



Introduction to Vue.js

```
<!doctype html>
<html>
  <head>
    <meta charset="utf-8" />
    <title>Getting started with Vue.js</title>

    <!-- development version, includes helpful console warnings -->
    <script src="https://cdn.jsdelivr.net/npm/vue/dist/vue.js"></script>
  </head>
  <body>
    <div id="app">
      {{ message }}
    </div>

    <script>
      var app = new Vue({
        el: '#app',
        data: {
          message: 'Hello Chemnitz!'
        }
      })
    </script>
  </body>
</html>
```

R
E
M
E
M
B
E
R
T
S

```
<!doctype html>
<html>
  <head>
    <meta charset="utf-8" />
    <title>Getting started with Vue.js</title>

    <!-- development version, includes helpful console warnings -->
    <script src="https://cdn.jsdelivr.net/npm/vue/dist/vue.js"></script>
  </head>
  <body>
    <div id="app">
      {{ message }}
    </div>

    <script>
      var app = new Vue({
        el: '#app',
        data: {
          message: 'Hello Chemnitz!'
        }
      })
    </script>
  </body>
</html>
```

R
E
M
E
M
B
E
R
T
S

```
<!doctype html>
<html>
  <head>
    <meta charset="utf-8" />
    <title>Getting started with Vue.js</title>

    <!-- development version, includes helpful console warnings -->
    <script src="https://cdn.jsdelivr.net/npm/vue/dist/vue.js"></script>
  </head>
  <body>
    <div id="app">
      {{ message }}
    </div>

    <script>
      var app = new Vue({
        el: '#app',
        data: {
          message: 'Hello Chemnitz!'
        }
      })
    </script>
  </body>
</html>
```

R
E
N
D
E
R
H
Z
G

```
<!doctype html>
<html>
  <head>
    <meta charset="utf-8" />
    <title>Getting started with Vue.js</title>

    <!-- development version, includes helpful console warnings -->
    <script src="https://cdn.jsdelivr.net/npm/vue/dist/vue.js"></script>
  </head>
  <body>
    <div id="app">
      {{ message }}
    </div>

    <script>
      var app = new Vue({
        el: '#app',
        data: {
          message: 'Hello Chemnitz!'
        }
      })
    </script>
  </body>
</html>
```

R
E
N
D
E
R
H
Z
Q

```
<!doctype html>
<html>
  <head>
    <meta charset="utf-8" />
    <title>Getting started with Vue.js</title>

    <!-- development version, includes helpful console warnings -->
    <script src="https://cdn.jsdelivr.net/npm/vue/dist/vue.js"></script>
  </head>
  <body>
    <div id="app">
      {{ message }}
    </div>

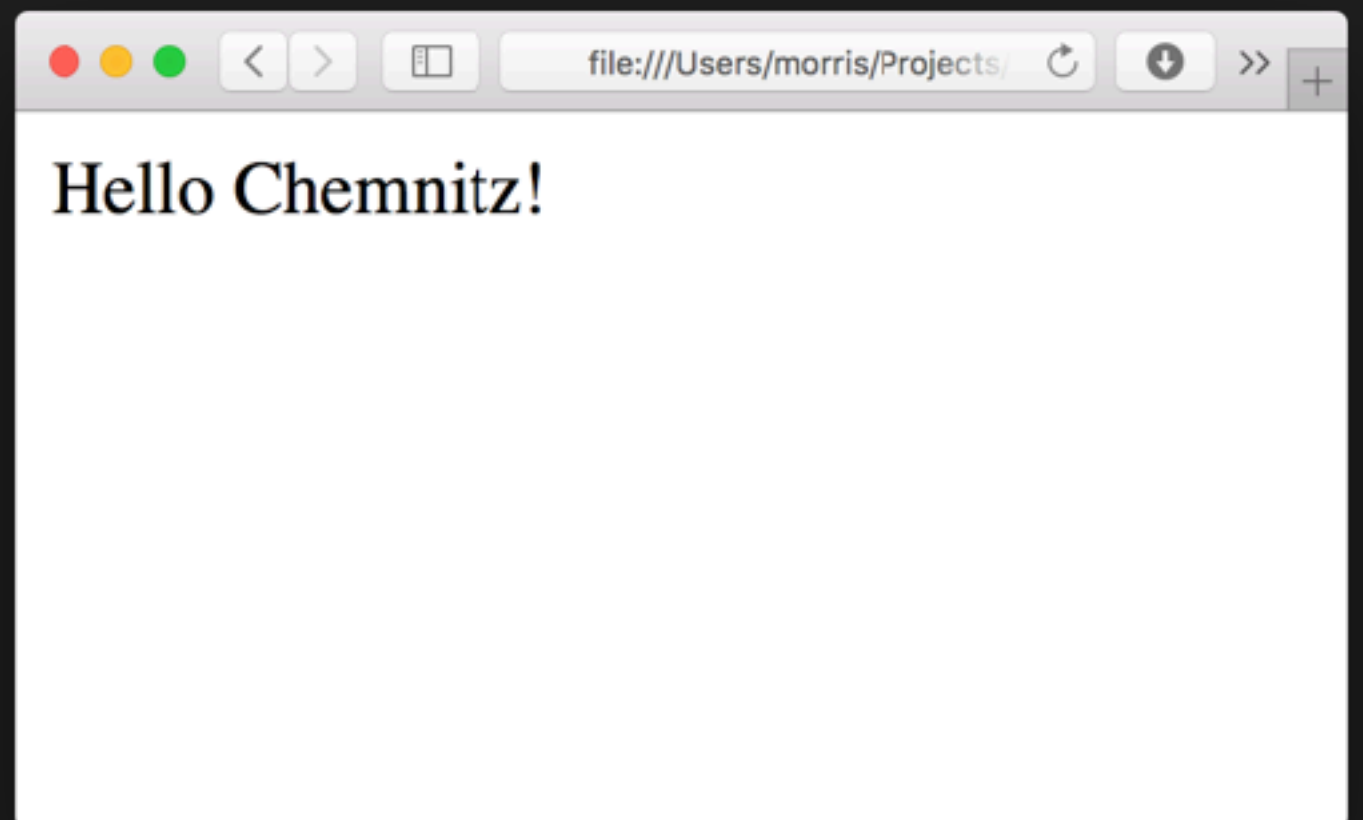
    <script>
      var app = new Vue({
        el: '#app',
        data: {
          message: 'Hello Chemnitz!'
        }
      })
    </script>
  </body>
</html>
```

R
E
M
E
M
B
E
R
T
S

```
<!doctype html>
<html>
  <head>
    <meta charset="utf-8" />
    <title>Getting started with Vue.js</title>

    <!-- development version, includes helpful console warnings -->
    <script src="https://cdn.jsdelivr.net/npm/vue/dist/vue.js"></script>
  </head>
  <body>
    <div id="app">
      {{ message }}
    </div>

    <script>
      var app = new Vue({
        el: '#app',
        data: {
          message: 'Hello Chemnitz!'
        }
      })
    </script>
  </body>
</html>
```



R
E
M
E
M
B
E
R
T
H
N
I
T
Z

```
<div id="app">
  <span v-if="visible">I'm visible</span>
  <ol>
    <li v-for="ship in ships">
      {{ ship.name }}
    </li>
  </ol>
</div>
```

```
<script>
  var app = new Vue({
    el: '#app',
    data: {
      visible: true,
      ships: [
        { name: 'Millennium Falcon' },
        { name: 'Death Star' },
        { name: 'Imperial shuttle' },
      ]
    }
  })
```

```
  // state can be changed
  app.visible = false
  app.ships.push({ name: 'Y-wing' })
</script>
```

#0
L0
OZ
OZ
OZ
PS
T
I
O
N
A
L
S


```
<div id="app">
  <span v-if="visible">I'm visible</span>
  <ol>
    <li v-for="ship in ships">
      {{ ship.name }}
    </li>
  </ol>
</div>
```

```
<script>
  var app = new Vue({
    el: '#app',
    data: {
      visible: true,
      ships: [
        { name: 'Millennium Falcon' },
        { name: 'Death Star' },
        { name: 'Imperial shuttle' },
      ]
    }
  })
```

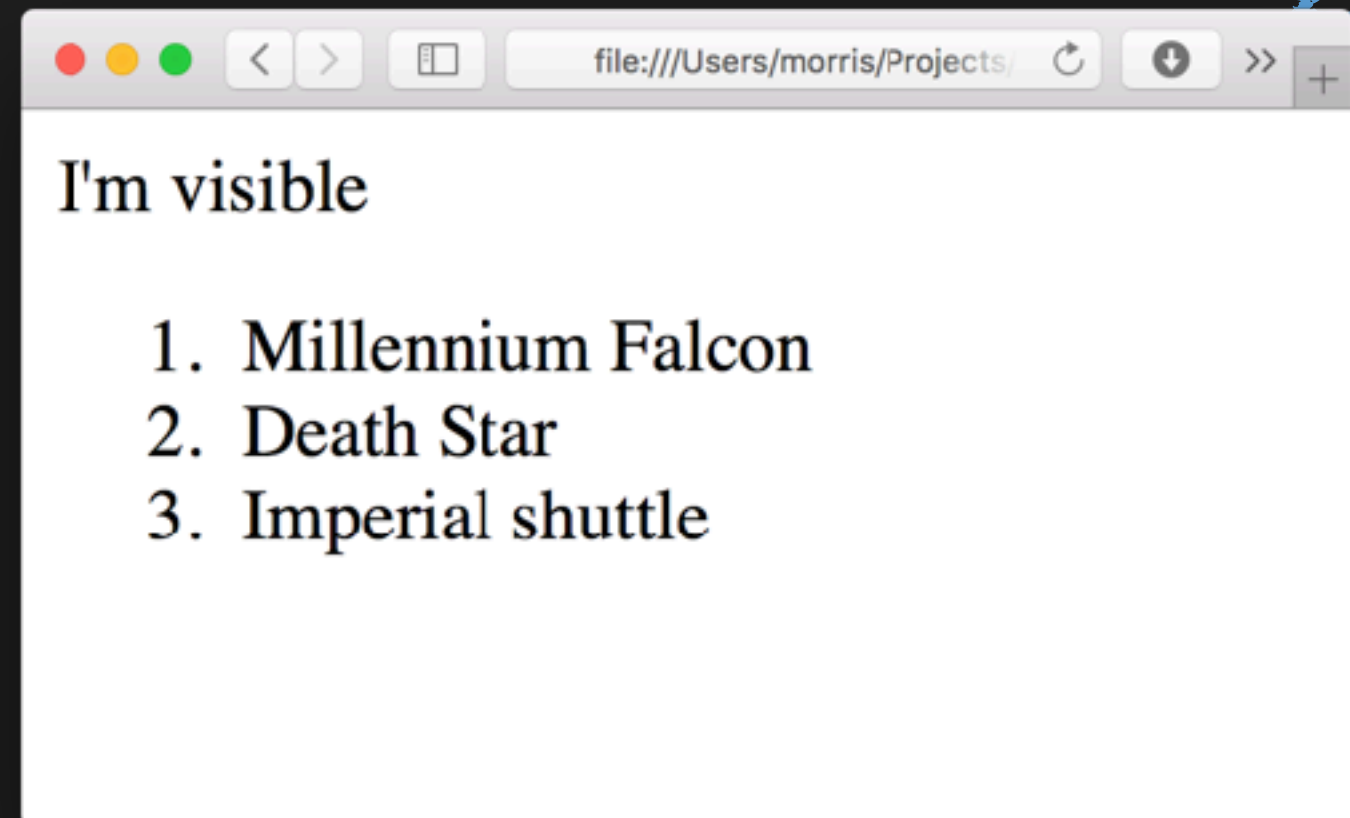
```
  // state can be changed
  app.visible = false
  app.ships.push({ name: 'Y-wing' })
</script>
```

#0
L0
ON
D0
PIT
S
ITIONALS

```
<div id="app">
  <span v-if="visible">I'm visible</span>
  <ol>
    <li v-for="ship in ships">
      {{ ship.name }}
    </li>
  </ol>
</div>

<script>
  var app = new Vue({
    el: '#app',
    data: {
      visible: true,
      ships: [
        { name: 'Millennium Falcon' },
        { name: 'Death Star' },
        { name: 'Imperial shuttle' },
      ]
    }
  })

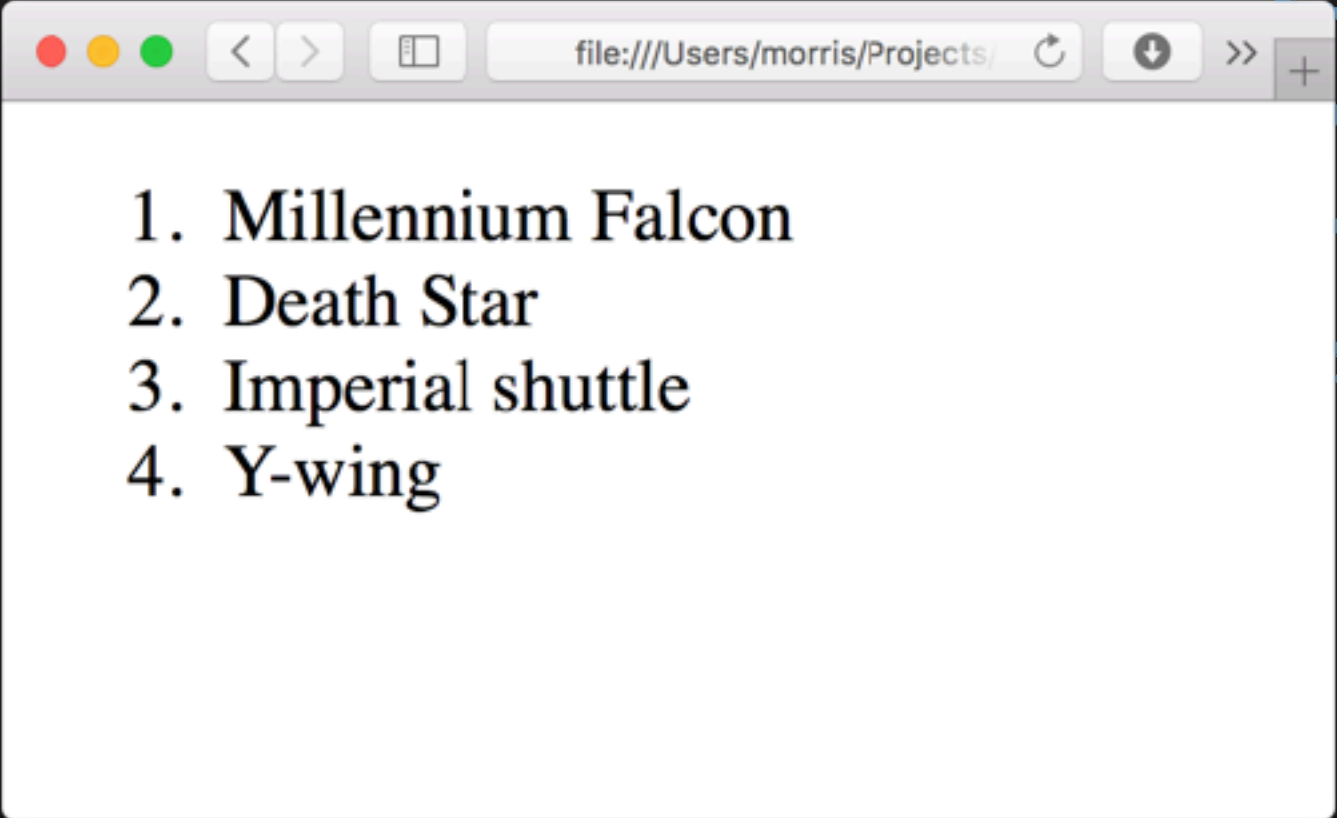
  // state can be changed
  app.visible = false
  app.ships.push({ name: 'Y-wing' })
</script>
```



```
<div id="app">
  <span v-if="visible">I'm visible</span>
  <ol>
    <li v-for="ship in ships">
      {{ ship.name }}
    </li>
  </ol>
</div>

<script>
  var app = new Vue({
    el: '#app',
    data: {
      visible: true,
      ships: [
        { name: 'Millennium Falcon' },
        { name: 'Death Star' },
        { name: 'Imperial shuttle' },
      ]
    }
  })

  // state can be changed
  app.visible = false
  app.ships.push({ name: 'Y-wing' })
</script>
```

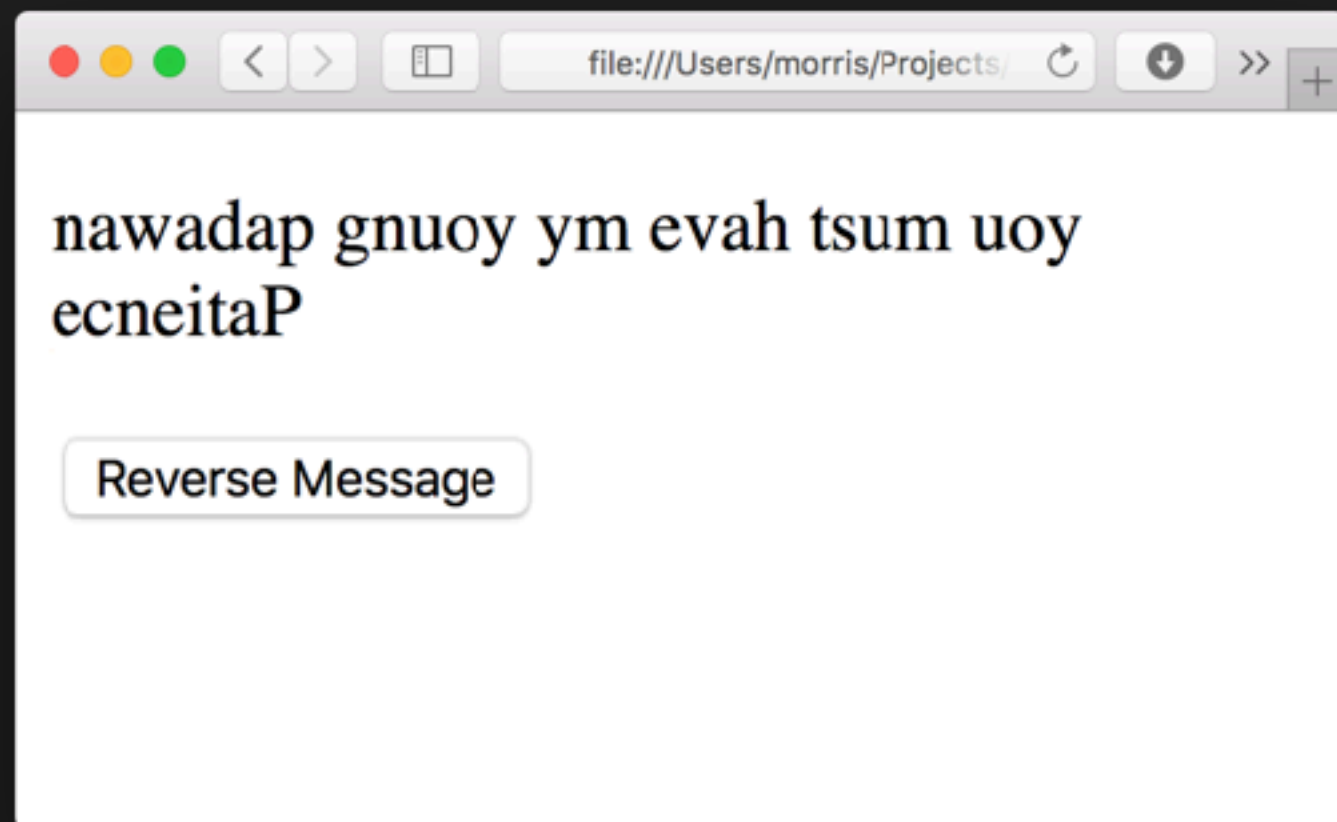
- 
- A screenshot of a web browser window. The address bar shows a file path: file:///Users/morris/Projects/. The browser displays a simple list of four items, each preceded by a number. The list is: 1. Millennium Falcon, 2. Death Star, 3. Imperial shuttle, and 4. Y-wing.
1. Millennium Falcon
 2. Death Star
 3. Imperial shuttle
 4. Y-wing

```
<div id="app">
  <p>{{ message }}</p>
  <button v-on:click="reverseMessage">Reverse Message</button>
</div>

<script>
  var app = new Vue({
    el: '#app',
    data: {
      message: 'Patience you must have my young padawan'
    },
    methods: {
      reverseMessage: function () {
        this.message = this.message.split('').reverse().join('')
      }
    }
  })
</script>
```

```
<div id="app">
  <p>{{ message }}</p>
  <button v-on:click="reverseMessage">Reverse Message</button>
</div>

<script>
  var app = new Vue({
    el: '#app',
    data: {
      message: 'Patience you must have my young padawan'
    },
    methods: {
      reverseMessage: function () {
        this.message = this.message.split('').reverse().join('')
      }
    }
  })
</script>
```



```
<div id="app">
  <p>{{ fullName }}</p>
</div>
```

```
<script>
  var app = new Vue({
    el: '#app',
    data: {
      firstName: 'Han',
      lastName: 'Solo'
    },
    computed: {
      fullName: function () {
        return this.firstName + ' ' + this.lastName
      }
    }
  })
</script>
```

COMPUTED
PROPERTIES

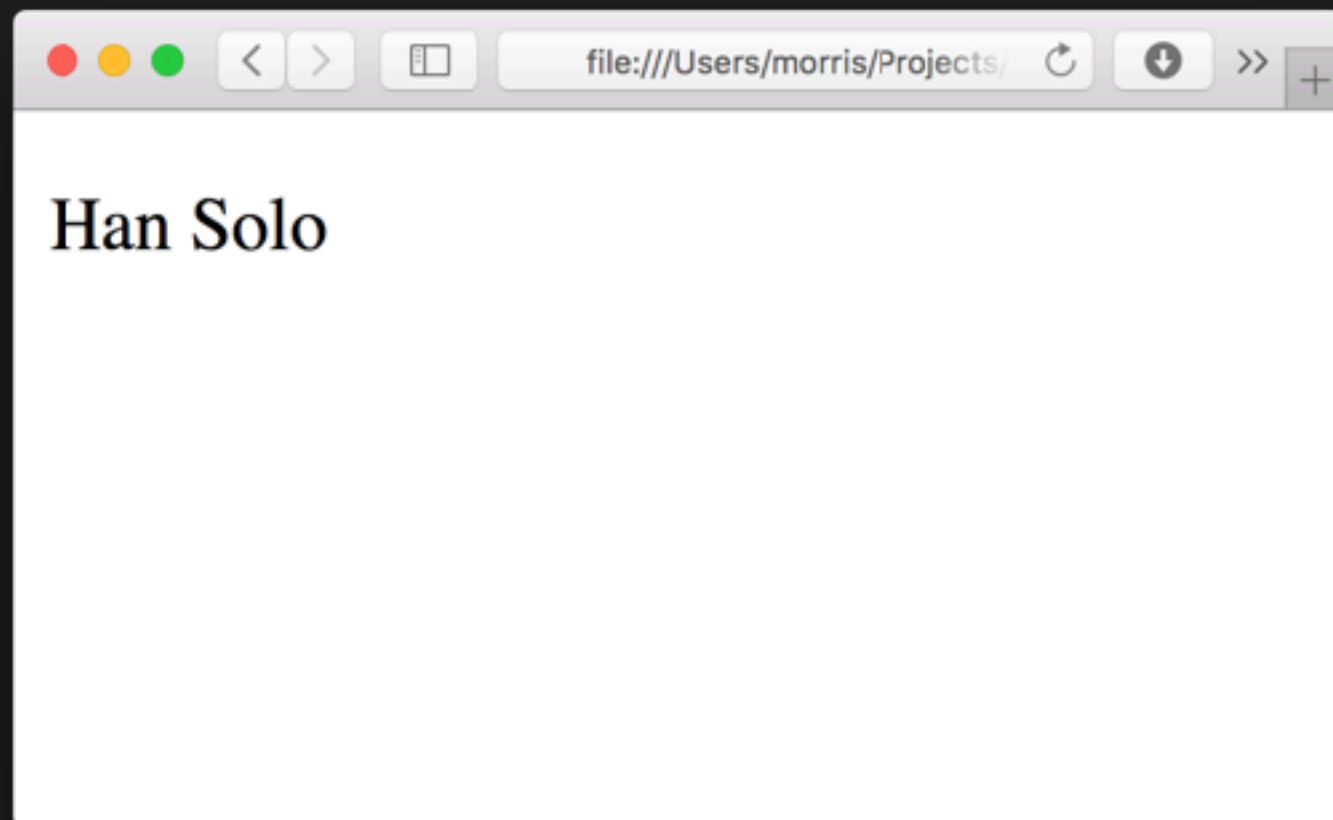
```
<div id="app">
  <p>{{ fullName }}</p>
</div>

<script>
  var app = new Vue({
    el: '#app',
    data: {
      firstName: 'Han',
      lastName: 'Solo'
    },
    computed: {
      fullName: function () {
        return this.firstName + ' ' + this.lastName
      }
    }
  })
</script>
```

COMPUTED
PROPERTIES

```
<div id="app">
  <p>{{ fullName }}</p>
</div>

<script>
  var app = new Vue({
    el: '#app',
    data: {
      firstName: 'Han',
      lastName: 'Solo'
    },
    computed: {
      fullName: function () {
        return this.firstName + ' ' + this.lastName
      }
    }
  })
</script>
```



COMPUTER
PROPERTIES


```
<div id="app">
  <div
    class="person"
    v-for="person in persons"
    v-bind:class="{ hairy: person.hairy }">
    {{ person.name }}
  </div>
</div>

<div id="result">
  <div class="person">Han Solo</div>
  <div class="person hairy">Chewbacca</div>
</div>

<script>
  var app = new Vue({
    el: '#app',
    data: {
      persons: [
        { name: 'Han Solo' },
        { name: 'Chewbacca', hairy: true }
      ]
    }
  })
</script>
```

CLASSES

```
<div id="app">
  <div
    class="person"
    v-for="person in persons"
    v-bind:class="{ hairy: person.hairy }">
    {{ person.name }}
  </div>
</div>

<div id="result">
  <div class="person">Han Solo</div>
  <div class="person hairy">Chewbacca</div>
</div>

<script>
  var app = new Vue({
    el: '#app',
    data: {
      persons: [
        { name: 'Han Solo' },
        { name: 'Chewbacca', hairy: true }
      ]
    }
  })
</script>
```

CLASSES

THEZG

```
<div id="app">
  <div
    class="person"
    v-for="person in persons"
    v-bind:class="{ hairy: person.hairy }">
    {{ person.name }}
  </div>
</div>

<div id="result">
  <div class="person">Han Solo</div>
  <div class="person hairy">Chewbacca</div>
</div>

<script>
  var app = new Vue({
    el: '#app',
    data: {
      persons: [
        { name: 'Han Solo' },
        { name: 'Chewbacca', hairy: true }
      ]
    }
  })
</script>
```

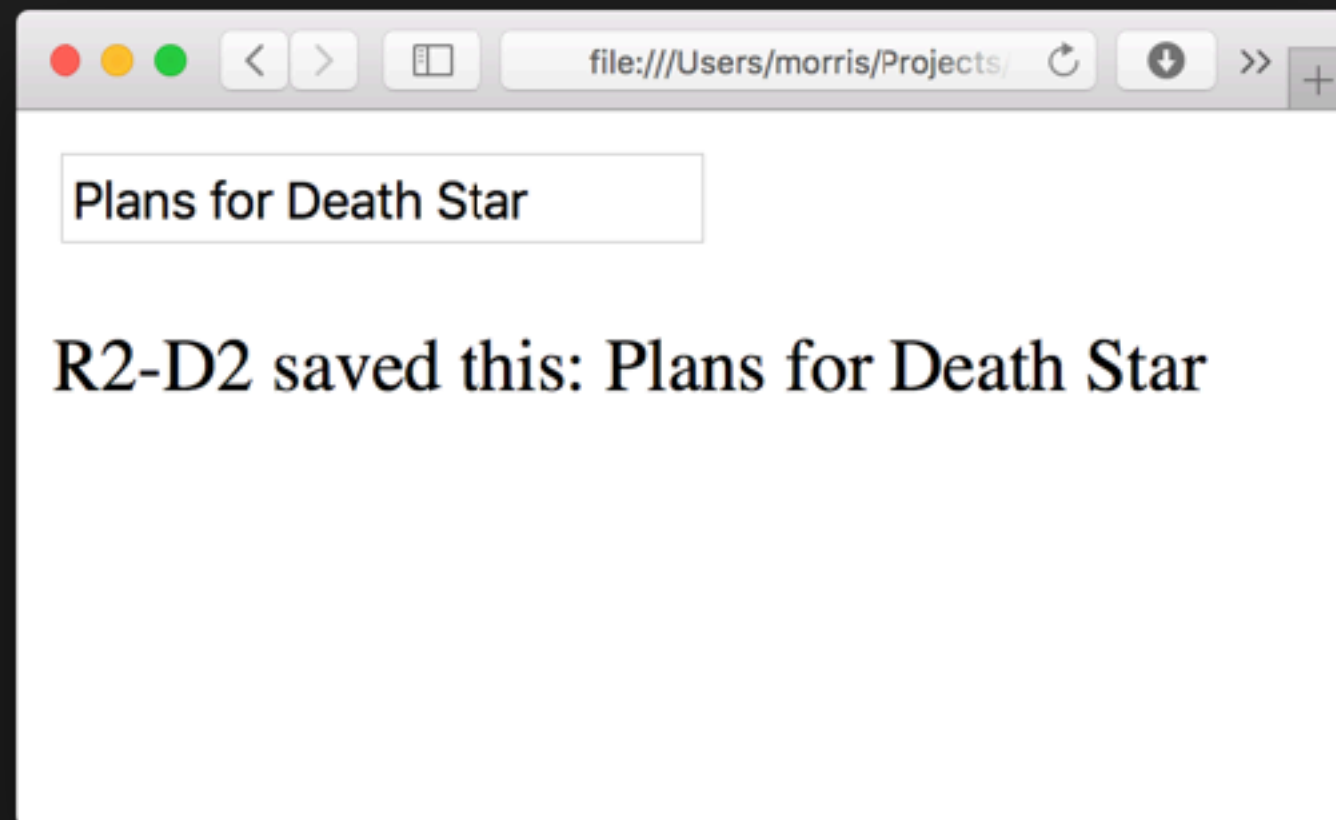
CU
LH
AN
SD
SH
EZ
SG

```
<div id="app">
  <input v-model="message" placeholder="Enter message ...">
  <p>R2-D2 saved this: {{ message }}</p>
</div>

<script>
  var app = new Vue({
    el: '#app',
    data: {
      message: ''
    }
  })
</script>
```

```
<div id="app">
  <input v-model="message" placeholder="Enter message ...">
  <p>R2-D2 saved this: {{ message }}</p>
</div>

<script>
  var app = new Vue({
    el: '#app',
    data: {
      message: ''
    }
  })
</script>
```



```
<div id="app">
  <ol>
    <list-item
      v-for="person in people"
      v-bind:element="person"
      v-bind:key="person.id">
    </list-item>
  </ol>
</div>

<script>
  Vue.component('list-item', {
    props: ['element'],
    template: '<li>{{ element.name }}</li>'
  })

  var app = new Vue({
    el: '#app',
    data: {
      people: [
        { id: 0, name: 'R2-D2' },
        { id: 1, name: 'Luke Skywalker' },
        { id: 2, name: 'Chewbacca' },
        { id: 3, name: 'Wat Tambor' },
        { id: 4, name: 'Shaak Ti' }
      ]
    }
  })
</script>
```

```
<div id="app">
  <ol>
    <list-item
      v-for="person in people"
      v-bind:element="person"
      v-bind:key="person.id">
    </list-item>
  </ol>
</div>

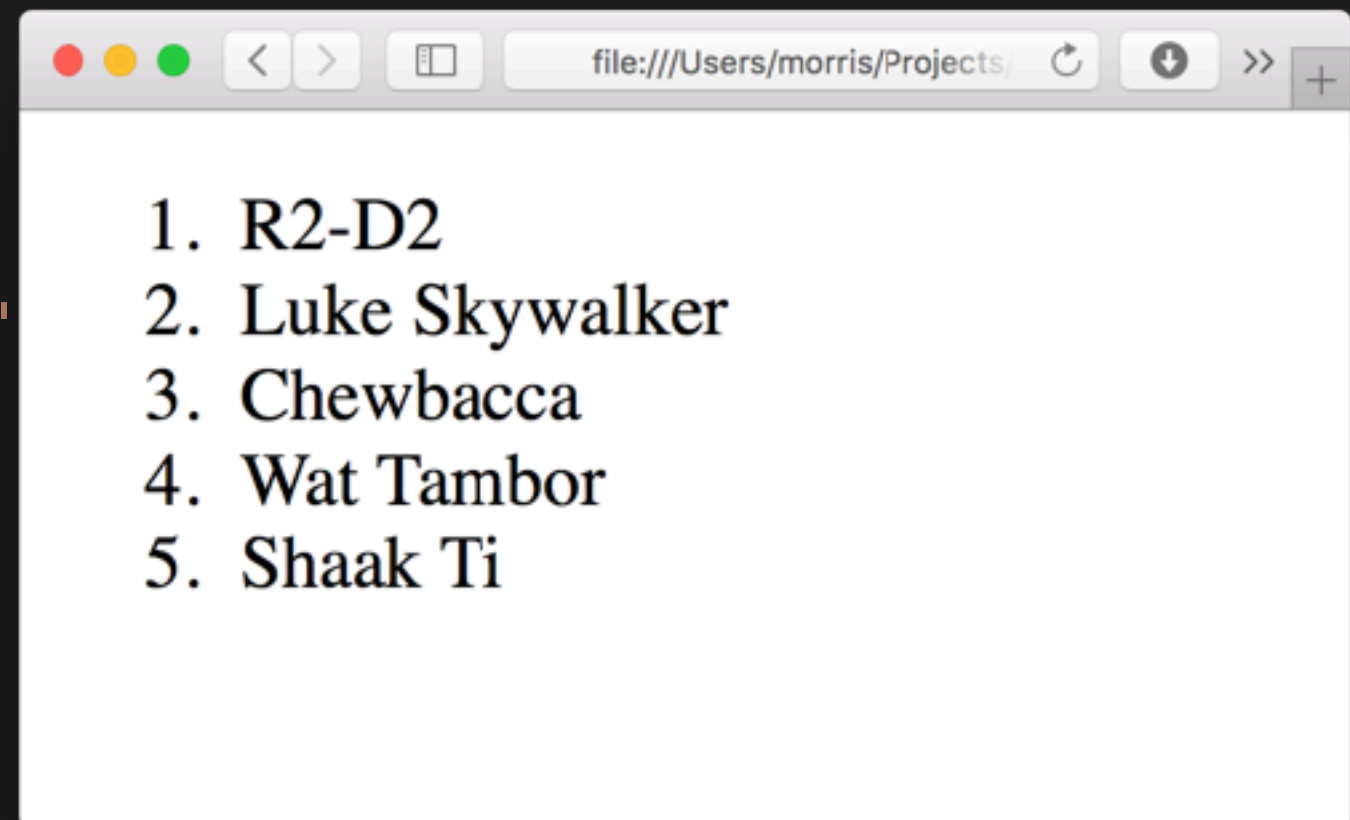
<script>
  Vue.component('list-item', {
    props: ['element'],
    template: '<li>{{ element.name }}</li>'
  })

  var app = new Vue({
    el: '#app',
    data: {
      people: [
        { id: 0, name: 'R2-D2' },
        { id: 1, name: 'Luke Skywalker' },
        { id: 2, name: 'Chewbacca' },
        { id: 3, name: 'Wat Tambor' },
        { id: 4, name: 'Shaak Ti' }
      ]
    }
  })
</script>
```

```
<div id="app">
  <ol>
    <list-item
      v-for="person in people"
      v-bind:element="person"
      v-bind:key="person.id">
    </list-item>
  </ol>
</div>

<script>
  Vue.component('list-item', {
    props: ['element'],
    template: '<li>{{ element.name }}</li>'
  })

  var app = new Vue({
    el: '#app',
    data: {
      people: [
        { id: 0, name: 'R2-D2' },
        { id: 1, name: 'Luke Skywalker' },
        { id: 2, name: 'Chewbacca' },
        { id: 3, name: 'Wat Tambor' },
        { id: 4, name: 'Shaak Ti' }
      ]
    }
  })
</script>
```




```
<template>
  <p>Hello {{ pilot }}!</p>
  <cockpit-component />
</template>

<script>
import CockpitComponent from './CockpitComponent.vue'

module.exports = {
  name: 'Millenium Falcon',
  data: function() {
    return {
      pilot: 'Chewbacca'
    }
  },
  components: {
    CockpitComponent
  }
}
</script>

<style scoped>
p {
  font-size: 32px;
  text-align: center;
}
</style>
```

SINGLE FILE
COMPONENTS

```
<template>
  <p>Hello {{ pilot }}!</p>
  <cockpit-component />
</template>

<script>
import CockpitComponent from './CockpitComponent.vue'

module.exports = {
  name: 'Millenium Falcon',
  data: function() {
    return {
      pilot: 'Chewbacca'
    }
  },
  components: {
    CockpitComponent
  }
}
</script>

<style scoped>
p {
  font-size: 32px;
  text-align: center;
}
</style>
```

SINGLE FILE
COMPONENTS

```
<template>
  <p>Hello {{ pilot }}!</p>
  <cockpit-component />
</template>

<script>
import CockpitComponent from './CockpitComponent.vue'

module.exports = {
  name: 'Millenium Falcon',
  data: function() {
    return {
      pilot: 'Chewbacca'
    }
  },
  components: {
    CockpitComponent
  }
}
</script>

<style scoped>
p {
  font-size: 32px;
  text-align: center;
}
</style>
```

SINGLE THE
COMPONENTS

```
<template>
  <p>Hello {{ pilot }}!</p>
  <cockpit-component />
</template>

<script>
import CockpitComponent from './CockpitComponent.vue'

module.exports = {
  name: 'Millenium Falcon',
  data: function() {
    return {
      pilot: 'Chewbacca'
    }
  },
  components: {
    CockpitComponent
  }
}
</script>

<style scoped>
p {
  font-size: 32px;
  text-align: center;
}
</style>
```

SINGLE THE
COMPONENTS

```
<template>
  <p>Hello {{ pilot }}!</p>
  <cockpit-component />
</template>

<script>
import CockpitComponent from './CockpitComponent.vue'

module.exports = {
  name: 'Millenium Falcon',
  data: function() {
    return {
      pilot: 'Chewbacca'
    }
  },
  components: {
    CockpitComponent
  }
}
</script>

<style scoped>
p {
  font-size: 32px;
  text-align: center;
}
</style>
```

SINGLE FILE
COMPONENTS

```
<template>
  <p>Hello {{ pilot }}!</p>
  <cockpit-component />
</template>

<script>
import CockpitComponent from './CockpitComponent.vue'

module.exports = {
  name: 'Millenium Falcon',
  data: function() {
    return {
      pilot: 'Chewbacca'
    }
  },
  components: {
    CockpitComponent
  }
}
</script>

<style scoped>
p {
  font-size: 32px;
  text-align: center;
}
</style>
```

SINGLE THE
COMPONENTS



- ▶ routing <https://router.vuejs.org>
- ▶ state management <https://vuex.vuejs.org>
- ▶ brilliant docs about testing, server side rendering, building
- ▶ ...



- ▶ view only
- ▶ single file components
- ▶ integrable into existing enviroment

Thanks

@MorrisJobke