* Documentação Oficial - Diretivas Personalizadas: <https://br.vuejs.org/v2/guide/custom-directive.html>

USANDO E CRIANDO DIRETIVAS

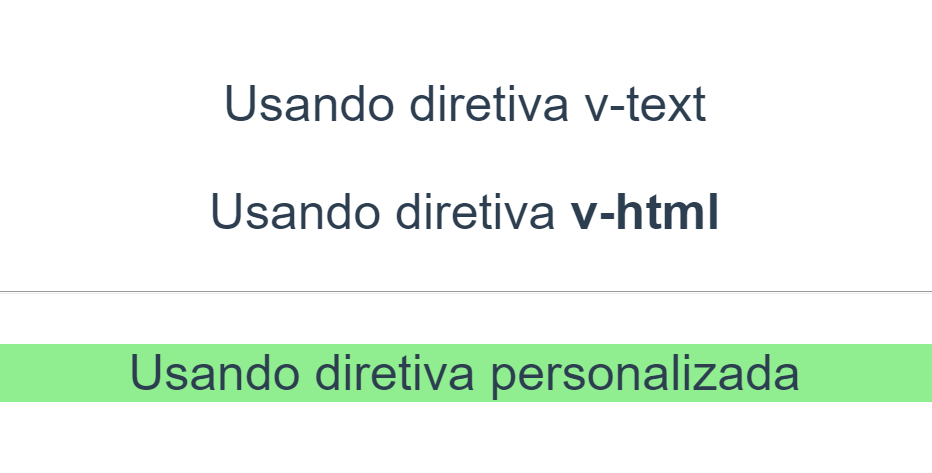
1. **Aula 183 – Entendendo diretivas**
2. V-text - Com esta diretiva podemos passar texto simples entre aspas simples.
3. V-html – Podemos passar código html
4. <p *v-text*="'Usando diretiva v-text'"></p>

<p *v-html*="'Usando diretiva <strong>v-html</strong>'" ></p>

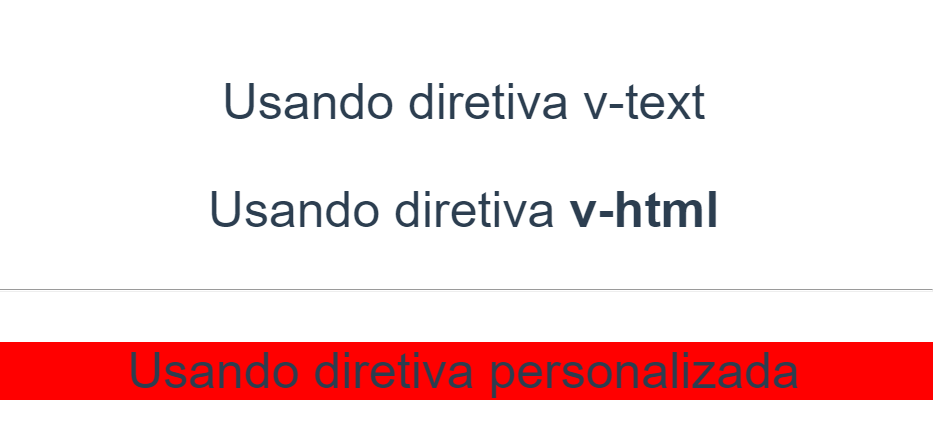
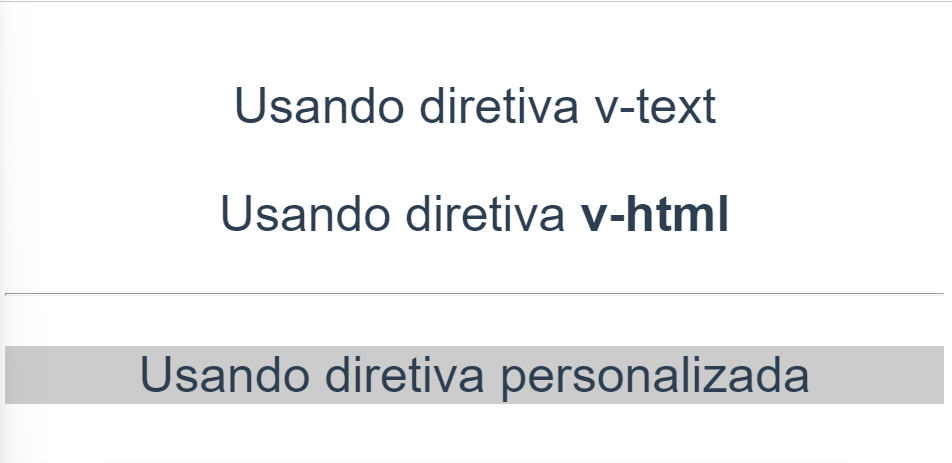
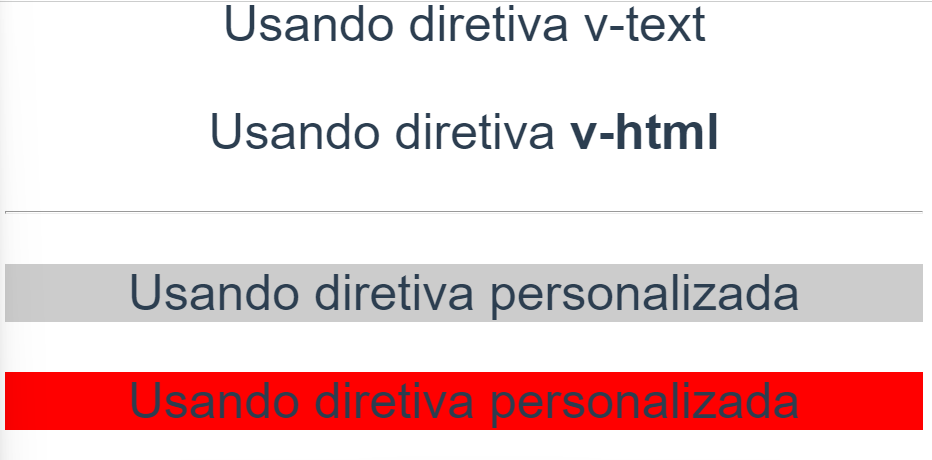
1. Criando diretivas globais.
2. No arquivo main.js
3. Criamos a diretiva utilizando Vue.directive passando para ela o nome da diretiva e um objeto. Dentro deste objeto deve-se utilizar um método do ciclo de vida que são métodos gatilhos ou métodos hooks.

**Aula 184 – Funções gatilho ou hooks**

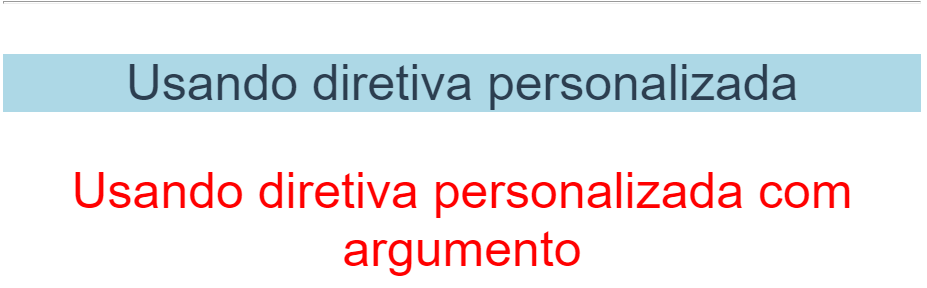
1. Veremos cada método do tipo hooks, seus atributos e quando estes métodos são chamados.
2. Hooks:
3. Criamos a diretiva utilizando Vue.directive passando para ela o nome da diretiva e um objeto. Dentro deste objeto deve-se utilizar um método do ciclo de vida que são métodos gatilhos ou métodos hooks.
4. Vue.directive('destaque', {
6. })
7. Tipos de Hooks:
8. bind(el, binding, vnode): Quando a diretiva é interligada com elemento
9. inserted(el, binding, vnode): Elemento inserido no nó pai.
10. update(el, binding, vnode, oldVNode): Quando o componente é atualizado (antes filhos)
11. componentUpdate(el, binding, vnode, oldVNode): Quando o componente é atualizado (depois filhos)
12. unbind(el, binding, vnode): Quando a diretiva é desvinculada do elemento
13. Aula 185 – Criando uma Diretiva simples
14. Essa diretiva irá utilizar o método bind() e irá carregar um estilo de cor de fundo no texto aplicado à diretiva.
15. Vue.directive('destaque', {
16. bind(*el*, *binding*, *vnode*){
17. *el*.style.backgroundColor = 'Lightgreen'
18. }
19. })
20. Agora vamos no arquivo App e iremos incluir a nova diretiva
21. <p *v-destaque*>Usando diretiva personalizada</p>



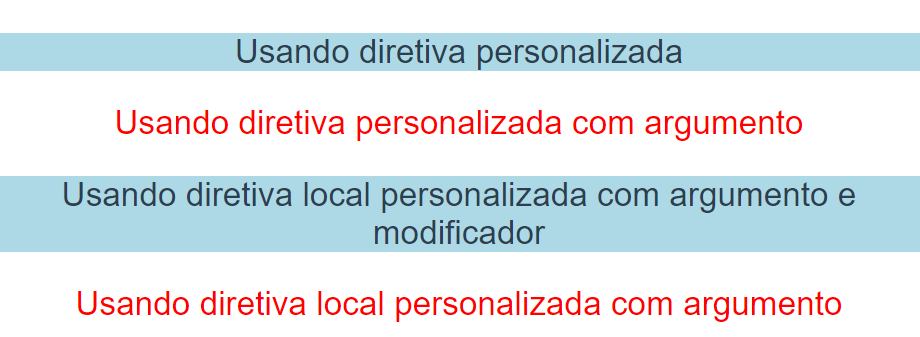
**Aula 186 – Passando valor para diretiva personalizada**

1. Na diretiva anterior v-destaque que criamos, vamos substituir a cor passada dentro da diretiva e iremos adicionar o ‘binding.value’ que será o parâmetro de cor que será passado na diretiva. Sendo que o valor deve ser passado entre aspas simples.
2. Vue.directive('destaque', {
3. bind(*el*, *binding*, *vnode*){
4. // el.style.backgroundColor = 'Lightgreen'
5. *el*.style.backgroundColor = *binding*.value
6. }
7. })
8. No arquivo App
9. <p *v-destaque*="'red'">Usando diretiva personalizada</p>
10. 
11. <p *v-destaque*="'#ccc'">Usando diretiva personalizada</p>
12. 
13. O valor passado para a diretiva pode ser um objeto ou referência que está dentro da instância. Por exemplo no atributo data() vamos incluir uma variável chamada cor e vamos passar o valor de vermelho.
14. export default {
15. data() {
16. return {
17. cor: 'red'
18. }
19. },
20. }
21. Agora basta passarmos a variável cor na diretiva, sem as aspas simples.
22. <p *v-destaque*="cor">Usando diretiva personalizada</p>
23. 

**Aula 187 – Passando argumento para diretiva personalizada**

1. O argumento da diretiva vem depois de : após a diretiva e os modificadores vem depois do argumento, precedidos de ponto.
2. Ex: <p v-teste:argumento.mod1.mod2.mod3=”’valor’”></p>.
3. Para exemplificar iremos criar um argumento chamado ‘fundo’ que irá aplicar a cor de fundo e caso não tenha o argumento a cor será aplicada na fonte.
4. No arquivo main. Vamos alterar a diretiva destaque.
5. Vue.directive('destaque', {
6. bind(*el*, *binding*, *vnode*){
7. // el.style.backgroundColor = 'Lightgreen'
8. // el.style.backgroundColor = binding.value
9. if(*binding*.arg === 'fundo'){
10. *el*.style.backgroundColor = *binding*.value
11. }else{
12. *el*.style.color = *binding*.value
13. }
14. }
15. })
16. No arquivo app.js iremos alterar a diretiva incluindo o fundo como argumento.
17. <p *v-destaque:fundo*="'lightblue'">Usando diretiva personalizada</p>
18. <p *v-destaque*="cor">Usando diretiva personalizada com argumento</p>
19. 

**Aula 188 – modificando diretivas personalizadas com modificadores**

1. Neste caso iremos criar um modificador que irá atrasar a alteração da cor de fundo em 3 segundos.
2. No arquivo app.js, vamos incluir o modificador atrasar na diretiva.
3. <p *v-destaque:fundo*.*atrasar*="'lightblue'">Usando diretiva personalizada</p>
4. No arquivmo main.js
5. Na diretiva ‘destaque’ iremos criar uma variável atraso que irá receber o valor e 0.
6. Iremos verificar se existe o modificador atrasar e neste caso seta 3000 milisegundos na variável atraso.
7. Após isso utilizamos a função setTimeout para executar o atraso antes da execução do código da diretiva.
8. Vue.directive('destaque', {
9. bind(*el*, *binding*, *vnode*){
10. // el.style.backgroundColor = 'Lightgreen'
11. // el.style.backgroundColor = binding.value
12. let atraso = 0
13. if(*binding*.modifiers['atrasar']) atraso = 3000
14. setTimeout(() => {
15. if(*binding*.arg === 'fundo'){
16. *el*.style.backgroundColor = *binding*.value
17. }else{
18. *el*.style.color = *binding*.value
19. }
20. }, atraso);
21. }
22. })
23. **Aula 190 – Registrando diretivas localmente.**
24. Dentro do export default() do componente vamos declarar o atributo directives: {} e dentro dele passamos a direva como fazemos de forma global.
25. No arquivo app.js
26. Vamos criar a diretiva ‘destaque-local’.
27. export default {
28. directives:{
29. 'destaque-local' : {
30. bind(*el*, *binding*, *vnode*){
31. // el.style.backgroundColor = 'Lightgreen'
32. // el.style.backgroundColor = binding.value
33. let atraso = 0
34. if(binding.modifiers['atrasar']) atraso = 3000
35. setTimeout(() => {
36. if(binding.arg === 'fundo'){
37. el.style.backgroundColor = binding.value
38. }else{
39. el.style.color = binding.value
40. }
41. }, atraso);
42. }
43. }
44. },
45. Ainda no arquivo app.js vamos duplicar os parágrafos das diretivas alterando para diretiva-local.
46. <p *v-destaque:fundo*.*atrasar*="'lightblue'">Usando diretiva personalizada</p>
47. <p *v-destaque*="cor">Usando diretiva personalizada com argumento</p>
48. <p *v-destaque-local:fundo*.*atrasar*="'lightblue'">Usando diretiva local personalizada com argumento e modificador</p>
49. <p *v-destaque-local*="cor">Usando diretiva local personalizada com argumento</p>
50. O resultado será o mesmo
51. 
52. Existe a opção também de criarmos um arquivo externo com as diretivas que iremos utilizar e importa-las quando necessário.

**Aula 191 – Utilizando múltiplos modificadores**

1. Vamos fazer outro modificador para vermos dois modificadores endo aplicados em uma única diretiva.
2. Para isso o modificador irá alternar entre duas cores, vamos acrescentar outro modificador chamado ‘alternar’ na diretiva.
3. <p *v-destaque-local:fundo*.atrasar.*alternar*="'lightblue'">Usando diretiva local personalizada com argumento e modificador</p>
4. <p *v-destaque-local*="cor">Usando diretiva local personalizada com argumento</p>
5. <p *v-destaque-local*.atrasar.*alternar*="cor">Usando diretiva local personalizada com argumento</p>
6. Agora vamos acrescentar na diretiva local uma função que irá alterar a cor do fundo ou da fonte de acordo com o argumento passado.
7. Essa função será chamada dentro de setTimeout para aplicar a cor passada por parâmetro após 3segundos.
8. Agora vamos criar três variáveis de cor que serão utilizadas para alternar as cores.
9. Sendo uma cor passada por parâmetro, outra passada diretamente e outra que irá receber o valor da cor1.
10. Após isso vamos incluir uma verificação dentro de setTimeout() onde iremos verificar se foi passado o modificador ‘alternar’
11. Dentro destra verificação iremos incluir um setInterval() que irá alternar entre as cores e aplica-las no intervalo de 1 segundo.
12. export default {
13. directives:{
14. 'destaque-local' : {
15. bind(*el*, *binding*, *vnode*){
16. const aplicarCor = *cor* => {
17. if(binding.arg === 'fundo'){
18. el.style.backgroundColor = cor
19. }else{
20. el.style.color = cor
21. }
22. }
23. let atraso = 0
24. if(*binding*.modifiers['atrasar']) atraso = 3000
25. const cor1 = binding.value
26. const cor2 = 'purple'
27. let corAtual = cor1
28. setTimeout(() => {
29. if(binding.modifiers['alternar']) {
30. setInterval(()=>{
31. corAtual = corAtual === cor1 ? cor2 : cor1
32. aplicarCor(corAtual)
33. },1000)
34. }else{
35. aplicarCor(binding.value)
36. }
37. }, atraso);
38. }
39. }
40. },
41. data() {
42. return {
43. cor: 'red'
44. }
45. },
46. }
47. </script>
48. **Aula 192 – Passando Valores mais complexos para as diretivas**
49. Nesta aula iremos passar a cor1, cor2 o atraso e o intervalo da atualização, através de um objeto.
50. export default {
51. directives:{
52. 'destaque-local' : {
53. bind(*el*, *binding*, *vnode*){
54. const aplicarCor = *cor* => {
55. if(*binding*.arg === 'fundo'){
56. *el*.style.backgroundColor = *cor*
57. }else{
58. *el*.style.color = *cor*
59. }
60. }
61. let atraso = 0
62. if(*binding*.modifiers['atrasar']) atraso = *binding*.value.atraso
63. const cor1 = *binding*.value.cor1
64. const cor2 = *binding*.value.cor2
65. let corAtual = cor1
66. setTimeout(() => {
67. if(*binding*.modifiers['alternar']) {
68. setInterval(()=>{
69. corAtual = corAtual === cor1 ? cor2 : cor1
70. aplicarCor(corAtual)
71. },*binding*.value.interval)
72. }else{
73. aplicarCor(*binding*.value.cor1)
74. }
75. }, *binding*.value.atraso);
76. }
77. }
78. },
79. data() {
80. return {
81. cor: 'red'
82. }
83. },
84. }
85. </script>
86. Aula 193 – Hora de praticar