

Vue.js の基本構文(v-bind,v-on,v-model)

ディレクティブとは

- HTML 要素に Vue.js の機能を追加するしくみ
- HTML テンプレート内で要素に直接指定することができる特別な属性
- その特別な属性は **v-** から始まる
- ディレクティブを使用することで、DOM 要素の表示や振る舞いを変更することができる

テンプレート構文

- HTML テンプレート内で Vue.js のデータバインディング、ディレクティブを記述するための独自記法
- テンプレート構文を使用することで JavaScript や TypeScript を記述せずに簡単にデータバインディングや DOM 操作を行うことができる。
- [テンプレート構文](#)

v-bind

- リアクティブに HTML 属性を更新する
- v-bind は引数をとるディレクティブ
- 引数をとる場合はディレクティブ名の後にコロンで表記する

```
<a v-bind:href="url"></a>
```

- 省略記法は以下

```
<a :href="url"></a>
```

- 引数は動的な引数をとることができる

```
<a v-bind:[attributeName]="url"></a>
```

Vue 側に `url` と `attributeName` という data プロパティを持っている

参考 : [v-bind](#)

v-on

- 要素にイベントの指定を行う
- イベント種別は引数で示される
- 式はメソッド名またはインラインステートメント(直接ロジックを埋め込む)のいずれかを指定できる

```
<a v-on:click="increment"></a>
```

```
<a v-on:click="num++"></a>
```

- 引数を動的に表現する

```
<a v-on:[event]="increment"></a>
```

- 省略記法は以下

```
<a @click="increment"></a> <a @click="num++"></a>
```

```
<a @click="num++"></a>
```

- 奥の要素へのイベントの伝播(イベントバブリング)を防ぐ`stopPropagation`
 - イベント修飾子`.stop`イベント修飾子をつけることで同じ機能となる

```
<p @mousemove="changeMouseposition($event, 100)">  
  この部分にマウスをのせるとマウスの座標軸を100倍した値を取得します。<span  
    @mousemove.stop  
    >この部分はマウスをのせても何も反応しません</span>  
>  
</p>
```

- a タグをクリックしても遷移しない(デフォルトの挙動を妨げる)`preventDefault`
 - イベント修飾子`.prevent`を使う

```
<a :href="url" @click.prevent>遷移させたくない</a>  
<br />
```

```
<button @click.prevent="onSubmit">保存</button>
```

参考 : [v-on](#)

v-model

- formのinput要素またはコンポーネント上に双方向バインディングを作成する
- テキストボックスの場合input要素でtype属性textに対してv-modelをつける

```
<input id="name" type="text" v-model="userInfo.name" />
```

- チェックボックス 1 つの場合input要素でtype属性checkboxに対してv-modelをつける

```
<label><input type="checkbox" v-model="userInfo.checked" />同意する</label>
```

- チェックボックス複数の場合input要素でtype属性checkboxに対して**全て同じ値**のv-modelをつける

```
<label><input type="checkbox" v-model="userInfo.request" />入会希望</label>
<label><input type="checkbox" v-model="userInfo.request" />メルマガ希望</label>
<label><input type="checkbox" v-model="userInfo.request" />どちらも希望しない
</label>
```

- セレクトボックス複数の場合select要素に対してv-modelをつける

```
<select id="prefectures" v-model="userInfo.prefecture">
  <option value="" disabled>選択してください</option>
  <option>東京</option>
  <option>名古屋</option>
  <option>大阪</option>
</select>
```

- ラジオボタンの場合input要素でtype属性radioに対して**全て同じ値**のv-modelをつける

```
<label><input type="radio" v-model="userInfo.gender" value="男" />男性
</label>
<label><input type="radio" v-model="userInfo.gender" value="女" />女性
</label>
```

- テキストエリアの場合textarea要素に対してv-modelをつける

```
<textarea id="message" cols="30" rows="10" v-model="userInfo.message">
</textarea>
```

参考：[v-model](#)

ワーク 1

上記の内容について、作成したsample-appのDirective.vueに追記して完成させましょう。

配布のコードはuserInfoの型をきちんと指定していないため一部上手く動作しないところがあります。初期値を設定しましょう。

ワーク 2

sample-appに新しくForm.vueを作成し、以下のような内容になるようにプログラムを書きましょう

補足

v-接頭辞は、テンプレート内の Vue 独自の属性を識別するための目印となる。頻繁に利用されるディレクティブに対しては冗長に感じることもある。シングルページアプリケーションを作成するにあたり、全てのテンプレートを Vue で管理している場合、v-接頭辞をつける必要性は低いので、Vue は 2 つの最もよく使われるディレクティブv-bindとv-onに対して特別な省略記法を提供している。