

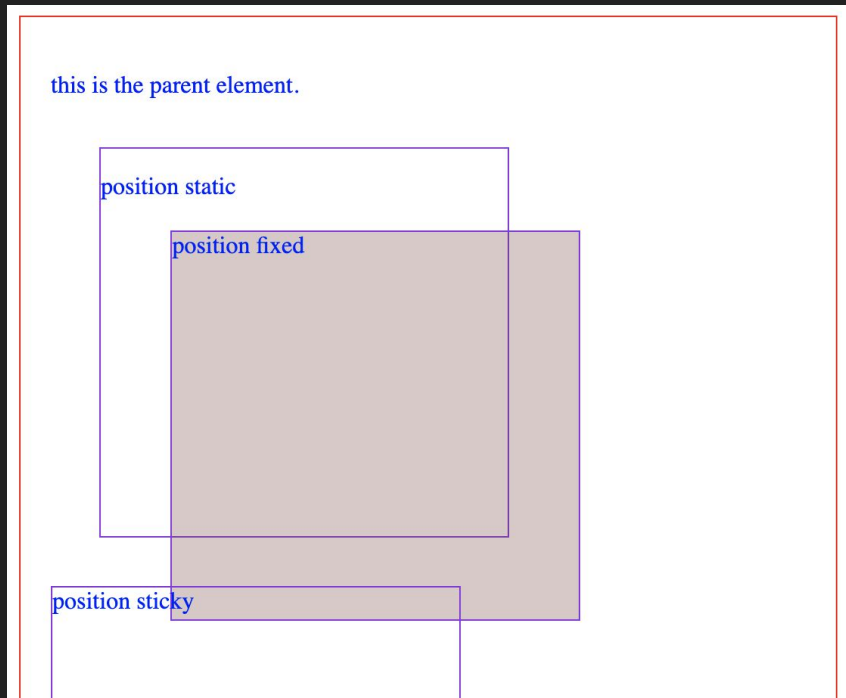
# Profundización CSS

Más propiedades, pseudo-classes & pseudo-elements

# Herencia

Algunas propiedades de CSS pueden ser heredadas por elementos que están contenidos en otros, como por ejemplo la propiedad de color.

Podemos usar la palabra `inherit` cuando queremos especificar que una propiedad debe ser heredada del elemento padre.



# Position

La propiedad position indica la forma en que el elemento se va a posicionar en la pantalla, nos permite usarla en gran variedad de situaciones.

**Static:** Es el comportamiento normal de los elementos, encaja en el layout.

**Absolute:** el elemento sale del layout y se posiciona en cualquier lugar dándole coordenadas.

**Fixed:** El elemento se posiciona en cualquier parte de la pantalla y se mantiene ahí.

**Sticky:** el elemento está en el lugar que corresponde en el layout pero al llegar a una posición asignada se queda ahí.

**Relative:** se le asigna un lugar específico tomando como referencia la posición normal del elemento.

```
.static {  
  margin: 2rem;  
  position: static;  
}  
  
.absolute {  
  position: absolute;  
  top: 45vh;  
  left: 40px;  
  background-color: rgba(78, 182, 182, 0.192);  
}  
  
.fixed {  
  position: fixed;  
  top: 150px;  
  left: 10vw;  
  background-color: rgba(119, 60, 60, 0.281);  
}  
  
.sticky {  
  position: -webkit-sticky;  
  position: sticky;  
  top: -1px;  
}  
  
.relative {  
  position: relative;  
  top: 30px;  
  border: 1px solid darkorange;  
}
```

# Pseudo-Elementos

Un pseudo-elemento es un keyword que agregamos en css después de un selector, lo usamos para seleccionar una parte del elemento que especificamos, generalmente los usamos para agregar estilos que ayuden a modificar la forma en la que se ve un elemento.

Usamos `::` para indicar que vamos a usar un pseudo-elemento.



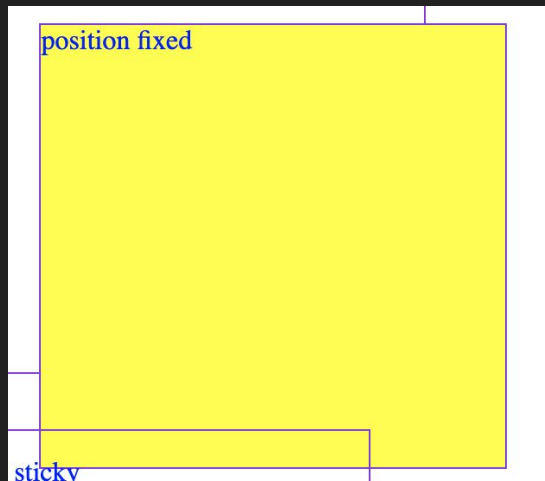
```
.sticky::first-letter {  
    color: ■ gold;  
    font-size: 2rem;  
}
```

# Pseudo-classes

Una pseudo-clase también se usa después de los selectores, las usamos para indicar estados especiales de los elementos y cambiar los estilos a el elemento cuando eso pasa

Usamos `:` para indicar que vamos a usar un pseudo-elemento.

```
.fixed: hover {  
    background-color: yellow;  
}
```



# Un poco de git

**Pull:** lo usamos para traer los cambios nuevos del repositorio remoto a nuestro local.

**Push:** lo usamos para enviar nuestros cambios locales al repo remoto, tenemos que hacer un commit primero.

**Commit:** es un snapshot del proyecto en ese momento, podemos tener todos los cambios o solo algunos.

**Add:** con este comando agregamos los cambios que deseamos en el commit.

**Pull request:** es la forma en la que solicitamos al owner o maintainer de un repositorio que lleve tus cambios a su repo.

**Branch:** es una rama de nuestro proyecto, normalmente las usamos para trabajar en algo de forma independiente de las otras personas.

**Merge:** es el acto de mezclar dos ramas, llevando así todos los commits de una rama hacia otra.

**Checkout:** Es la forma en que cambiamos de rama o cambiamos los archivos locales.

# Ejemplo de Git

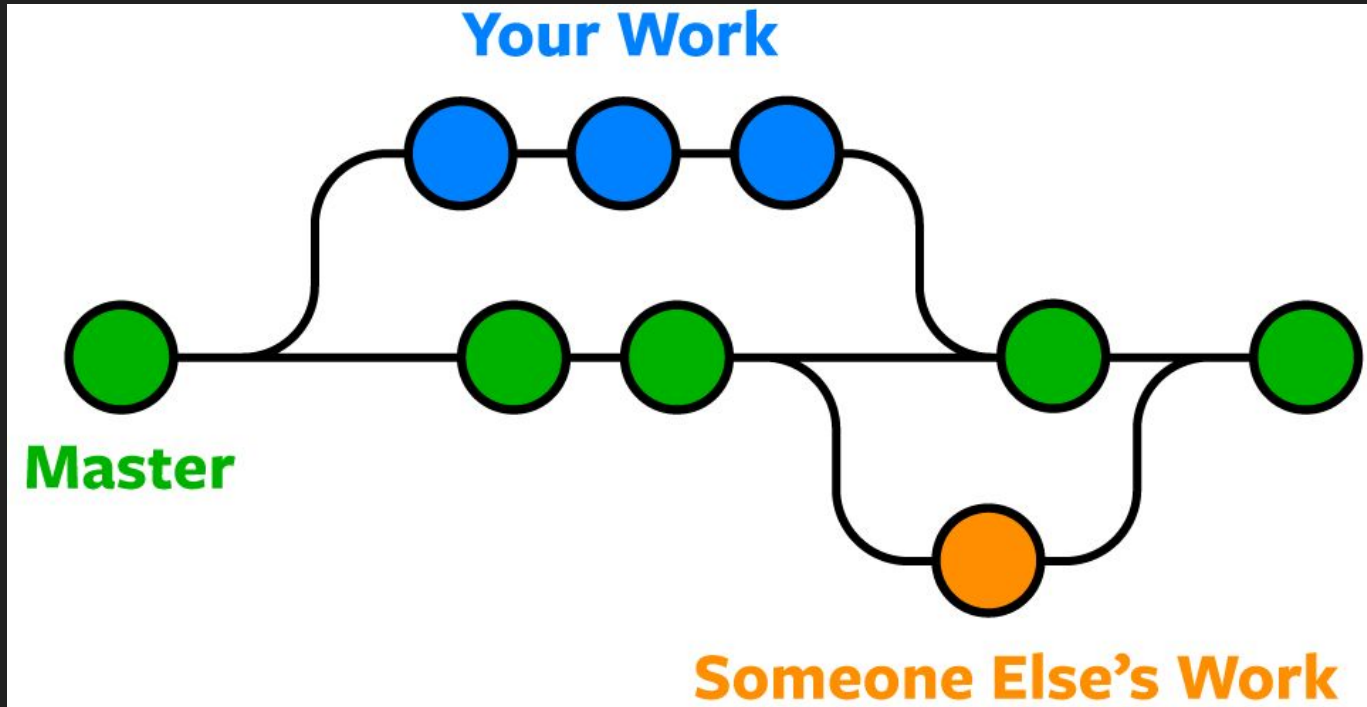


Imagen obtenida de:

<https://www.nobledesktop.com/image/gitresources/git-branches-merge.png>

# Ejemplo de Git

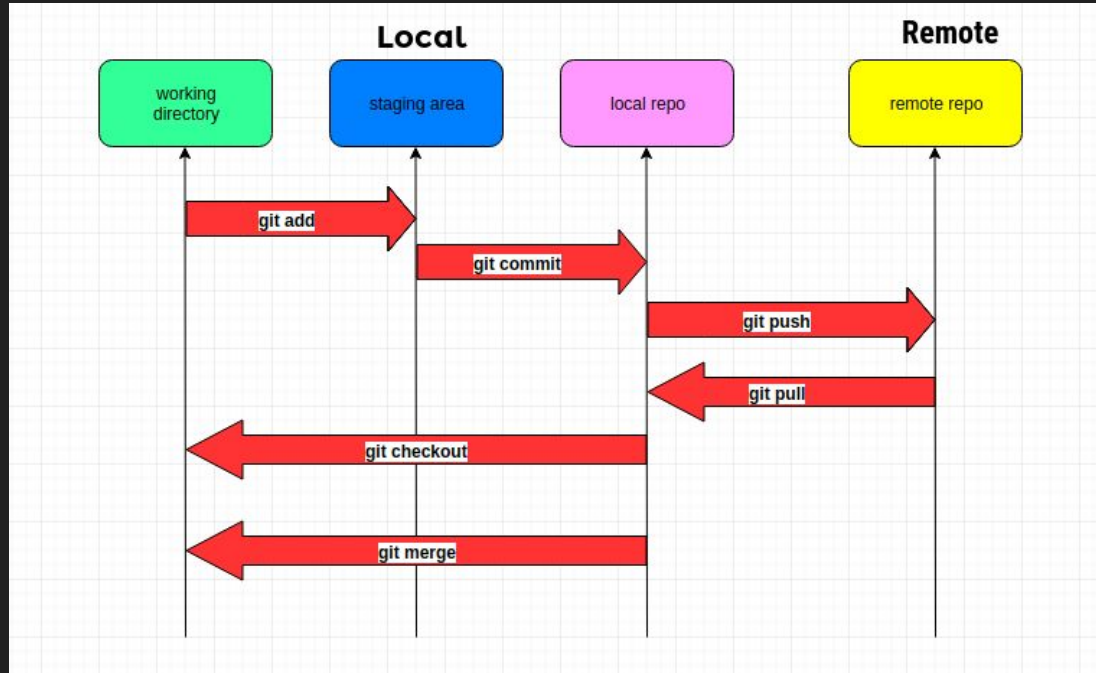


Imagen obtenida de: <https://techrish.com/wp-content/uploads/2019/08/git.png>