



## component

### Single-file components

Only one component per file

### Consistent component filename casing

Always use only PascalCase or only kebab-case

### Single-file component top level element order

<template><script><style> or <script><template><style>

### Implicit parent-child communication

Don't mutate props or reference parent data directly

## NAMING

### Base component names

Prefix app specific components w/ Base, V or App

### Full word component names

Avoid abbreviations

### Multi-word component names

Always use multi-word names unless root level app

### Order of words in component names

Name components from most general to most descriptive word

### Single-Instance component names (once per page)

Start with 'The' and shouldn't accept props

### Tightly coupled component names

Prefix child component name with parent component name

### Component name casing in JS/JSX

Always use PascalCase in JS or JSX

### Component name casing in templates

Always use PascalCase in components and kebab-case in the DOM

## <template/>

### Directive shorthands

Always use ':' for v-bind and '@' for v-on

### Element attribute order

Maintain consistent order- see [options order]

Always use **key** with **v-for**

**v-if** / **v-else-if** / **v-else** without **key**

Use with caution if same element type

### Multi-attribute elements

One attribute per line for readability

### Quoted attribute values

Always use quotes with non-HTML attributes

### Self-closing components

Always use in templates/JSX but never in DOM!

### Simple expressions in templates

Move complex logic to model

## <style/>

### Avoid element selectors with scoped

Use classes instead to improve performance

### Component style scoping

Only base app component and layout components should have global styles

## <script/>

Component data must be function in single-file component

### Component/instance options order

Maintain consistent order- see Attributes Order

Empty line between component/instance options

Helps with readability

### Simple computed properties

Split complex computed properties into several simple properties

### Detailed prop definitions

Available properties: type, required, default, validator

### Prop name casing

Use camelCase for declaration and kebab-case for templates/JSX

## other

### Private property names

For mix-ins/plugins, prefix property names with \$ \_

### Non-flux state management

Use Vuex over **this.\$root** or global event bus.

## Options Order

el  
name  
parent  
functional  
delimiters  
comments  
components  
directives  
filters  
extends  
mixing  
inheritAttrs  
model  
props / propsData  
data  
computed  
watch  
methods  
template / render  
renderError

## Attribute Order

is  
v-for  
v-if  
v-else-if  
v-else  
v-show  
v-cloak  
v-pre  
v-once  
id  
ref  
key  
slot  
v-model  
v-on @  
v-html  
v-text