Directives are something to tell vuejs to do something. For example, v-if, v-for and etc...

(They start with v-) these directives listed above are already existed in vue, and if vue does not have a directive we want to use, we can customize our own.

So how do we create a directive such as <v-colorful>, which the functionality is to change color of the title of the blog? We can create globally by typing globally in main.js:

Vue.directive(‘colorful’, {

Bind(el,binding,vnode){

el.style.color = “#”+ Math.random().toString().slice(2,8);

}

})

Just like to create a new component, it has life cycles like created (), destroyed () … Directives also have their life cycles. For directive it has a life cycle hook called bind (), it will be activated once the directive has been bound to an element tag. It contains three parameters; the first parameter refers to the whole element for example the html line: <h4 v-colorful>XXXX</h4>. The second parameter is the name of the directive name, for example <h4 v-colorful=” red”>XXXX</h4>. The third parameter is the virtual node.

We can do so many things we directives. We may also change the theme of the page, like change the width for the blogs…

Vue.directive(‘theme, {

Bind(el,binding,vnode){

If (binding.value ==” wide”) {

el.style.maxWidth=”1200px”;

}

})

And we should use it as <div v-theme=” ‘wide’”>XXXX</div>.

Another thing we can notice is that we can put argument after the directives. The arguments are something placed after a column. Such as <div v-theme: abc=” ‘wide’”>XXXX</div>.

We can create custom arguments by writing

Vue.directive(‘theme, {

Bind(el,binding,vnode){

If(binding.arg == ‘abc’){

Do…

}

}

})

How to register directive locally in a specific component?

directives: {

'colorful’: {

bind(el,binding,vnode){

el.style.color = "#" + Math.random().toString().slice(2,8);

}

}

}