

CSS modules

[Знакомство с Sass модулями](#)

USE vs IMPORT

Новый @use похож на @import, но имеет некоторые заметные отличия:

- Файл импортируется единожды, неважно сколько раз вы используете @use в проекте.
- Переменные, миксины и функции (которые в Sass называются «членами»), начинающиеся с подчеркивания (_) или дефиса (-), считаются приватными и не импортируются.
- Члены из подключенного через @use файла (в наше случае buttons.scss) доступны только локально и не передаются последующему импорту.
- Аналогично, @extends будет применяться только вверх по цепочке; то есть расширение применяется только к стилям, которые импортируются, а не к стилям, которые импортируют.
- Все импортированные члены по умолчанию имеют свое пространство имен.

```
@use 'buttons'; /* создает пространство имен `buttons` */
@use 'forms'; /* создает пространство имен `forms` */
/* переменные: <namespace>.$variable */
$btn-color: buttons.$color;
$form-border: forms.$input-border;
//Мы можем изменить или удалить пространство
// имен по умолчанию, добавив к импорту as <name>.
@use 'buttons' as *; /* звездочка удаляет любое пространство имен - НЕ
РЕКОМЕНДУЕТСЯ ТАК ДЕЛАТЬ */
@use 'forms' as 'f';

$btn-color: $color; /* buttons.$color без пространства имен */
$form-border: f.$input-border; /* forms.$input-border пользовательским пространством
имен */
```

Андерскор, дефис в имени файлов

Если имя файла начинается со знака нижнего подчеркивания, то этот файл игнорируется компилятором. Но содержимое этих файлов может и, как правило, импортируют в обычные sass файлы, также в такие файлы включают миксины, функции и они вообще могут не содержать объявления стилей и создаются исключительно для их использования в других файлах. То есть, по сути это вспомогательные файлы, что-то вроде модулей, для использования другими обычными sass файлами.

Приватные сущности

Приватной переменной, начинается с _ или - Такие переменные могут использоваться только внутри файлов, где они определены

```
$_private: true !default; /* не доступна для конфигурации в силу приватности */
```

Импорт множества модулей с помощью forward

Нам не всегда нужно использовать файл и обращаться к его членам. Иногда мы просто хотим передать его последующему импорту. Допустим, у нас есть несколько файлов, связанных с формами, и мы хотим подключить их все вместе как одно пространство имён. Мы можем сделать это с помощью @forward:

```
/* forms/_index.scss */
@forward 'input';
@forward 'textarea';
@forward 'select';
@forward 'buttons';

/* styles.scss */
@use 'forms'; /* подключение всех проброшенных членов в пространство имён `forms` */

/* пробросить только миксин `border()` и переменную `$border-color` из модуля `input` */
@forward 'input' show border, $border-color;

/* пробросить все члены модуля `buttons` за исключением функции `gradient()` */
@forward 'buttons' hide gradient;
```

classnames

```
import React from "React";
import cn from "classnames";
import styles from "./component.css";

const Button = ({ label, icon, disabled }) => (
  <button className={cn(styles.button, { [styles.buttonDisabled]: disabled })}>
    <span className={cn(styles.buttonIcon, { [styles.buttonIconDisabled]: disabled
    })}>
      <i className={cn("fa", `fa-${icon}`)} />
    </span>
    <span className={styles.buttonLabel}>{label}</span>
  </button>
);
```