TECHNIQUE

# NAMING CONVENTIONS

## WHAT IS NAMING CONVENTION ?

- Rules for naming variables, functions, classes or other entities in source code.

## PURPOSE

- To reduce the effort needed to read and understand source code.
- To enable code reviews to focus on more important issues.

NAMING CONVENTION

## COMMON RULES

## CAMEL CASE

- Start with lowercase.
- Multiple word start with uppercase.
- Ex. productList, renderFooter, getUserById.
- Common used for variables, functions.

## PASCAL CASE

- Same as camelCase but first character start with uppercase.
- Ex. ProductList, RenderFooter, GetUserById.
- Common used for classname.

## SNAKE CASE

- Use only lowercase.
- Separate word with underscore.
- Ex. resume_code, user_id.
- Common used for variables.

## KEBAB CASE

- Use only lowercase.
- Separate word with hyphen.
- Ex. login-button, btn-danger.
- Common used for id, css name in HTML.
- * Cannot use in javascript.

## ALL UPPER CASE

- Use only uppercase.
- Separate word with underscore.
- Ex. API_ENDPOINT, REQUEST_TIMEOUT.
- Common used for application global variable, constant.

NAMING CONVENTION

## RESERVED WORDS

## RESERVED WORDS

- a word that cannot be used as identifier.
- In Javascript such as const, let, class, export, etc…

NAMING CONVENTION

## FOLDERS & FILES

## FOLDER

- Common used in lowercase.
- Multiple word use mostly use snake_case or camelCase.
- But mostly named in one word.

## FILE

- Common used in lowercase.
- If has more than two word use snake_case or camelCase.
- * In Javascript sometime use exported entity convention such as export class so use class convention.

NAMING CONVENTION

# CLASS NAME

## CLASS NAME

- Common used in PascalCase.
- Should be a noun.
- Ex. ProductList, About.

NAMING CONVENTION

## FUNCTION NAME

## FUNCTION NAME

- Common used in camelCase.
- Should be a verb.
- Ex. renderDatePicker, createUser, getActiveMenu.
- * In React functional component use convention same as class name convention.

## CLASS & FUNCTION COMBINATION

- Class should be a noun and function should be a verb makes more readability.
- Ex. product.remove(), image.resizeTo().

NAMING CONVENTION

## VARIABLES

## COMMON VARIABLES

- Common use in camelCase and snake_case.
- Should be a noun.

## CONSTANT VARIABLES

- Common used in all uppercase with underscore.
- Ex. REQUEST_TIMEOUT, API_ENDPOINT.

NAMING CONVENTION

## BOOLEAN

## BOOLEAN

- true/false.
- Start with is, has, can.
- Ex. isActive, hasPhoneNumber.

NAMING CONVENTION

## CSS CONVENTION

## COMMON CSS CONVENTION

- All lowercase with hyphen (-).
- Ex. product-item, icon-github.

## CSS STYLE GUIDE

- BEM - getbem.com
- RSCSS - rscss.io

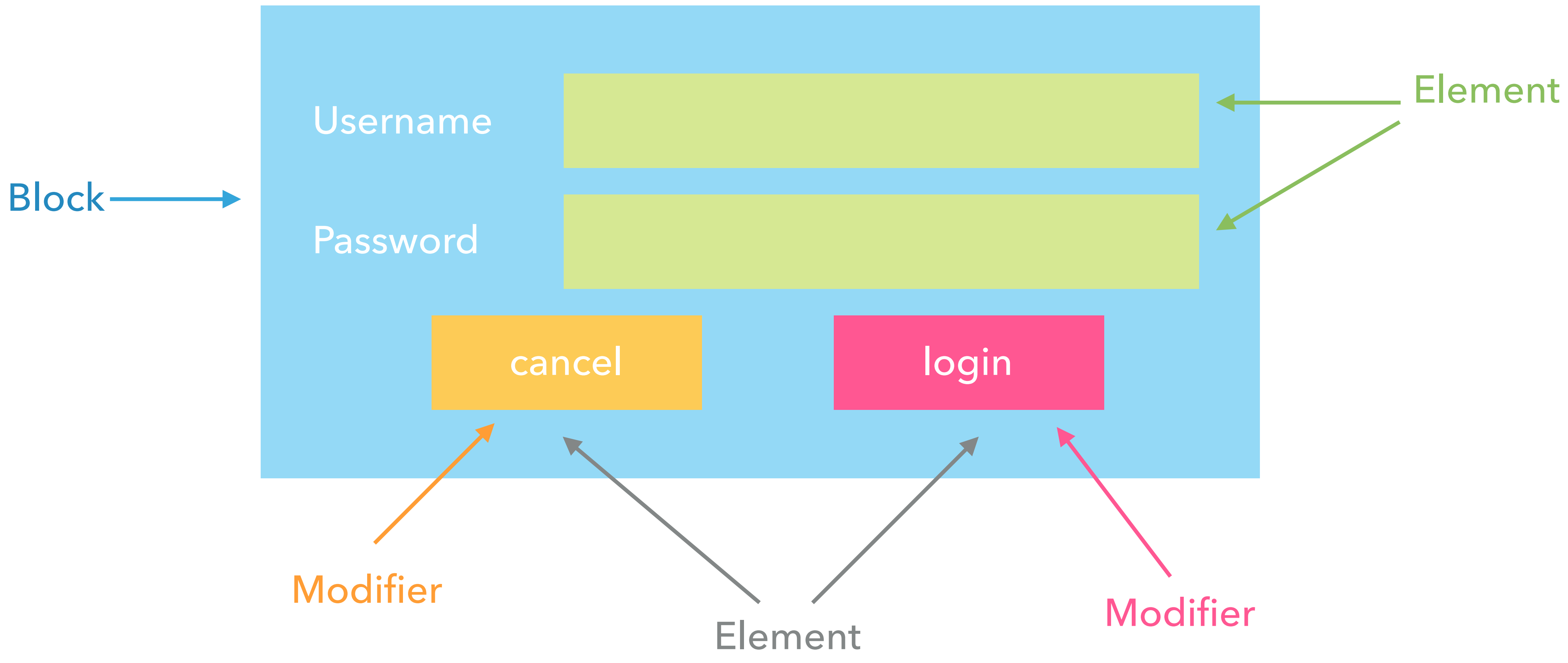
## BEM

- **Block** - Standalone entity that is meaningful on its own.
- **Element** - Parts of a block and have no standalone meaning.
- **Modifier** - Use them to change appearance, behavior or state.

## BEM

- **Block** - Contain letter and can have dash.
- **Element** - Begin with two underscore.
- **Modifier** - Begin with two hyphen.

## BEM


Username
Password
cancel
login
Block
Element
Modifier
Element
Modifier

## BEM

**Concept**

```
.block {
  ...
}

.block__element {
  ...
}

.block__element--modifier {
  ...
}
```

**Example**

```
.loginbox {
  ...
}

.loginbox__button {
  ...
}

.loginbox__button--red {
  ...
}
```

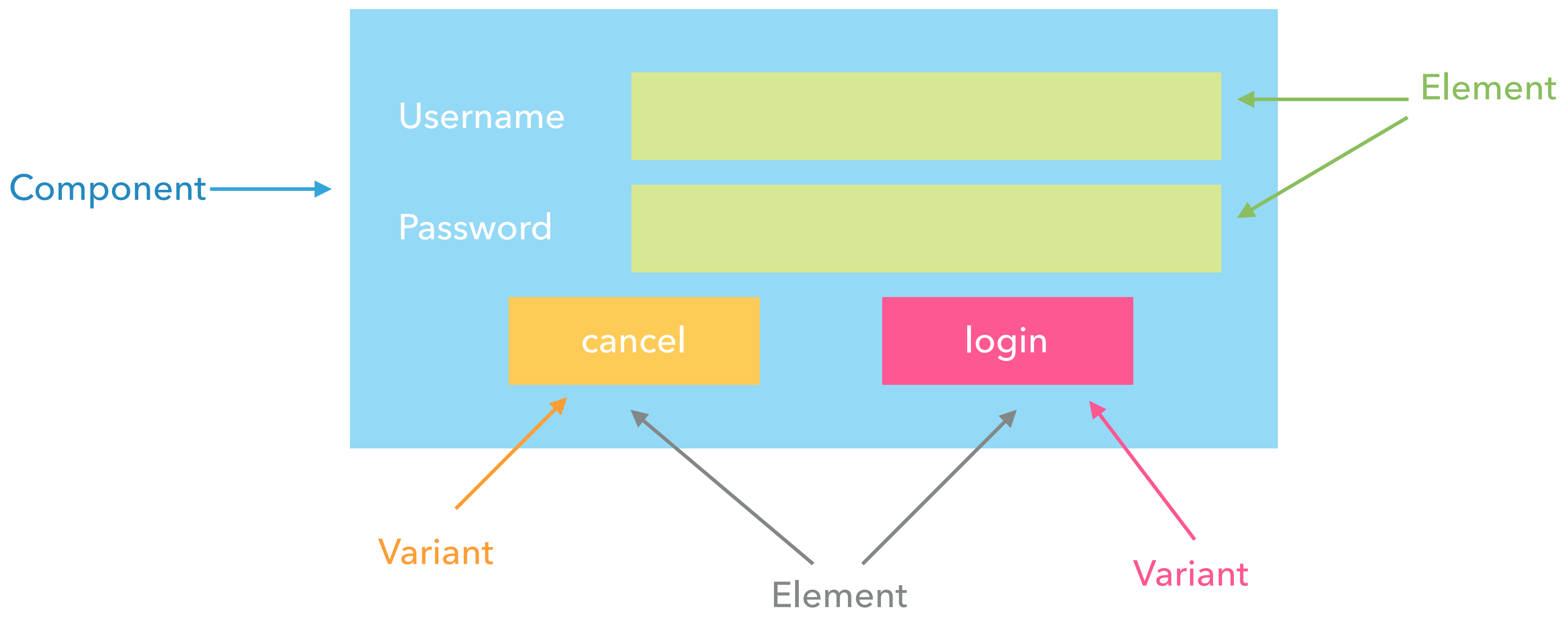
## RSCSS

- **Component** - Standalone entity that is meaningful on its own.
- **Element** - Parts of a block and have no standalone meaning.
- **Variant** - Use them to change appearance, behavior or state.
- **Helper** - Override all style.

## RSCSS

- **Component** - Named at least two words with hyphen each word.
- **Element** - Only one word.
- **Variant** - Only one word and begin with underscore.
- **Helper** - Only one word and begin with hyphen.

## RSCSS


Username
Password
cancel
login
Component
Element
Variant
Element
Variant

## RSCSS

**Concept**

```
.component-box {
  ...
}

.component-box element {
  ...
}

.component-box element -modifier {
  ...
}

.component-box _helper {
  ...
}
```

**Example**

```
.login-box {
  ...
}

.login-box button {
  ...
}

.login-box button -red {
  ...
}

.login-box _nomargin {
  ...
}
```

NAMING CONVENTION

## NAMING TECHNIQUE

## CREATE CLEAR NAMING

- Specify what variable is or function do.
- Ex. checkApply() should be canApply().

## COMMON LETTER VARIABLES

- i, j, k is valid for index counter.
- e is valid for Javascript event Ex. e.target.value.
- e is valid for JAVA exception Ex. catch(exception e).

## DUPLICATE NAME IN FOLDER DIRECTORY

- actions/productAction.js should be actions/product.js
- containers/ProductContainer.js should be containers/Product.js

## DUPLICATE NAME IN OBJECT

- company.companyName should be company.name

## SINGLE AND PLURAL

- Ex. product should contain one product.
- Ex. products should contain array of product.
- Ex. getProducts() should not return an object.

## NAMING WITHOUT ABBREVIATION

- Ex. prod should be product because it can be product, production, produce.

## LANGUAGE SPECIFIC CONVENTION

- Use language specific convention to define identity.
- Ex. Javascript use camelCase.
- Ex. C++ use snake_case.

## GAINS MORE WORD

- Code review.
- Read other code or blog.