

Al definir unas clases que son utilizadas por otras, podemos establecer que una de ellas es **abstracta**, queriendo significar que solo es utilizada como base para definir otras clases, y que no puede ser instanciada directamente.

También podemos definir una función como **abstracta**, indicando únicamente su **signatura** (visibilidad, nombre y parámetros). Con esto establecemos que cualquier clase hija de ésta, debe concretizar el código de dicha función, o de lo contrario seguir siendo considerada abstracta.

Por último podemos definir **interfaces**, que son una especie de definición de métodos de clase. Si una clase incluye todos los métodos de una interfaz según su signatura, entonces se dice que **implementa dicha interfaz**, y un objeto esa clase puede ser admitida siempre que se esté pidiendo un objeto con un interfaz dado. Nótese que la clase en realidad puede tener definida muchas más propiedades, y que puede implementar más de una interfaz.

En la definición de la clase (después de indicar a qué clase hereda si es que es hija de alguna) definiremos mediante la palabra reservada **implements** los interfaces que la clase implementa separados por comas.

```
interface Printable {
    function __toString();
}

interface Atributable {
    function setAttribute($attributeName, $value);
}

abstract class HTMLInputAbstract extends HTMLTag implements Printable, Atributable {
    abstract function setAttribute($attributeName, $value);
    public $tag = 'input';
}

class HTMLInputCheckbox extends HTMLTag implements Printable, Atributable {
    public $attributes = array( 'type'=>'checkbox');
}
```