**Раздел 1. Общие понятия:**

## 1)БЛОК:

Функционально независимый компонент страницы, который может быть повторно использован.

*Примеры:* header, signature, table

***2)ЭЛЕМЕНТ*:**

Составная часть блока, которая не может использоваться в отрыве от него.

Особенности:

* [Название элемента](https://ru.bem.info/methodology/naming-convention/#%D0%98%D0%BC%D1%8F-%D1%8D%D0%BB%D0%B5%D0%BC%D0%B5%D0%BD%D1%82%D0%B0) характеризует смысл («что это?» — «пункт»: item, «текст»: text), а не состояние («какой, как выглядит?» — «красный»: red, «большой»: big).
* Структура полного имени элемента соответствует схеме: имя-блока\_\_имя-элемента. Имя элемента отделяется от имени блока двумя подчеркиваниями (\_\_).

Пример

<!-- Блок `search-form` -->

<form class="search-form">

<!-- Элемент `input` блока `search-form` -->

<input class="search-form\_\_input">

<!-- Элемент `button` блока `search-form` -->

<button class="search-form\_\_button">Найти</button>

</form>

**Когда создавать блок, когда — элемент?**

Создавайте блок: Если фрагмент кода может использоваться повторно и не зависит от реализации других компонентов страницы.

Создавайте элемент: Если фрагмент кода не может использоваться самостоятельно, без родительской сущности (блока).

***3)МОДИФИКАТОР***

*Cущность, определяющая внешний вид, состояние или поведение блока либо элемента.*

* [Название модификатора](https://ru.bem.info/methodology/naming-convention/#%D0%98%D0%BC%D1%8F-%D0%BC%D0%BE%D0%B4%D0%B8%D1%84%D0%B8%D0%BA%D0%B0%D1%82%D0%BE%D1%80%D0%B0) характеризует внешний вид («какой размер?», «какая тема?» и т. п. — «размер»: size\_s, «тема»: theme\_islands), состояние («чем отличается от прочих?» — «отключен»: disabled, «фокусированный»: focused) и поведение («как ведет себя?», «как взаимодействует с пользователем?» — «направление»: directions\_left-top).
* Имя модификатора отделяется от имени блока или элемента одним подчеркиванием (\_).

**Принципы работы с модификаторами:**

Модификатор нельзя использовать самостоятельно, модификатор не может использоваться в отрыве от модифицируемого блока или элемента.

(В имени модификатора обязательно нужно указывать имя блока )

**Это позволяет ограничить влияние элементов и модификаторов одного блока на другой.**

**Пример:**

<!-- Модификаторы `button\_size\_m` и `select\_size\_m` не будут влиять друг на друга. -->

<div class="button button\_size\_m">...</div>

<div class="select select\_size\_m">...</div>

**Раздел 2.Разделение страницы на блоки темплейтов**.

***1)ФАЙЛОВАЯ СТРУКТУРА***

***Все блоки хранятся в папке Common***

В папке коммон будут лежать xslt файлы с названием какого то блока страницы(шапка, таблица, блок подписей, итд) и в них будут лежать темплейты- разные реализации этих блоков

Пример:

Common/\_\_header.xslt

Common/\_\_table.xslt

Именования темплейтов - title-form\_\_header, title-form\_\_signature)

-перед элементами два подчеркивания: title-form\_\_header

-перед модификаторами и его значением- одно подчеркивание: title-form\_header-inn\_disabled

**Раздел 3.Правила именования имен**

**Общие правила:**block-name\_\_elem-name\_mod-name\_mod-val

* Имена записываются латиницей в нижнем регистре.
* Для разделения слов в именах используется дефис (-).
* Имя блока задает [пространство имен](https://ru.wikipedia.org/wiki/%D0%9F%D1%80%D0%BE%D1%81%D1%82%D1%80%D0%B0%D0%BD%D1%81%D1%82%D0%B2%D0%BE_%D0%B8%D0%BC%D1%91%D0%BD) для его элементов и модификаторов.
* Имя элемента отделяется от имени блока двумя подчеркиваниями (\_\_).
* Имя модификатора отделяется от имени блока или элемента одним подчеркиванием (\_).
* Значение модификатора отделяется от имени модификатора одним подчеркиванием (\_).
* Значение булевых модификаторов в имени не указывается.

Примеры:  
  
**1)XSLT**

<xsl:template name="header\_\_inn-kpp-block ">

<xsl:template name="header\_\_ inn-kpp-block \_inn\_disabled">

<xsl:template name="header ">

<xsl:template name="header\_\_inn-kpp-block ">

</xsl:template>

</xsl:template>

**1)CSS**

.text\_size\_9px { font-size:9px; }

.menu\_\_text-block\_color\_blue { color: blue; }

.button\_theme\_islands .button\_\_text {line-height: 1.5;}

**Раздел 4.Базовые принципы код-стайла:**

***Модификаторы:*** Разделение CSS кода для блока и его модификаторов<td class="button button\_size\_s">...</td >

.button {

font-family: Arial, sans-serif;

text-align: center;

}

.button\_size\_s {

font-size: 13px;

line-height: 24px;

}

.button\_size\_m {

font-size: 15px;

line-height: 28px;

}

***Принцип единственной ответственности:*** Как и в объектно-ориентированном программировании, принцип единственной ответственности в CSS означает, что каждая CSS-реализация должна иметь одну ответственность.  
<header class="header">

<button class="button header\_\_button">...</button>

</header>

.button {

font-family: Arial, sans-serif;

border: 1px solid black;

background: #fff;

}

.header\_\_button {

margin: 30px;

position: relative;

}

**Принцип открытости/закрытости*:*** Любой HTML-элемент страницы должен быть открыт для модификации, но закрыт для изменения. Разрабатывать новые CSS-реализации следует так, чтобы не пришлось менять уже существующие.