*Actividad practica implemente los códigos de ejemplo capture una imagen del resultado y elabore una presentación con el código y el resultado de su aplicación los documentos de apoyo son entregados por el instructor

*implemente lo aprendido en la guía en los diseños web de la consulta de javascript

Outline

La propiedad **outline** es una vieja propiedad CSS que ha sido expandida en CSS3 para incluir un valor de desplazamiento. Esta propiedad era usada para crear un segundo borde, y ahora ese borde puede ser mostrado alejado del borde real del elemento.

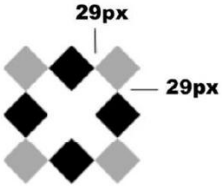
```
#principal {
  display: block;
  width: 500px;
  margin: 50px auto;
  padding: 15px;
  text-align: center;
  border: 1px solid #999999;
  background: #DDDDDD;

  outline: 2px dashed #000099;
  outline-offset: 15px;
}
```

Border-image

Los posibles efectos logrados por las propiedades border y outline están limitados a líneas simples y solo algunas opciones de configuración. La nueva propiedad **border-image** fue incorporada para superar estas limitaciones y dejar en manos del diseñador la calidad y variedad de bordes disponibles ofreciendo la alternativa de utilizar imágenes propias.

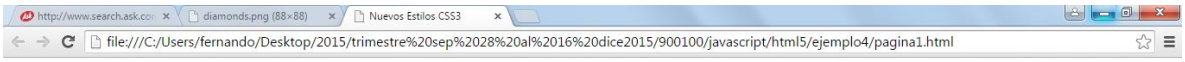
utilizar una imagen PNG que incluye diamantes **diamonds.png**

La propiedad **border-image** toma una imagen y la utiliza como patrón. De acuerdo a los valores otorgados, la imagen es cortada como un pastel, las partes obtenidas son luego ubicadas alrededor del objeto para construir el borde.



Para hacer el trabajo, necesitamos especificar tres atributos: el nombre del archivo de la imagen, el tamaño de las piezas que queremos obtener del patrón y algunas palabras clave para declarar cómo las piezas serán distribuidas alrededor del objeto.

```
#principal {
  display: block;
  width: 500px;
  margin: 50px auto;
  padding: 15px;
  text-align: center;

  border: 29px;
  -moz-border-image: url("diamonds.png") 29 stretch;
  -webkit-border-image: url("diamonds.png") 29 stretch;
  border-image: url("diamonds.png") 29 stretch;
}
```



Nota compruébelo en diferentes navegadores

Transform y transition

Los elementos HTML, cuando son creados, son como bloques sólidos e inamovibles. Pueden ser movidos usando código Javascript o aprovechando librerías populares como jQuery (*www.jquery.com*), por ejemplo, pero no existía un procedimiento estándar para este propósito hasta que CSS3 presentó las propiedades **transform** y **transition**.
Ahora ya no tenemos que pensar en cómo hacerlo. En su lugar, solo tenemos que conocer cómo ajustar unos pocos parámetros y nuestro sitio web puede ser tan flexible y dinámico como lo imaginamos.
La propiedad **transform** puede operar cuatro transformaciones básicas en un elemento: **scale** (escalar), **rotate** (rotar), **skew** (inclinar) y **translate** (trasladar o mover). Veamos cómo funcionan:

Transform: scale

```
#principal {
  display: block;
  width: 500px;
  margin: 50px auto;
  padding: 15px;
  text-align: center;
  border: 1px solid #999999;
  background: #DDDDDD;

  -moz-transform: scale(2);
  -webkit-transform: scale(2);
}
```

aplicamos transformación duplicando la escala del
elemento. La función **scale** recibe dos parámetros: el valor **X** para la escala horizontal y el valor **Y** para la escala vertical. Si solo un valor es provisto el mismo valor es aplicado a ambos parámetros.

```
#principal {
  display: block;
  width: 500px;
  margin: 50px auto;
  padding: 15px;
  text-align: center;
  border: 1px solid #999999;
  background: #DDDDDD;

  -moz-transform: scale(1,-1);
  -webkit-transform: scale(1,-1);
}
```

dos parámetros han sido declarados para cambiar la escala de la
caja **principal**. El primer valor, 1, mantiene la proporción original para la dimensión horizontal de la caja. El segundo valor también mantiene la proporción original, pero invierte el elemento verticalmente para producir el efecto espejo.

Transform: rotate

La función **rotate** rota el elemento en la dirección de las agujas de un reloj. El valor debe ser especificado en grados usando la unidad "**deg**":

```
    display: block;
    width: 500px;
    margin: 50px auto;
    padding: 15px;
    text-align: center;
    border: 1px solid #999999;
    background: #DDDDDD;

    -moz-transform: rotate(30deg);
    -webkit-transform: rotate(30deg);
  }
```

Transform: skew

Esta función cambia la simetría del elemento en grados y en ambas dimensiones.

```
#principal {
  display: block;
  width: 500px;
  margin: 50px auto;
  padding: 15px;
  text-align: center;
  border: 1px solid #999999;
  background: #DDDDDD;

  -moz-transform: skew(20deg);
  -webkit-transform: skew(20deg);
}
```

La función **skew** usa dos parámetros, pero a diferencia de otras funciones, cada parámetro de esta función solo afecta una dimensión (los parámetros actúan de forma independiente). realizamos una operación **transform** a la caja de la cabecera para inclinarla. Solo declaramos el primer parámetro, por lo que solo la dimensión horizontal de la caja será modificada. Si usáramos los dos parámetros, podríamos alterar ambas dimensiones del objeto. Como alternativa podemos utilizar funciones diferentes para cada una de ellas: **skewX** y **skewY**.

Transform: translate

Similar a las viejas propiedades **top** y **left**, la función **translate** mueve o desplaza el elemento en la pantalla a una nueva posición.

```
#principal {
  display: block;
  width: 500px;
  margin: 50px auto;
  padding: 15px;
  text-align: center;
  border: 1px solid #999999;
  background: #DDDDDD;

  -moz-transform: translate(100px);
  -webkit-transform: translate(100px);
}
```

La función **translate** considera la pantalla como una grilla de pixeles, con la posición original del elemento usada como un punto de referencia. La esquina superior izquierda del elemento es la posición **0,0**, por lo que valores negativos moverán al objeto hacia la izquierda o hacia arriba de la posición original, y valores positivos lo harán hacia la derecha o hacia abajo.

Transformando todo al mismo tiempo

A veces podría resultar útil realizar sobre un elemento varias transformaciones al mismo tiempo. Para obtener una propiedad **transform** combinada, solo tenemos que separar cada función a aplicar con un espacio:

```
  width: 500px;
  margin: 50px auto;
  padding: 15px;
  text-align: center;
  border: 1px solid #999999;
  background: #DDDDDD;

  -moz-transform: translateY(100px) rotate(45deg) scaleX(0.3);
  -webkit-transform: translateY(100px) rotate(45deg) scaleX(0.3);
}
```

Una de las cosas que debe recordar en este caso es que el orden es importante. Esto es debido a que algunas funciones mueven el punto original y el centro del objeto, cambiando de este modo los parámetros que el resto de las funciones utilizarán para operar.

Transformaciones dinámicas

Lo que hemos aprendido hasta el momento es cambiará la forma de la web, pero la mantendrá tan estática como siempre. Sin embargo, podemos aprovecharnos de la combinación de transformaciones y pseudo clases para convertir nuestra página en una

aplicación dinámica:

```
#principal {
  display: block;
  width: 500px;
  margin: 50px auto;
  padding: 15px;
  text-align: center;
  border: 1px solid #999999;
  background: #DDDDDD;
}
#principal:hover{
  -moz-transform: rotate(5deg);
  -webkit-transform: rotate(5deg);
}
```

usando la vieja pseudo clase **:hover**. El resultado obtenido es que cada vez que el puntero del ratón pasa sobre esta caja, la propiedad **transform** rota la caja en 5 grados, y cuando el puntero se aleja la caja vuelve a rotar de regreso a su posición original.

Transiciones

De ahora en más, hermosos efectos usando transformaciones dinámicas son accesibles y fáciles de implementar. Sin embargo, una animación real requiere de un proceso de más de dos pasos.

La propiedad **transition** fue incluida para suavizar los cambios, creando mágicamente el resto de los pasos que se encuentran implícitos en el movimiento. Solo agregando esta propiedad forzamos al navegador a tomar cartas en el asunto, crear para nosotros todos esos pasos invisibles, y generar una transición suave desde un estado al otro.

```
#principal {
  display: block;
  width: 500px;
  margin: 50px auto;
  padding: 15px;
  text-align: center;
  border: 1px solid #999999;
  background: #DDDDDD;

  -moz-transition: -moz-transform 1s ease-in-out 0.5s;
  -webkit-transition: -webkit-transform 1s ease-in-out 0.5s;
}
#principal:hover{
  -moz-transform: rotate(5deg);
  -webkit-transform: rotate(5deg);
}
```

Como puede ver la propiedad **transition** puede tomar hasta cuatro parámetros separados por un espacio. El primer valor es la propiedad que será considerada para hacer la transición (en nuestro ejemplo elegimos **transform**). Esto es necesario debido a que varias propiedades pueden cambiar al mismo tiempo y probablemente necesitemos crear los pasos del proceso de transición solo para una de ellas. El segundo parámetro especifica el tiempo que la transición se tomará para ir de la posición inicial a la final. El tercer parámetro puede ser cualquiera de las siguientes palabras clave: **ease**, **linear**, **ease-in**, **ease-out** o **ease-in-out**. Estas palabras clave determinan cómo se realizará el proceso de transición basado en una curva Bézier. Cada una de ellas representa diferentes tipos de curva Bézier, y la mejor forma de saber cómo trabajan es viéndolas funcionar en pantalla. El último parámetro para la propiedad **transition** es el retardo. Éste indica cuánto tiempo tardará la transición en comenzar. Para producir una transición para todas las propiedades que están cambiando en un objeto, la palabra clave **all** debe ser especificada. También podemos declarar varias propiedades a la vez listándolas separadas por coma.