



Tutor: Limber Choque Quisbert

Clase 6

Flexbox

Propiedades para los elemento hijos

- **Display:** Se coloca al elemento padre
 - **flex** caja flexible
 - **inline-flex** caja en linea flexible
- **flex-direction:** Nos permite establecer la dirección en la que se acomodaran nuestros elementos.
 - **row** | **column** | **row-reverse** | **column-reverse**
- **flex-wrap:** Podemos indicar si todos los elementos estarán en línea
 - **wrap** respeta el ancho y pone abajo si no entra
 - **no-wrap** no respeta el ancho y pone en una linea
 - **wrap-reverse** los elementos se ponen al revés



- **flex-flox: row wrap;** Es un atajo entre flex-direction y flex-wrap, de los distintos valores de los dos
- **justify-content** Nos permite decidir la posición de nuestros elementos como los espacios libres.
 - **flex-start** por defecto
 - **flex-end** posicionamiento a la derecha
 - **center** posiciona los elementos al centro
 - **space-around** los distribuye por todo el ancho
 - **space-between** al inicio y al final no hay márgenes
- **align-items** Nos permite posicionar / distribuir elementos verticalmente.
 - **flex-start** por defecto



- **flex-end** posición abajo
- **center** centrea verticalmente
- **Stretch** tomo el 100% de la altura “height automatico”
- **Baseline** posiciona en la línea de la fuente
- **align-content** Nos permite posicionar / distribuir elementos verticalmente, igual que align-items, solo que funciona para líneas de elementos, por ejemplo cuando tenemos establecido un flex-wrap:wrap
 - **flex-start** | **flex-end** | **center** | **stretch** | **space-between** | **space-around**



Propiedades para los elemento hijos

- **flex-basis;** Establece el tamaño que tendrá un elemento se podría decir que es un sinónimo de width.
- **flex-grow:1;** Establece la proporción de crecimiento que tendrá un elemento a comparación con los demás
- **flex-shrink:1;** Establece la proporción de encogimiento que tendrá un elemento a comparación con los demás.
- **Flex: 1 1 50px;** Atajo entre flex-grow, flex-shrink, flex-basis



- **order: ;** Establecemos el orden que tendrá un elemento a comparacion con otros elementos.
- **align-self:** Nos permite alinear un elemento independientemente de los demás con flex-direction column -> horizontal, row -> verticalmente
flex-start | flex-end | center | stretch | baseline ;
- **z-index: auto | *number* | initial | inherit;** propiedad especifica el orden de pila de un elemento. Un elemento con un orden de apilamiento mayor está siempre delante de un elemento con un orden de apilamiento inferior.



POSICIONAMIENTO

- **position** Posicionamiento del elemento
 - static: Valor por defecto
 - fixed: posiciona con respecto al navegador
 - relative: colocado al elemento “padre
 - absolute: abarca todo el espacio de su contenedor padre
- **top** Distancia de márgenes superior
- **left** Distancia de márgenes izquierda
- **bottom**: Distancia de márgenes inferior
- **right**: Distancia de márgenes derecho



Animaciones

- **animate: name time repetition** Una propiedad abreviada para establecer todas las propiedades de animación.
- **@key-frames name {** la animación cambiará gradualmente del estilo actual al estilo nuevo en ciertos momentos.
0, 25, 50, 75, 100%{ } }
- **animation-delay:** propiedad especifica un retraso para el inicio de una animación.
- **animation-timing-function** propiedad especifica la curva de velocidad de la animación.
 - **ease** - Especifica una animación con un inicio lento, luego rápido, luego termina lentamente (esto es predeterminado)



- **linear** - Especifica una animación con la misma velocidad de principio a fin.
- **ease-in** - Especifica una animación con un inicio lento.
- **ease-out** - Especifica una animación con un final lento.
- **ease-in-out** - Especifica una animación con un inicio y final lento.
- **cubic-bezier(n,n,n,n)** - Le permite definir sus propios valores en una función de cubic-bezier



Transformaciones de 2D

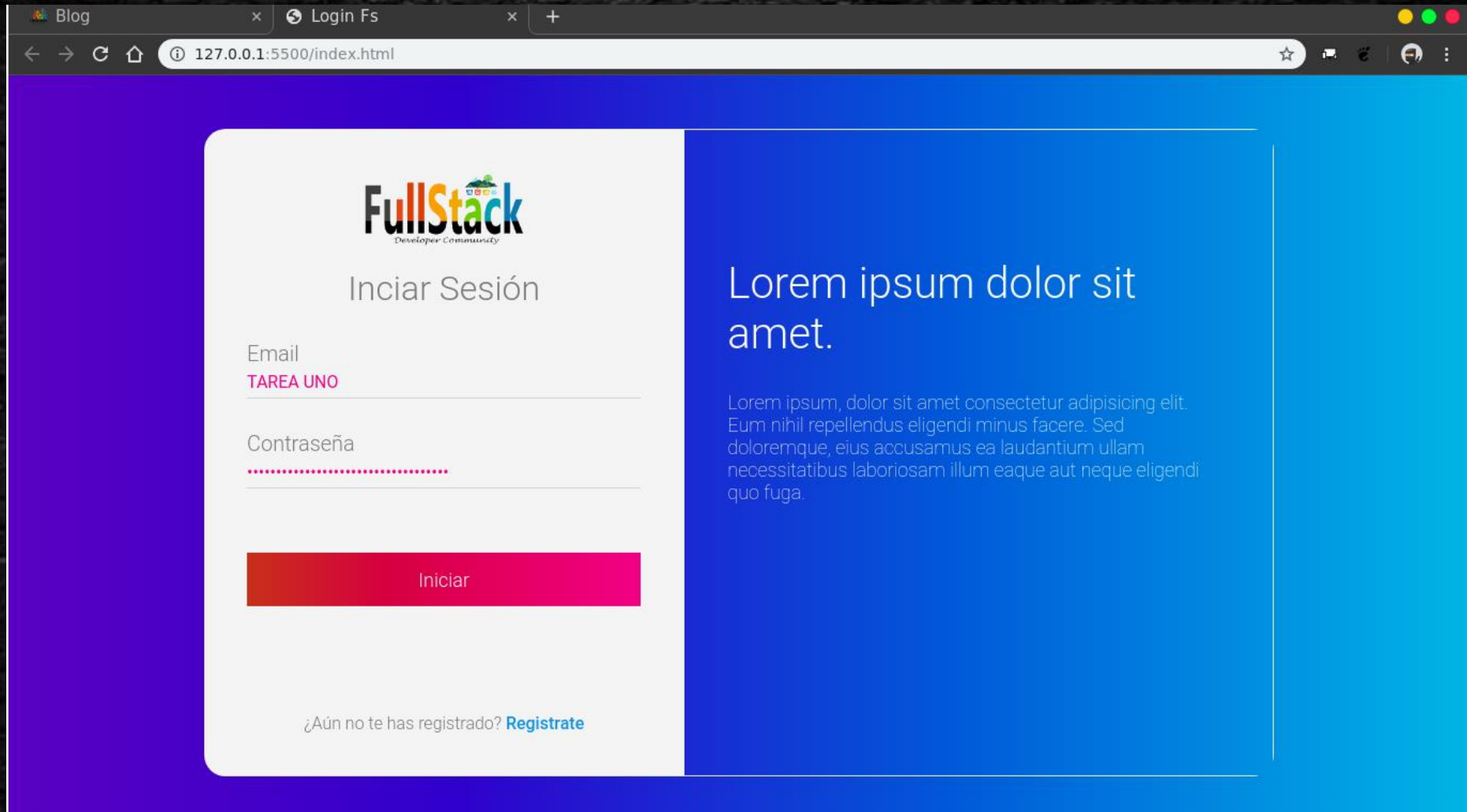
- **transform:** Una transformación es un efecto que permite que un elemento cambie de forma, tamaño y posición CSS soporta transformaciones 2D y 3D.
 - **translate()** método mueve un elemento desde su posición actual (de acuerdo con los parámetros dados para el eje X y el eje Y)
 - **rotate()** gira un elemento en sentido horario o antihorario según un grado dado.
 - **scale()** método aumenta o disminuye el tamaño de un elemento (de acuerdo con los parámetros dados para el ancho y la altura).



- **skewX()** | **skewY()** método sesga un elemento a lo largo del eje X o el eje Y según el ángulo dado
- **-skew()** método sesga un elemento a lo largo de los ejes X e Y según los ángulos dados.
- **matrix()** método combina todos los métodos de transformación 2D en uno.
matriz (scaleX (), skewY (), skewX (), scaleY (), translateX (), translateY ())



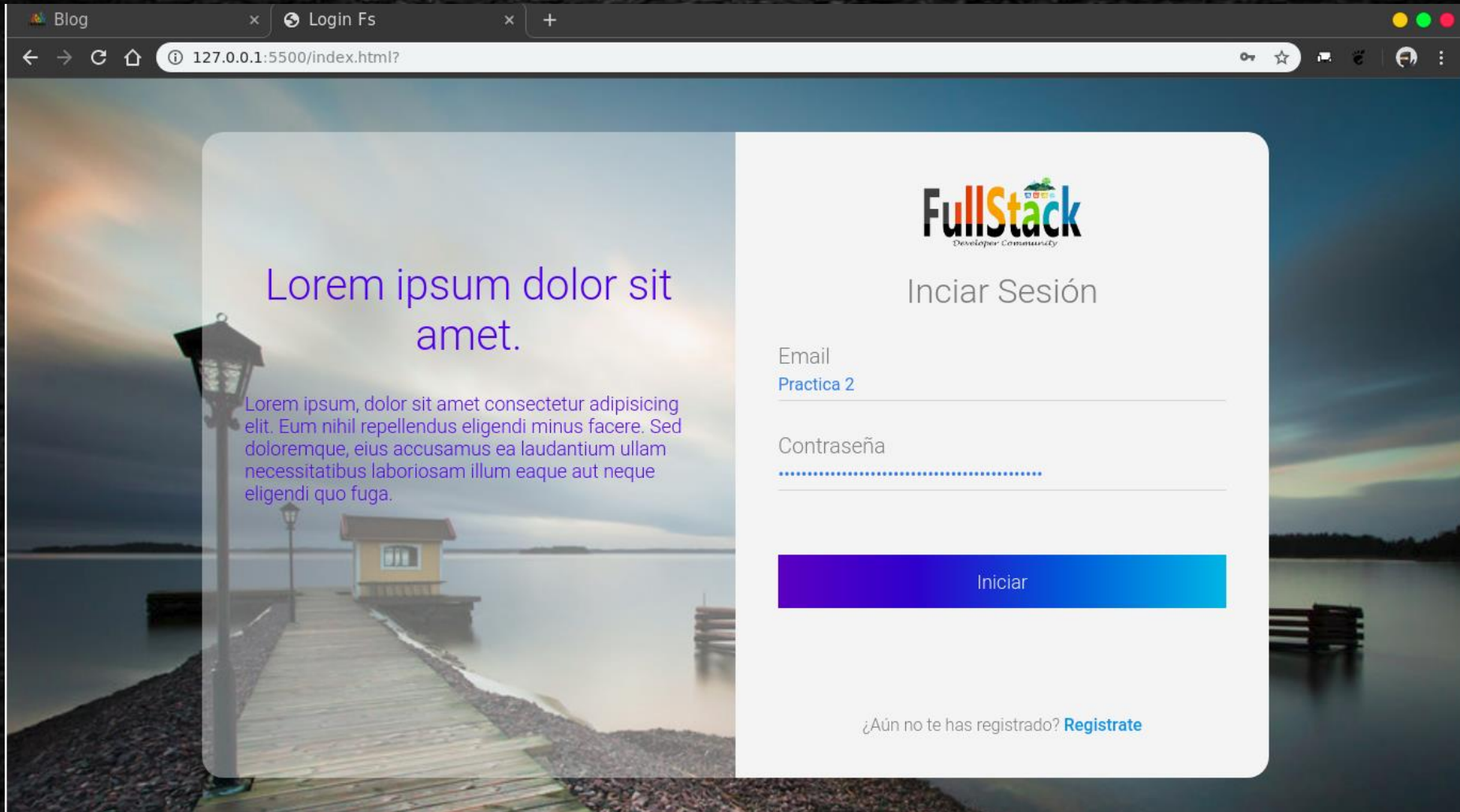
Tarea 1



ayuda

background: linear-gradient(90deg, rgb(58, 34, 191), rgb(15, 185, 227));

Tarea 2



Opcional

