computed(算出プロパティ)、watch(監視プロパティ)

サンプルコード：ComputedWatch.vue

computed(算出プロパティ) とは

- 算出プロパティとも呼ばれる
- テンプレート内に式を書けることは便利だがロジックが複雑になるとコードが煩雑になる
- リアクティブなデータをそのまま描画するのではなく、文字列を加工したり、数値計算したりなどロジックを挟むときに使用する
- 関数の戻り値をデータのように扱える

> 参考：comoputed

```
<label for="lastName">氏</label>
<input type="text" if="lastName" v-model="user.lastName" />
<label for="firstName">名</label>
<input type="text" if="firstName" v-model="user.firstName" />
<p>{{ bindName }}</p>
```

```
get bindName(): string {
    if (this.user.firstName && this.user.lastName) {
      return this.user.lastName + this.user.firstName;
    }
    return "お名前を入力してください";
  }
```

computed と method の違い

同じことを method を使っても書くことができる

```
<label for="lastName">氏</label>
<input type="text" if="lastName" v-model="user.lastName" />
<label for="firstName">名</label>
<input type="text" if="firstName" v-model="user.firstName" />
<p>{{ bindNameMethod() }}</p>
```

```
  bindNameMethod(): string {
    if (this.user.firstName && this.user.lastName) {
      return this.user.lastName + this.user.firstName;
    }
    return "お名前を入力してください";
  }
```

ログを出して違いを確認する

- method は他のリアクティブな値に変更が加わるたびに発動する
- computed は対象のリアクティブな値が変更されたときのみ発動する

```
<h2>watchとmethod</h2>
<label for="lastName">氏</label>
<input type="text" if="lastName" v-model="user.lastName" />
<label for="firstName">名</label>
<input type="text" if="firstName" v-model="user.firstName" />
<p>computedを使用：{{ bindName }}</p>
<p>methodを使用：{{ bindNameMethod() }}</p>

<button @click="number++">+</button>
<p>カウントアップ{{ number }}</p>
```

```
//computed
  get bindName(): string {
    console.log("computed発動");
    if (this.user.firstName && this.user.lastName) {
      return this.user.lastName + this.user.firstName;
    }
    return "お名前を入力してください";
  }
  //method
  bindNameMethod(): string {
    console.log("method発動");
    if (this.user.firstName && this.user.lastName) {
      return this.user.lastName + this.user.firstName;
    }
    return "お名前を入力してください";
  }
```

> computed はリアクティブな依存関係に基づきキャッシュされる

watch(監視プロパティ)とは

- Vue インスタンス上の**特定のデータの変更**を監視する
- データの変更に応じて必要な処理を行う場合に使用する
- `computed`か`watch`かで迷った場合は`computed`を使うのがよい場合が多い

> 参考： watch

computed と method で書いたコードを watch に置き換える

```
<h2>computedとmethodとwatch</h2>
<label for="lastName">氏</label>
<input type="text" if="lastName" v-model="user.lastName" />
<label for="firstName">名</label>
<input type="text" if="firstName" v-model="user.firstName" />
<p>computedを使用：{{ bindName }}</p>
<p>methodを使用：{{ bindNameMethod() }}</p>
<p>watchを使用：{{ userName }}<span v-if="!userName">お名前を入力してください</span></p>
```

```
  @Watch("user.firstName")
  bindNameWatch(newValue: string, oldValue: string): void {
    console.log(`watch:${newValue},${oldValue}`);
    this.userName = this.user.lastName + newValue;
  }
  @Watch("user.lastName")
  bindNameWatchL(newValue: string): void {
    this.userName = newValue + this.user.firstName;
  }
```

> とても冗長なコードになる

watch を使ってエラー処理をする

```
<h2>watchでエラー処理</h2>
<label for="title">タイトル入力</label>
<input type="title" id="title" v-model="title" />
<p>{{ title }}</p>
<p v-if="error" :style="red">{{ error }}</p>
```

```
  @Watch("title")
  titleWatch(newValue: string): void {
    console.log(newValue);
    this.error = newValue.length > 10 ? "10文字以内でお願いします！" : "";
  }
```