

再利用対象となるコンポーネントの作成

main.ts の import を理解する

`import App from "./App.vue"`とは、コンポーネントのオブジェクトを `import` していることと同義。

- `App.vue` は`template`と`script`と`style`から成り立っている単一ファイルコンポーネント
- コンポーネントの一種
- コンポーネントはオブジェクト
- 3つのエリアに分かれている`vue`ファイルをインポートすることで1つのオブジェクトになる
- そもそも `vue` ファイルは一般的には使えないがなぜ使えるか？
 - `Vue-cli` の中で `webpack`(バンドルツール)動いており、その中の `vue-loader` がオブジェクトにコンパイルしている

`vue`ファイルは`import`をすると最終的に**コンポーネントのオブジェクト**になる

ワーク 1

実際にコンソールログを確認し、`App`の中身を確認しよう

```
import Vue from "vue";
import App from "./App.vue";
import ComputedWatch from "./components/ComputedWatch.vue";
Vue.config.productionTip = false;
console.log(App);
new Vue({
  render: (h) => h(App),
}).$mount("#app");
```

App.vue の中身を確認する

- 他のVueファイルをインポートしている

```
import ComputedWatch from "./components/ComputedWatch.vue";
import OriginalFilter from "./components/OriginalFilter.vue";
```

import すると vue ファイルはオブジェクトになる

- コンポーネントとして登録している

```
@Component({
  components: {
    Directive,
    ComputedWatch,
    OriginalFilter,
  },
})
```

以下の書き方の省略系(つまりオブジェクトとして扱っている) オブジェクトにおいてキーと値が同じであれば省略できる

```
@Component({
  components: {
    Directive: Directive,
    ComputedWatch: ComputedWatch,
    OriginalFilter: OriginalFilter,
  },
})
```

- 登録した DOM テンプレートを使用する

```
<h2>算出プロパティとウォッチャ</h2>
<ComputedWatch></ComputedWatch>
<h2>フィルター</h2>
<OriginalFilter></OriginalFilter>
```

再利用対象となるコンポーネント

- vue ファイルで単一ファイルコンポーネントを作成
- 別の vue ファイルに import する
- import した vue ファイルをコンポーネントとして登録

上記の手順を踏むことで、再利用対象となるコンポーネントを作成し、使用することができる

コンポーネントは基本的に `components` フォルダにまとめることが一般的