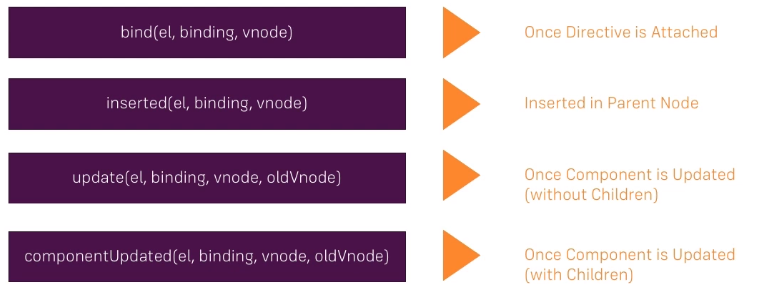
**Section 34 – Using And Creating Directives**

Chapter 489 – Chapter 500

1. Hook Function



1. Arg, modifiers, and delayed

<div class="row">

            <div class="col-xs-12 col-sm-8 col-sm-offset-2 col-md-6 col-md-offset-3">

                <h1>Custom Directives</h1>

                <p v-highlight:background="'red'">Color this</p>

                <p v-highlight="'red'">Color that</p>

                <p v-highlight:background.delayed="'red'">Color delayed</p>

            </div>

</div>

Vue.directive('highlight', {

  bind(el, binding, vnode) {

    // el.style.backgroundColor = 'green';

    // el.style.backgroundColor = binding.value;

    var delay = 0;

    if (binding.modifiers['delayed']) {

      delay = 3000;

    }

    setTimeout(() => {

      if (binding.arg == 'background') {

        el.style.backgroundColor = binding.value;

      } else {

        el.style.color = binding.value;

      }

    }, delay);

  }

});

1. Local-highlight

<div class="row">

            <div class="col-xs-12 col-sm-8 col-sm-offset-2 col-md-6 col-md-offset-3">

                <h1>Custom Directives</h1>

                <p v-highlight:background="'red'">Color this</p>

                <p v-highlight="'red'">Color that</p>

                <p v-highlight:background.delayed="'red'">Color delayed</p>

                <p v-local-highlight:background.delayed="'red'">Color this, too</p>

            </div>

        </div>

<script>

export default {

    directives: {

        'local-highlight': {

            bind(el, binding, vnode) {

                var delay = 0;

                if (binding.modifiers['delayed']) {

                delay = 3000;

                }

                setTimeout(() => {

                if (binding.arg == 'background') {

                    el.style.backgroundColor = binding.value;

                } else {

                    el.style.color = binding.value;

                }

                }, delay);

            }

        }

    }

}

</script>

1. Multiple modifiers, delayed and blink

<div class="row">

            <div class="col-xs-12 col-sm-8 col-sm-offset-2 col-md-6 col-md-offset-3">

                <h1>Custom Directives</h1>

                <p v-highlight:background="'red'">Color this</p>

                <p v-highlight="'red'">Color that</p>

                <p v-highlight:background.delayed="'red'">Color delayed</p>

                <p v-local-highlight:background.delayed.blink="'red'"> Color this, too</p>

            </div>

</div>

<script>

export default {

    directives: {

        'local-highlight': {

            bind(el, binding, vnode) {

                var delay = 0;

                if (binding.modifiers['delayed']) {

                delay = 3000;

                }

                if (binding.modifiers['blink']) {

                    let mainColor = binding.value;

                    let secondColor = 'blue';

                    let currentColor = mainColor;

                    setTimeout(() => {

                        setInterval(() => {

                            currentColor == secondColor ? currentColor = mainColor : currentColor = secondColor;

                            if (binding.arg == 'background') {

                                el.style.backgroundColor = currentColor;

                            } else {

                                el.style.color = secondColor;

                            }

                        }, 1000);

                    }, delay);

                } else {

                    setTimeout(() => {

                        if (binding.arg == 'background') {

                            el.style.backgroundColor = binding.value;

                        } else {

                            el.style.color = binding.value;

                        }

                    }, delay);

                }

            }

        }

    }

}

</script>

1. Passing more complex values to directives

<div class="row">

            <div class="col-xs-12 col-sm-8 col-sm-offset-2 col-md-6 col-md-offset-3">

                <h1>Custom Directives</h1>

                <p v-highlight:background="'red'">Color this</p>

                <p v-highlight="'red'">Color that</p>

                <p v-highlight:background.delayed="'red'">Color delayed</p>

                <p v-local-highlight:background.delayed.blink="'red'">Color this, too</p>

                <p v-local-highlight:background.delayed.blink="{mainColor: 'red', secondColor: 'green', delay: 500}">Color this, too</p>

            </div>

</div>

<script>

export default {

    directives: {

        'local-highlight': {

            bind(el, binding, vnode) {

                var delay = 0;

                if (binding.modifiers['delayed']) {

                delay = 3000;

                }

                if (binding.modifiers['blink']) {

                    // let mainColor = binding.value;

                    let mainColor = binding.value.mainColor;

                    // let secondColor = 'blue';

                    let secondColor = binding.value.secondColor;

                    let currentColor = mainColor;

                    setTimeout(() => {

                        setInterval(() => {

                            currentColor == secondColor ? currentColor = mainColor : currentColor = secondColor;

                            if (binding.arg == 'background') {

                                el.style.backgroundColor = currentColor;

                            } else {

                                el.style.color = secondColor;

                            }

                        }, binding.value.delay);

                        // }, 1000);

                    }, delay);

                } else {

                    setTimeout(() => {

                        if (binding.arg == 'background') {

                            // el.style.backgroundColor = binding.value;

                            el.style.backgroundColor = binding.value.mainColor;

                        } else {

                            // el.style.color = binding.value;

                            el.style.color = binding.value.mainColor;

                        }

                    }, delay);

                }

            }

        }

    }

}

</script>