaden con el <p>elemento.
3. Las listas no ordenadas se crean con los <ul>se añaden utilizando los elementos y los elementos de la lista <li>elemento.
4. Las listas ordenadas se crean con los <ol>se añaden utilizando los elementos y los elementos de la lista <li>elemento.
5. Se pueden añadir enlaces a su página Web utilizando el <a>elemento - no se olvide el hrefatributo!
6. Las imágenes se pueden agregar con el <img>elemento - no se olvide el srcatributo!
7. Imágenes de ayudar a mantener a los usuarios con discapacidad visual cuando <img>elementos incluyen el altatributo.
8. Puede convertir cualquier cosa en un enlace envolviéndolo con un <a>elemento.
9. El espacio en blanco en el archivo HTML no *no* afecta a la colocación de elementos en el navegador.
10. El W3C recomienda 2 espacios de indentación para elementos HTML anidadas.
11. Los comentarios se utilizan para tomar notas en el interior de un archivo HTML. Puede añadir un comentario con <!-- This is a comment -->.

Hasta el momento, el contenido que ha añadido en esta unidad no es muy atractivo a la vista. En la siguiente unidad, aprenderá cómo modificar el aspecto de su contenido med

cómo vincular un archivo HTML y un archivo CSS juntos.

Vamos a revisar lo que ha aprendido hasta ahora:

1. HTML y CSS se guardan en archivos separados con el fin de mantener el código mantenible y de fácil lectura, así como mantener la estructura separada de estilo.

2. El <style>elemento le permite escribir código CSS dentro de un archivo HTML.

3. Una hoja de estilo CSS se puede vincular a un archivo HTML con el <link>elemento, que requiere tres atributos:

* href - conjunto igual a la ruta del archivo CSS.
* type- establecer igual a text/css.
* rel- establecer igual a stylesheet.

En esta lección, usted aprendió sobre los dos lugares diferentes en las que se puede escribir código CSS, pero no escribió ningún código CSS en absoluto.

En la siguiente lección, usted aprenderá acerca de la estructura básica y la sintaxis de CSS para que pueda empezar a utilizar CSS por su cuenta.

Vamos a revisar lo que ha aprendido hasta ahora:

1. Un selector CSS apunta a un elemento HTML.

2. CSS elementos HTML estilo declaraciones. Las declaraciones deben contener las dos cosas siguientes:

* propiedad - la propiedad que desea el estilo.
* valor - el valor de la propiedad que está peinado.

3. declaraciones CSS deben terminar con un punto y coma ( ;)

4. Una regla CSS consta de un selector CSS y las declaraciones dentro del selector.

5. selectores de elementos múltiples pueden ser utilizados para el estilo de múltiples elementos a la vez.

6. Comentarios mantener el código fácil de leer y le permiten experimentar con nuevo código sin tener que quitar el código antiguo.

7. CSS sigue ciertas prácticas recomendadas para el espaciamiento y la sangría:

* Una línea de separación entre un selector y la abertura de llave de rizado.
* No hay separación entre las declaraciones CSS y la apertura y cierre las llaves de la regla CSS.
* Dos espacios de sangría para las declaraciones CSS.
* Una línea de separación entre las reglas CSS.

Review: Colores

¡Gran trabajo! Usted ha aprendido cómo el estilo de un aspecto importante de la experiencia del usuario: el color.

Vamos a revisar lo que ha aprendido hasta ahora:

1. Color de primer plano se refiere al color real de un elemento, combinada con la colorpropiedad.
2. Color de fondo se refiere al color detrás de un elemento, diseñado con la background-colorpropiedad.
3. Hay 147 nombres de colores disponibles.
4. Los colores RGB y hexadecimal ofrecen más de 16 millones de colores posibles.
5. HSL es una nueva manera de definir colores en CSS3.
6. Se puede modificar la opacidad de un color con colores RGBA o HSLA.
7. No todos los navegadores soportan características de CSS más nuevos, como opacidad o HSL, por lo que se deben hacer declaraciones adicionales para soportar una amplia audiencia de usuarios.
8. Hay muchos recursos en el selector de color disponibles en Internet para ayudarle a seleccionar los colores específicos, así como proporcionar los colores en diferentes formatos.

Fuentes

¡Gran trabajo! Ha aprendido a Estilo un aspecto importante de la experiencia del usuario: fuentes!

Vamos a revisar lo que ha aprendido hasta ahora:

1. La font-family propiedad cambia el tipo de letra del texto.
2. Las fuentes serif (time new roman, Georgia) tienen detalles adicionales sobre los extremos de cada letra. fuentes sans-serif (ej: arial, helvética) no lo hacen.
3. fuentes de repliegue se utilizan cuando una determinada fuente no está instalada en el ordenador de un usuario. Se ponen todas las opciones de letra por si no están guardadas en la compu. EJ: h1 { font-family: Garamond, Times, serif; }
4. Fuentes de Google proporciona fuentes libres que se pueden utilizar en un archivo HTML con el <link elemento>. https://fonts.google.com/?query=r
5. Tamaño de la fuente se puede especificar utilizando pixeles, EMS, o porcentajes. (Font-size)(px-em-%)
6. El espaciado vertical entre las líneas de texto puede ser modificado con la line-spacing (em (preferible) o px)propiedad.
7. La separación horizontal entre las palabras puede ser modificado con la word-spacing propiedad. (predeterminado es de 0.25 em)
8. El espacio entre las letras, el núcleo, puede ser modificado con la letter-spacing propiedad.
9. El texto puede aparecer en negrilla con la font-weight (bold o normal) y se le puede aplicar peso (100,200,300) .
10. El texto puede aparecer en cursiva con la font-style (italic o normal) propiedad.
11. El texto puede aparecer en mayúsculas o minúsculas con la text-transform propiedad. (uppercase o lowercase)
12. El texto puede ser alineado con la text-align propiedad. (rigth, left, center)

 Modelo Box

¡Buen trabajo! En esta lección, usted aprendió sobre las cajas que contienen todos los elementos HTML y cómo manipular sus dimensiones.

Vamos a revisar lo que ha aprendido:

1. Todos los elementos HTML están contenidas dentro de una caja.
2. Cajas tienen dos dimensiones: una anchura y una altura. Estas dimensiones se pueden modificar con el width y  height propiedades.
3. Anchura y la altura dimensiones se pueden ajustar mediante una de las tres unidades de medida: píxeles, EMS, o porcentajes.
4. Una anchura o la altura mínima y máxima se pueden establecer para una caja. Esto ayuda a asegurar que el contenido sigue siendo legible cuando un usuario se contrae o se expande la ventana de su navegador. (max-height o min-height) (max-width o min-width)
5. Si la caja de un elemento se hace demasiado pequeño, el contenido puede desbordar. El contenido desbordado se puede controlar con la overflow (scroll (barra para que sea desplazable) o hidden (se oculta el contenido q se se desborda)

Bordes

El estilo de un borde se puede establecer la border-style . Esta propiedad puede tomar uno de los siguientes valores:

1. solid - frontera es una línea continua.
2. dashed - borde es una serie de líneas o rayas.
3. dotted - frontera es una serie de puntos cuadrados.
4. double - borde son dos líneas de color negro sólido.
5. groove - frontera es una ranura (o talla).
6. inset - borde aparece a cortar en la pantalla.
7. outset - borde aparece a salir de la pantalla.
8. ridge - borde aparece como un marco de imagen.
9. hiddeno none- ninguna frontera.

div { border-style: solid; }

En el ejemplo anterior, una línea sólida y negro aparecerá alrededor de todos los divs en la página.

Ancho del borde I

Se puede controlar el grosor o la anchura, de las fronteras con la border-width. El valor de border-width se da en píxeles.

p { border-style: solid; border-width: 5px; }

En el ejemplo anterior, los bordes sólidos de todos los párrafos de la página se pueden configurar para un espesor de 5 píxeles.

También es posible configurar también la border-widthpropiedad a uno de los siguientes espesores nombre:

1. thin
2. medium
3. thick

Si bien estos valores son perfectamente válidos, es posible que no se ve a menudo, pero es bueno saber que existen.

La propiedad border-width le permite especificar el ancho de cada lado de la frontera.

p { border-style: solid; border-width: 3px 1px 2px 1px; }

# Los valores en el ejemplo anterior se refieren a la anchura del borde en *sentido horario* orden (de arriba: 3 píxeles, a la derecha: 1 pixel, abajo: 2 píxeles, izquierdo: 1 pixel). Ancho del borde III

Si desea ser aún más específico sobre el ancho de diferentes lados de la frontera, se pueden utilizar las siguientes propiedades:

1. border-top-width
2. border-right-width
3. border-bottom-width
4. border-left-width

Cada propiedad afecta a la anchura de un solo lado de la frontera, dándole más flexibilidad en la personalización.

Color del borde

El color de una frontera también se puede personalizar con la border-color propiedad.

div.container { border-style: solid; border-width: 3px; border-color: rgb(22, 77, 100); }

El border-color establecimiento acepta colores en los diversos formatos sobre los cuales aprendiste anteriores: nombrado colores RGB (a) colores y colores hexagonales.

 CSS le permite el estilo de las tres propiedades a la vez con una propiedad abreviada.

La forma abreviada de ajuste de estilo de borde, ancho y el color se puede lograr con la border propiedad.

 la mejor práctica a seguir el orden de la anchura de estilo color cuando se utiliza

la borderpropiedad. Bordes del cuadro no tienen que ser cuadrados. Sus esquinas se pueden redondear con la border-radius propiedad.

Vamos a revisar lo que ha aprendido hasta ahora:

1. El relleno es el espacio entre el contenido de una caja y las fronteras de la caja.
2. El relleno se puede establecer por igual en todos los lados del contenido, o se puede configurar específicamente para ciertos lados de sólo el contenido.
3. El margen es el espaciado directamente fuera de las fronteras de una caja.
4. Los márgenes se pueden establecer por igual en todos los lados de una caja, o se pueden configurar específicamente para ciertos lados de la caja solamente.
5. Los displaycambios en el comportamiento predeterminado de los elementos HTML.
6. La visibilitypropiedad se esconde un elemento, pero no elimina la cantidad de espacio que el elemento ocupa en la página. Si desea ocultar ese elemento y eliminar el espacio vacío, utilice la displaypropiedad en lugar

CAMBIAR EL MODELO DE CAJA

Comentario: El cambio del modelo de caja

En esta lección, usted aprendió sobre una importante limitación del modelo de caja por defecto: dimensiones de la caja son afectados por grosor del borde y el relleno.

Vamos a revisar lo que ha aprendido:

En el modelo de caja predeterminada, dimensiones de la caja son afectados por grosor del borde y el relleno.

La box-sizing propiedad controla el modelo de caja utilizado por el navegador.

El valor predeterminado de la box-sizing propiedad es content-box.

El valor para el nuevo modelo de caja es border-box.

El border-box modelo no se ve afectada por el espesor de borde o de relleno.

DISEÑO

Position: relative

Una forma de modificar la posición por defecto de un elemento es estableciendo su positionpropiedad a relative.

Este valor le permite colocar un elemento en relación a su posición estática por defecto en la página web.

.box-bottom {

background-color: DeepSkyBlue;

position: relative;

}

Aunque el código en el ejemplo anterior indica al navegador que esperar un posicionamiento relativo de la div, que no especifica donde el div se debe colocar en la página.

.box-bottom {

background-color: DeepSkyBlue;

position: relative;

top: 20px;

left: 50px;

}

En el ejemplo anterior, el div ha sido posicionado utilizando dos de las cuatro propiedades de desplazamiento . Las propiedades de desplazamiento válidas son:

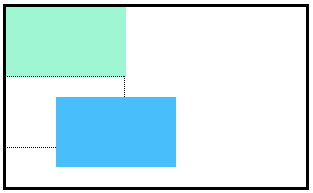
top - mueve el elemento hacia abajo.

bottom - mueve el elemento hacia arriba.

left - mueve el elemento de la derecha.

right - mueve el elemento de la izquierda.

En el ejemplo anterior, el div se moverá hacia abajo 20 píxeles y hacia la derecha 50 píxeles desde su posición estática predeterminada. La imagen de abajo muestra la nueva posición de la caja. La línea de puntos representa el lugar donde se coloca el cuadro de estáticamente posicionado (predeterminado).



Unidades para propiedades de desplazamiento se pueden especificar en píxeles, EMS, o porcentajes. Tenga en cuenta que compensado propiedades no funcionará si la posición del elemento no está ajustado a relative.

**DISEÑO**

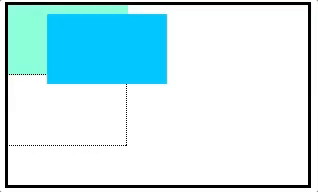
Position: absolute

Otra forma de modificar la posición de un elemento es estableciendo su posición para absolute.

Cuando la posición de un elemento se ajusta a absolutetodos los demás elementos de la página va a *ignorar* el elemento y actuar como que no está presente en la página.

.box-bottom { background-color: DeepSkyBlue; position: absolute; top: 20px; left: 50px; }

En el ejemplo anterior, el .box-bottomdiv se moverá hacia abajo y derecha desde la esquina superior izquierda de la vista. Si propiedades de desplazamiento no se especificaron, el cuadro de arriba sería totalmente cubierto por el fondo de la caja. Echar un vistazo al gif a continuación:



El cuadro de fondo en esta imagen (de color azul) se desplaza desde la esquina superior izquierda de su contenedor. Es 20 píxeles inferiores y 50 píxeles a la derecha de la caja superior.

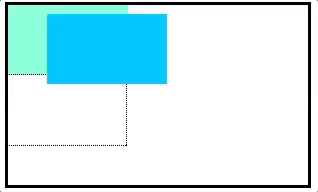
Posición: Fijo

Cuando la posición de un elemento se establece en absolute, al igual que en el último ejercicio, el elemento se desplazará fuera de la vista cuando un usuario se desplaza.

Podemos *fijar* un elemento a una posición específica en la página (independientemente del usuario desplazarse) estableciendo su posición para fixed.

.box-bottom { background-color: DeepSkyBlue; position: fixed; top: 20px; left: 50px; }

En el ejemplo anterior, el div se mantendrá fija en su posición, no importa donde el usuario se desplaza en la página, como en la siguiente imagen:



Este técnica se utiliza a menudo para las barras de navegación en una página web.

DISEÑO

Índice Z

Cuando las cajas en una página web, tienen una combinación de diferentes posiciones, las cajas (y, por lo tanto, su contenido) se pueden superponer entre sí, por lo que el contenido difícil de leer o consumir.

.box-top {

background-color: Aquamarine;

}

.box-bottom {

background-color: DeepSkyBlue;

position: absolute;

top: 20px;

left: 50px;

}

En el ejemplo anterior, el .box-bottomdiv ignora el .box-topdiv y se superpone como un usuario se desplaza.

Los z-indexcontroles de propiedades hasta qué punto "atrás" o qué tan lejos "hacia adelante" un elemento deben aparecer en la página web.

La z-indexpropiedad acepta valores enteros. En función de sus valores, los números enteros indican al navegador en el orden en que los elementos se deben mostrar en la página web.

.box-top {

background-color: Aquamarine;

position: relative;

z-index: 2;

}

.box-bottom {

background-color: DeepSkyBlue;

position: absolute;

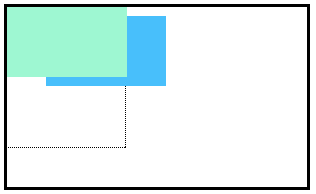
top: 20px;

left: 50px;

z-index: 1;

}

En el ejemplo anterior, hemos creado la .box-topposición de relativa y el índice z a 2. Cambiamos posición para relative, porque la propiedad z-index no no funciona en elementos estáticos. El índice z de 2mueve el .box-topelemento hacia adelante, debido a que es mayor que el .box-bottomíndice z, 1. Ver la imagen siguiente ejemplo:



En la imagen de arriba, se puede ver el cuadro de la parte superior se mueve en frente de la caja inferior.