

Una de las características más importantes de la programación orientada a objetos es la de la herencia. Ésta consiste en que, cuando una clase **hereda** a otra, obtiene por defecto todas sus propiedades (métodos y atributos), pudiendo definir algunas adicionales. Esto se expresa mediante la palabra clave **extends** y el nombre de la clase padre, al definir una clase hija que hereda de ésta.

Las propiedades que tengan la visibilidad de **privada** no serán accesibles desde esta nueva clase, aunque seguirán existiendo dentro del objeto.

Esto nos permite definir inicialmente unas clases más genéricas, que sirvan de base para muchas otras clases más específicas que compartan ese código en común.

En cualquier momento podemos sobrescribir las propiedades del método padre, así como acceder directamente a las mismas mediante la palabra reservada **parent**.

```
class HTMLTag {
    public $tag = 'input';
    public $attributes = array();
    public function __toString() {
        $result = '<'.$tag.' ';
        foreach($attributes as $key => $value ) {
            $result .= $key .'="'.$value.'" ';
        }
        $result .= '>';
        return $result;
    }
    public function setAttribute($attributeName, $value) {
        $this->attributes[$attributeName] = $value;
    }
}

class HTMLInputText extends HTMLTag {
    public $tag = 'input';
    public $attributes = array( 'type'=>'text');
    public function __toString() {
        $result = parent::__toString();
        $result .= '</input>';
    }
}
```