Основы верстки в Mobile SMARTS

Последние изменения: 2024-03-26

Методология разработки в Mobile SMARTS такова, что функционал приложения изначально неотделим от пользовательского интерфейса.

Чтобы сформировать внешний вид приложения (текст, картинки, поля ввода данных), который впоследствии будет отображаться на экране мобильного устройства, мы используем элементы HTML. Они позволяют задать стиль неопределенных в дереве элементов документа, а также генерировать содержимое, которого нет в исходном коде текста.

К тегам форматирования текста в HTML можно отнести теги, изменяющие отображение выделенного фрагмента. Использование форматирования позволяет отобразить информацию на экране в удобном для клиента виде.

Основная задача верстки — сделать интерфейс приложения максимально простым и понятным пользователю, поэтому мы предоставили возможность менять внешний вид приложений на платформе Mobile SMARTS «под клиента» с помощью следующих инструментов:

1. **HTML-теги** — используются для форматирования текста. С их помощью можно задать необходимый цвет, размер, стиль, что позволяет отобразить информацию на экране в удобном пользователям виде, выделить важные сообщения в тексте и т. п.

   Теги строятся по принципу: <имя тега>. Имя тега может состоять из английских букв и цифр. Теги обычно пишутся парами — открывающий тег и соответствующий ему закрывающий. Разница между открывающим и закрывающим тегами в том, что в закрывающем теге после открывающей угловой скобки стоит слеш.

2. **Классы** — позволяют менять стиль элемента в зависимости от действий пользователя, например, поменять цвет кнопки, по которой уже осуществлялся переход.

3. **Атрибуты тегов и события.** Позволяют задать значения для свойств, применимые к данному тегу. Атрибуты размещаются внутри открывающего тега.

   Значения HTML-атрибутов всегда пишутся в двойных кавычках.

На платформе Mobile SMARTS применяются две группы верстки:

- упрощенная верстка;
- полнофункциональная верстка.

> С момента выхода 3.3 платформы у Android-клиента появились встроенные отступы от краев экрана. Данные отступы отключить нельзя.

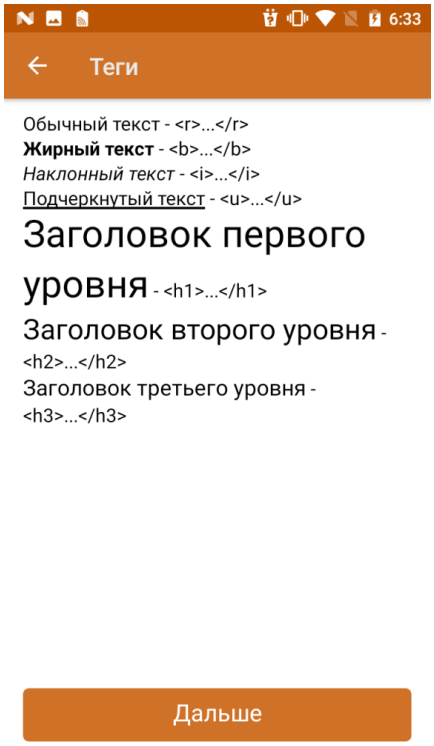
Упрощенная верстка в Mobile SMARTS

Последние изменения: 2024-03-26

Для простого форматирования текста можно использовать теги упрощённой верстки. Основными тегами упрощенной верстки являются:

- **<r> — обычный шрифт;**
- ** — жирный шрифт;**
- **<u> — подчеркнутый шрифт;**
- ** — наклонный шрифт;**
- **<h1> — заголовок первого уровня;**
- **<h2> — заголовок второго уровня;**
- **<h3> — заголовок третьего уровня.**

Для вышеописанных тегов необходимо использовать закрывающий тег, например:



<r>Обычный текст</r>

Жирный текст

Наклонный текст

<u>Подчеркнутый текст</u>

<h1>Заголовок первого уровня</h1>

<h2>Заголовок второго уровня</h2>

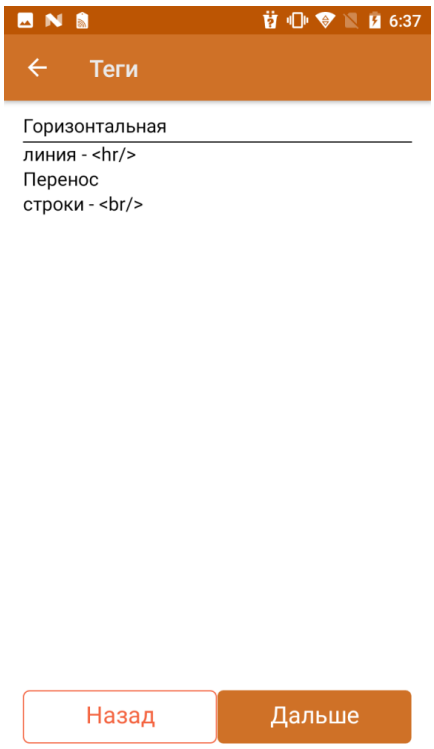
<h3>Заголовок третьего уровня</h3>

Также, существуют теги упрощенной верстки, для которых закрывающий тег не нужен:

- **
 — перенос строки;**

- **<hr/> — горизонтальная линия**

Пример простого синтаксиса:

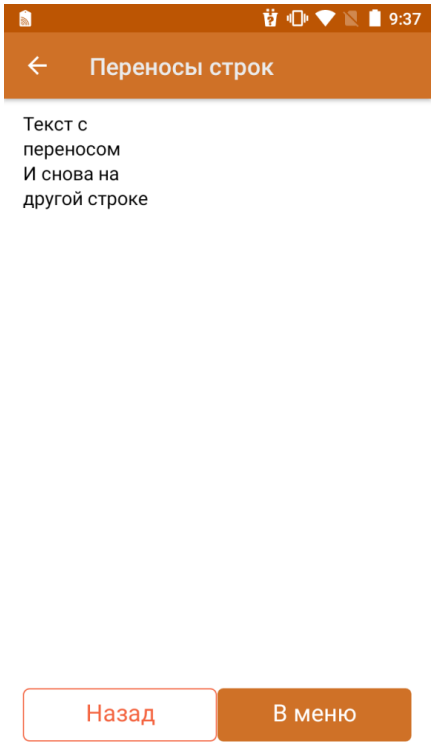


Горизонтальная<hr/>линия — \<hr/>

**Перенос
строки — \
**

Основное отличие упрощенной верстки заключается в правилах формирования отображения: при полнофункциональной верстке переносы строк игнорируются, в то время как в упрощенной нет.

Пример простого синтаксиса:

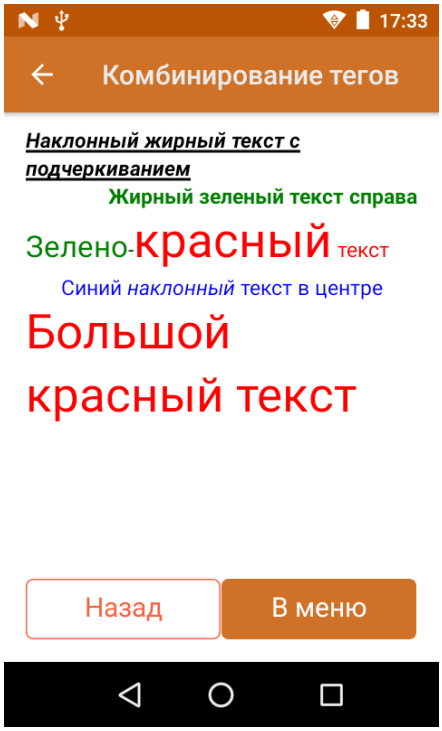


**Текст с
 переносом
И снова на
другой строке**

В случае, если теги упрощенной верстки используются внутри тега , данная верстка будет являться и обрабатываться по правилам блочной верстки.

Для формирования правильного отображения теги можно комбинировать между собой. Главное, следить за тем, где находится закрывающий тег. В следующем примере теги используются совместно с атрибутами(подробнее в статье «Использование атрибутов тегов»).

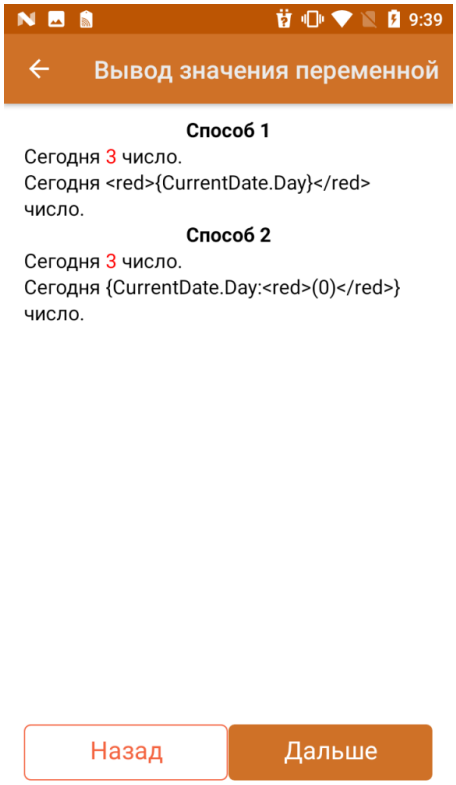
Пример простого синтаксиса:




```
<b><u><i>Наклонный жирный текст с подчеркиванием</i></u></b>

<b color="green» align="right">Жирный зеленый текст справа</b>

<green size="10">Зелено</green>-<red size="14">красный</red> <red>текст</red>

<r color="#0000FF» align="center">Синий <i>наклонный</i> текст в центре</r>

<red size="14">Большой красный текст</red>
```

Вышеуказанные теги можно применять при отображении данных из переменной/свойств объектов.

Пример простого синтаксиса:

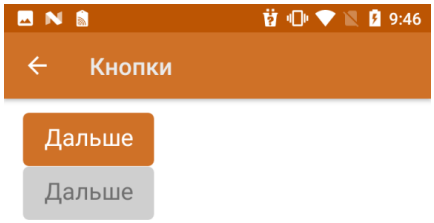



```
<b align="center">Способ 1</b>

Сегодня <red>{CurrentDate.Day}</red> число.

Сегодня \<red>\{CurrentDate.Day\}\</red> число.

<b align="center">Способ 2</b>

Сегодня {CurrentDate.Day:<red>(0)</red>} число.

Сегодня \{CurrentDate.Day:\<red>(0)\</red>\} число.
```

В отличие от вышеуказанных тегов, которые используются для формирования отображения текста в упрощенной верстке, также могут использоваться следующие теги: <button> и <img>.

Тег <button> используется для создания кнопки. Обязательно должен иметь завершающий тег. С помощью данного тега можно создать дополнительные кнопки управления в различных частях окон.

Примеры использования данного тега:




```
<button direction="">Дальше</button>

<button direction=»» enabled="false">Дальше</button>

<button direction=»» visible="false">Дальше</button>
```

С помощью атрибута direction указывается, на какое действие алгоритма будет совершен переход при нажатии. Атрибуты enabled и visible управляют доступностью нажатия/отображения элемента.

Тег <img> используется для отображения изображения. Обязательно должен иметь завершающий тег.
С помощью данного тега можно отображать изображения из файла ресурсов, по ссылке, через переменную из таблицы/по ссылке.

Пример простого синтаксиса:

{logo = "https://www.cleverence.ru/local/templates/cleverence/img/logo.png"}

<img>https://www.cleverence.ru/local/templates/cleverence/img/logo.png</img>

<img>logo.png</img>

<img>testres.logo</img>

<img>{logo}</img>

Само изображение для отображения может быть задано несколькими путями:

Шаблон
Путь к файлу
Описание

<img>\Images\picture.jpg</img> <img>\Flash\Images\picture.jpg</img>
\Images\picture.jpg \Flash\Images\picture.jpg
Изображение находится по заданному абсолютному пути на терминале
<img>picture.jpg</img> <img>Images\picture.jpg</img>
\Application\MobileSMARTS\picture.jpg \ Application\MobilesSMARTS\Images\ picture.jpg
Изображение ищется по пути <Папка программы на терминале>\<заданный относительный путь> если не найдено ищется по пути <Папка программы на терминале>\<папка базы на терминале>\<заданный относительный путь>
<img>{ПеременнаяСПутем}</img>
\Flash\Images\pic1.jpg Переменная в сессии {ПеременнаяСПутем}=”\Flash \Images\pic1.jpg”
Изображение ищется по пути, лежащему в переменной сессии.

Не нашли что искали?

Задать вопрос в техническую поддержку

Полнофункциональная верстка в Mobile SMARTS

Последние изменения: 2024-03-26

Признаком использования полнофункциональной верстки является использование тега . С помощью этого тега можно настроить визуальное расположение вкладываемых в него элементов, а также размер элемента/цвет фона и другие.

Для одного свойства визуального действия доступен только один внешний . При использовании тега всё визуальное отображение в свойстве визуального действия, которое написано вне тега, отображено не будет, однако, вычисляемые выражения будут выполнены.

Подробное описание тега с примерами смотрите в статье «Тег ».

Тег <table>

Данный тег служит контейнером для элементов, определяющих содержимое таблицы и для размещения 2х блоков на одной плоскости. Любая таблица состоит из строк, колонок и ячеек, которые задаются с помощью тегов <tr> и <td>.

Тег <tr> используется для разметки строки таблицы.

Тег <td> предназначен для создания одной ячейки таблицы. Данный тег должен размещаться внутри контейнера <tr>, который в свою очередь располагается внутри тега <table>.

> Все перечисленные теги должны обязательно иметь завершающий тег.

Подробное описание тегов с примерами смотрите в статье «Теги <table>, <tr> и <td>».

Тег <input>

Также существует возможность добавить поле ввода прямо в верстке.

Для этого нужно воспользоваться тегом <input>. У данного тега есть основной атрибут type, с помощью которого определяется, как будет выглядеть поле ввода.

Подробное описание тега с примерами смотрите в статье «Тег <input>».

Использование атрибутов тегов

HTML-теги могут содержать один или несколько атрибутов. Атрибуты добавляются в тег для того, чтобы информировать браузер о том, как данный тег должен отображаться в приложении. Атрибуты задаются в начальном теге элемента и состоят из имени и значения, которые отделяются друг от друга знаком равно (=).

В HTML используются «двойные кавычки», например синтаксис применения атрибута к тегу:

```
<tag attribute="value"></tag>
```

Некоторые атрибуты стандартны для большинства html-тегов, это так называемые общие атрибуты, а есть частные атрибуты, которые используются только для определенных тегов/

Наиболее часто используемыми атрибутами являются универсальные атрибуты (например class, id), определяющие общие свойства элементов, и локализующие атрибуты, которые указывают на свойства языка написания содержимого элемента.

Не нашли что искали?

Задать вопрос в техническую поддержку

Общие атрибуты тегов в Mobile SMARTS

Последние изменения: 2024-03-26

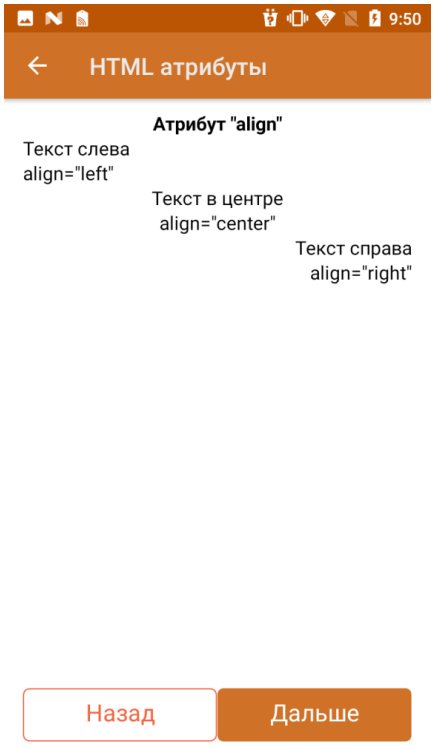
Общие (глобальные, универсальные) атрибуты применяются практически ко всем элементам HTML, поэтому выделены в отдельную группу, чтобы не повторять их для каждого элемента.

Список атрибутов;

- align
- width/height
- class
- id
- color
- maxlines
- style

Атрибут
Значение
Описание
align
right, left, center
горизонтальное выравнивание

Для атрибута align синтаксис будет выглядеть следующим образом:




```
<b align="center">Атрибут "align"</b>

<r align="left">Текст слева</r>

<r align="left">align="left"</r>

<r align="center">Текст в центре</r>

<r align="center">align="center"</r>

<r align="right">Текст справа</r>

<r align="right">align="right"</r>
```

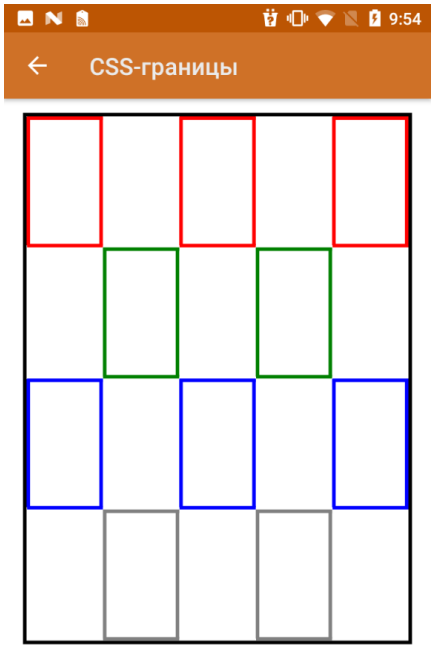
Атрибут
Значение
Описание
width
числовое значение
определяет ширину таблиц, изображений или ячеек таблицы
height
числовое значение
определяет высоту таблиц, изображений или ячеек таблицы

На примере форматирования кнопки для атрибутов width и height синтаксис будет выглядеть следующим образом:

```
<button direction="4" width="100%" height="15%">Дальше</button>
```

Атрибут
Значение
Описание
class
правило класса или стиль класса
Задает стилевой класс, который позволяет связать определенный тег со стилевым оформлением. В значении допускается указывать сразу несколько классов, разделяя их между собой пробелом

Для атрибута class синтаксис будет выглядеть следующим образом:



CSS: class="имя_класса"

Еще

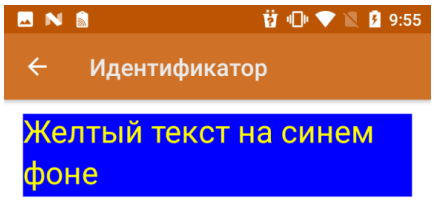
CSS: class="имя_класса"

```
<div width="100%">
<table width="100%" style="border:3dp solid black;" >
<tr>
<td width="20%" height="20%" class="redBorder"></td>
<td></td>
<td class="redBorder"></td>
<td></td>
<td class="redBorder"></td>
</tr>
<tr>
<td width="20%" height="20%"></td>
<td class="greenBorder"></td >
<td ></td>
<td class="greenBorder"></td>
<td></td>
</tr>
<tr>
<td width="20%" height="20%" class="blueBorder"></td>
<td></td>
<td class="blueBorder"></td>
<td></td>
<td class="blueBorder"></td>
</tr>
<tr>
<td width="20%" height="20%"></td>
<td class="grayBorder"></td>
<td></td>
<td class="grayBorder"></td>
<td></td>
</tr>
</table>
</div>
```

Атрибут
Значение
Описание

id
идентификатор должен обязательно начинаться с латинского символа и может содержать в себе латинские буквы (A-Z, a-z), цифры (0-9), символ дефиса (-) и подчеркивания (_). Использование русских букв в именах идентификатора недопустимо.
задает стилевой идентификатор — уникальное имя элемента, которое используется для изменения его стиля; идентификатор в коде документа должен быть в единственном экземпляре, иными словами, встречаться только один раз.

Для атрибута id синтаксис будет выглядеть следующим образом:




```
CSS:
#example{
font-size: 12pt;
background-color: blue;
color: yellow;
}
```

```
<div id="example">Желтый текст на синем фоне</div>

<div id="example">Здесь тоже используется этот идентификатор, но этот блок не отобразится</div>
```

Атрибут
Значение
Описание
color
цвет
устанавливает цвет текста, используя либо название цвета, либо шестнадцатеричный формат #RRGGBB (см. Таблица цветов в HTML)

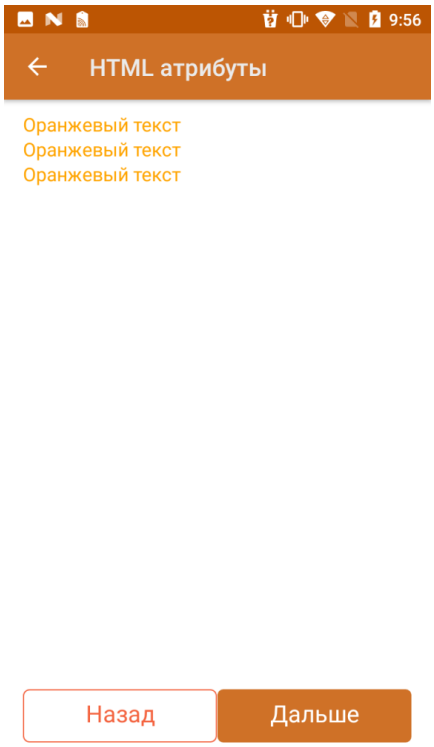
Для задания цвета текста используется тег с именем цвета или необязательный атрибут color="...".

> Если атрибут не задан, то для вывода текста используется цвет по умолчанию, в соответствии со стилем отображения.

Цвет может задаваться тремя способами:

- тег — имя цвета;
- название цвета на английском языке;
- код цвета в шестнадцатеричном виде.

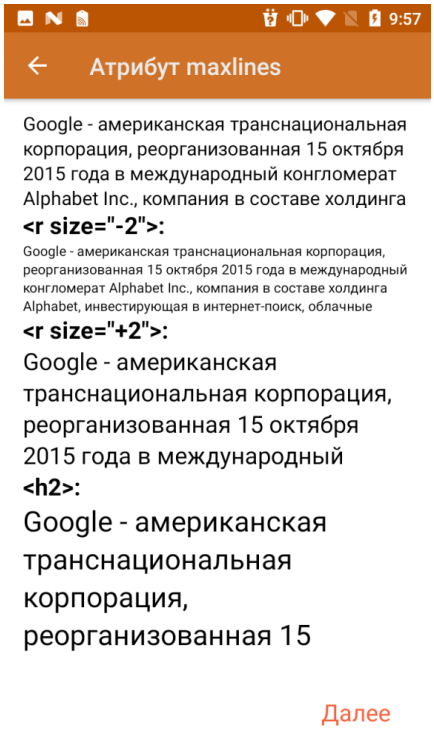
Для атрибута color в зависимости от варианта введения метаданных синтаксис будет выглядеть следующим образом:




```
<Orange>Оранжевый текст</Orange>
<r color="Orange">Оранжевый текст</r>
<r color="#FFA500">Оранжевый текст</r>
```

Атрибут
Значение
Описание
maxlines
числовое значение
обрезает текст до N строк

Для атрибута maxlines синтаксис будет выглядеть следующим образом:




```
<div align="left">
<r maxlines="4">Google — американская транснациональная корпорация, реорганизованная 15 октября 2015 года в международный конгломерат Alphabet Inc., компания в составе холдинга Alphabet, инвестирующая в интернет-поиск, облачные вычисления и рекламные технологии.</r><br />
<b size="+2">\<r size="-2">:</b><br />
<r size="-2" maxlines="4">Google — американская транснациональная корпорация, реорганизованная 15 октября 2015 года в международный конгломерат Alphabet Inc., компания в составе холдинга Alphabet, инвестирующая в интернет-поиск, облачные вычисления и рекламные технологии.</r><br />
<b size="+2">\<r size="+2">:</b><br />
<r size="+2" maxlines="4">Google — американская транснациональная корпорация, реорганизованная 15 октября 2015 года в международный конгломерат Alphabet Inc., компания в составе холдинга Alphabet, инвестирующая в интернет-поиск, облачные вычисления и рекламные технологии.</r><br />
<b size="+2">\<h2>:</b><br />
<h2 maxlines="4">Google — американская транснациональная корпорация, реорганизованная 15 октября 2015 года в международный конгломерат Alphabet Inc., компания в составе холдинга Alphabet, инвестирующая в интернет-поиск, облачные вычисления и рекламные технологии.</h>
</div>
```

Ограничить длину выводимого текста можно также с помощью форматирования:

{SomeString = "Google — американская транснациональная корпорация, реорганизованная 15 октября 2015 года в международный конгломерат Alphabet Inc., компания в составе холдинга Alphabet, инвестирующая в интернет-поиск, облачные вычисления и рекламные технологии."}

{SomeString:Исходная строка: (0)}

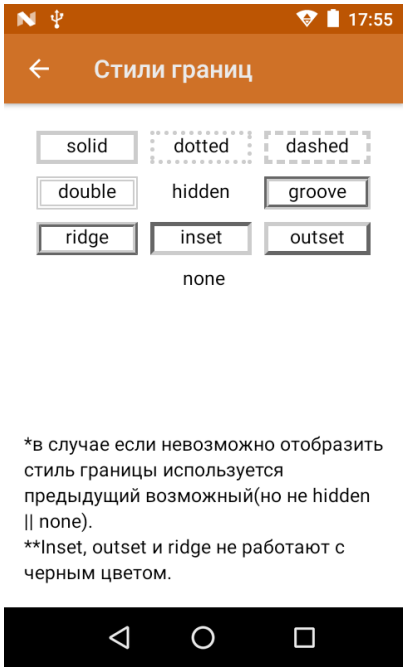
{SomeString:Обрезана до 100 символов: (0:T100)}

{SomeString:Обрезана до 100 символов + троеточие: (0:E100)}

{SomeString:Начало и конец + троеточие в середине: (0:M100)}

Атрибут
Значение
Описание
style
в качестве значений указываются стилевые правила
применяется для определения стилей элементов с помощью правил CSS

Для атрибута style синтаксис будет выглядеть следующим образом:



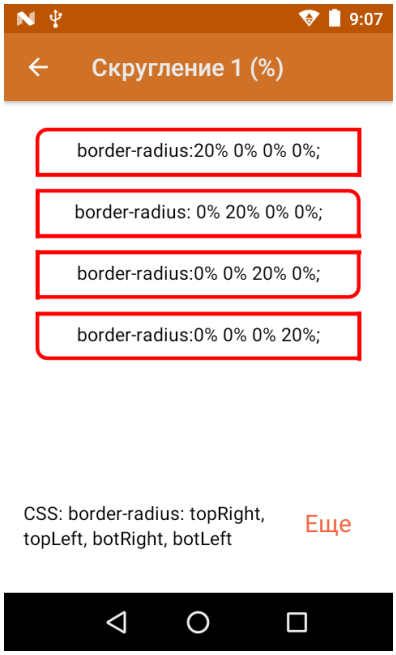

```
<div>
<table cols="3" width="100%" cellspacing="10dp" align="center" valign="middle">
<tr>
<td style="border:3dp solid #cccccc;" >solid</td>
<td style="border:3dp dotted #cccccc;">dotted</td>
<td style="border:3dp dashed #cccccc;">dashed</td>
</tr>
<tr>
<td style="border:3dp double #cccccc;">double</td>
<td style="border:3dp hidden #cccccc;">hidden</td>
<td style="border:3dp groove #cccccc;">groove</td>
</tr>
<tr>
<td style="border:3dp ridge #cccccc;">ridge</td>
<td style="border:3dp inset #cccccc;">inset</td>
<td style="border:3dp outset #cccccc;">outset</td>
</tr>
<tr>
<td colspan="3" style="border:3dp none #cccccc;">none</td>
</tr>
</table>
</div>
```

С помощью стилевого правила border-radius атрибута style можно устанавливать радиус скругления углов рамок. Можно использовать одно, два, или четыре значения.

В зависимости от количества значений скругление будет применяться (по очереди):

- 1 — для всех четырех углов;
- 2 — первое значение определяет радиус скругления верхнего левого и нижнего правого, второе — верхнего правого и нижнего левого углов;
- 3 — первое значение определяет радиус скругления верхнего левого угла, второе — одновременно для верхнего правого и нижнего левого, третье — для нижнего правого
- 4 — верхнего левого, верхнего правого, нижнего правого, нижнего левого.

В качестве значения принимаются числа в поддерживаемом формате (рекомендуется использовать dp) или проценты. Синтаксис выглядит следующим образом:




```
<div>
<table width="100%" align="center" style="vertical-align: middle;" cellspacing="10dp">
<tr>
<td width="100%" height="10%" style="border: 3dp solid red; border-radius:20% 0% 0% 0%; ">
border-radius:20% 0% 0% 0%; </td>
</tr>
<tr>
<td width="100%" height="10%" style="border: 3dp solid red; border-radius:0% 20% 0% 0%; ">
border-radius: 0% 20% 0% 0%; </td>
</tr>
<tr>
<td width="100%" height="10%" style="border: 3dp solid red; border-radius:0% 0% 20% 0%; ">
border-radius:0% 0% 20% 0%; </td>
</tr>
<tr>
<td width="100%" height="10%" style="border: 3dp solid red; border-radius:0% 0% 0% 20%; ">
border-radius:0% 0% 0% 20%; </td>
</tr>
</table>
</div>
```

Еще один пример с использованием различного количества значений:

```
<div>
   <table width="100%" align="center" valign="middle" cellspacing="10dp">
   <tr>
   <td>
   CSS: border-radius: topRight+topLeft, botRight + botLeft
   </td>
   </tr>
   <tr>
   <td width="100%" height="10%" style="border: 3dp dotted red; border-radius:15dp 0dp; ">
   border-radius:15dp 0dp;
   </td>
   </tr>
   <tr>
   <td>
   CSS: border-radius: all
   </td>
   </tr>
   <tr>
   <td width="100%" height="10%" style="border: 3dp inset red; border-radius:15dp; ">
   border-radius: 15dp;
   </td>
   </tr>
   <tr>
   <td>CSS: border-radius: topLeft, topRight, botRight, botLeft</td>
   </tr>
   <tr>
   <td width="100%" height="10%" style="border: 3dp outset red; border-radius:5dp 10dp 15dp
20dp; ">border-radius: 5dp 10dp 15dp 20dp;</td>
   </tr>
   <tr>
   <td>CSS: border-radius: topLeft, topRight + botLeft, botRight</td>
   </tr>
   <tr>
   <td width="100%" height="10%" style="border: 3dp inset red; border-radius:5dp 10dp 20dp;
">border-radius: 5dp 10dp 20dp;</td>
   </tr>
   </table>
   </div>
```

форматирование

Не нашли что искали?

Задать вопрос в техническую поддержку

Частные атрибуты тегов в Mobile SMARTS

Последние изменения: 2024-03-26

Многие атрибуты в HTML являются общими для всех элементов, однако большинство из них являются специфическими для данного элемента или группы элементов. Это так называемые **частные атрибуты**.

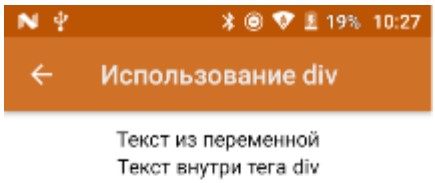
Атрибуты обеспечивают дополнительную информацию об элементе, при этом они всегда определяются в начальном теге независимо от того парный это тег, либо одиночный. Пользователь не может создавать свои собственные атрибуты или использовать значения, не определенные спецификацией, так как это может вызывать проблемы правильной интерпретации.

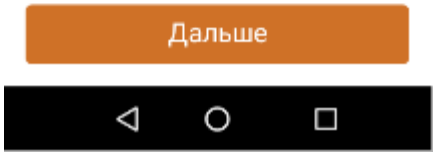
Рассмотрим частные атрибуты подробнее с примерами.

- Тег
- Тег <p>
- Теги <table>, <tr>, <td>
- Тег <input>
- Тег <img>
- Тег <button>
- Тег <a>

Тег и его частные атрибуты

Пример простого синтаксиса:

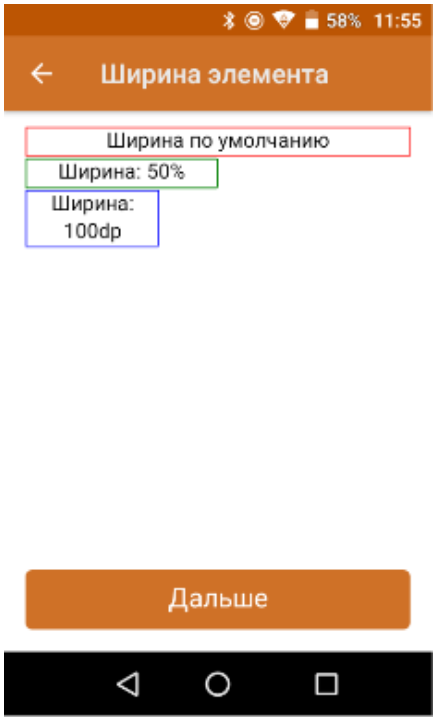





```
Текст вне тега div
{text = "Текст из переменной"}
<div>
<p>{text}</p>
<p>Текст внутри тега div</p>
</div>
```

По умолчанию, при использовании тега используется вся доступная область экрана, изменить размер можно с помощью атрибута width.

Пример простого синтаксиса:




```
<div>
<div style="border:1dp solid red;">
Ширина по-умолчанию
</div>
<div style="border:1dp solid green; width:50%;">
Ширина: 50%
</div>
<div style="border:1dp solid blue; width:100dp;">
Ширина: 100dp;
</div>
</div>
```

Как видим из примера, блоки размещаются вертикально.

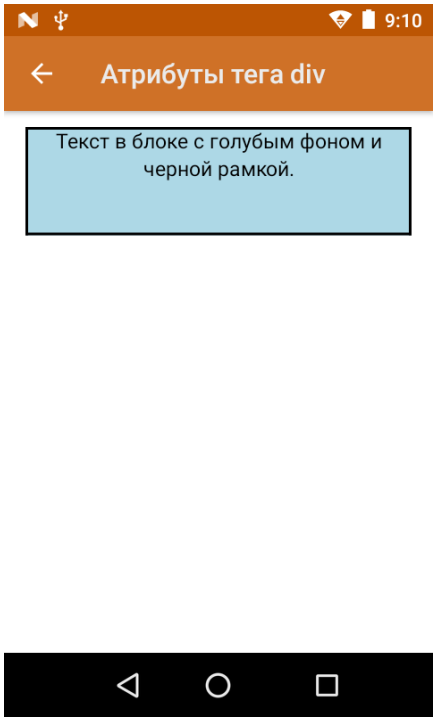
Для тега доступны следующие частные атрибуты:

Атрибут	Значение	Описание
align	right, left, center	горизонтальное выравнивание
bgcolor	цвет	определяет цвет фона блока/контейнера
border	числовое значение	определяет толщину границ блока/контейнера

height
числовое значение
определяет высоту

maxlines
числовое значение
обрезает текст до N строк

Еще пример использования атрибутов тега :

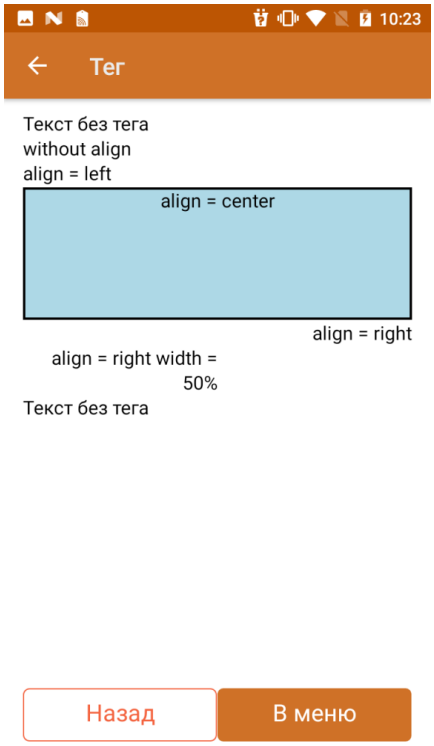



```
<div height="20%" align="center" bgcolor="lightblue" border="2dp">Текст в блоке с голубым фоном и черной рамкой.</div>
```

Тег <p> и его частные атрибуты

Представляет собой абзац. По умолчанию использует всю доступную область родительского элемента.

Пример синтаксиса:



Текст без тега without align align = left align = center align = right align = right width = 50% Текст без тега

Для тега доступны следующие частные атрибуты:

Атрибут	Значение	Описание
align	right, left, center	горизонтальное выравнивание

bgcolor
цвет
определяет цвет фона блока/контейнера

border
числовое значение
определяет толщину границ блока/контейнера
height
числовое значение
определяет высоту
width
числовое значение
ширина элемента
maxlines
числовое значение
обрезает текст до N строк

Теги <table>, <tr>, <td> и их частные атрибуты

Элемент <table> служит контейнером для элементов, определяющих содержимое таблицы.

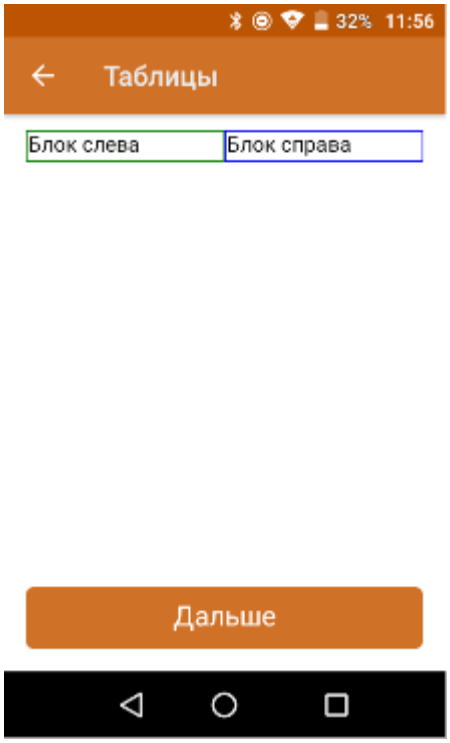
Любая таблица состоит из строк и ячеек, которые задаются с помощью тегов <tr> и <td>.

Внутри <table> допустимо использовать такие атрибуты как bgcolor, cellpadding, cols, valign (подробнее см. список ниже).

> **Теги <table>, <tr>, <td> обязательно должны иметь завершающий тег.**

Тег <td> предназначен для создания одной ячейки таблицы. Данный тег должен размещаться внутри контейнера <tr>, который в свою очередь располагается внутри тега <table>.

Пример простого синтаксиса:




```
<div>
<table width="100%">
<tr>
<td style="border:1dp solid green;" width="50%">
Блок слева
</td>
<td style="border:1dp solid blue;">
Блок справа
</td>
</tr>
</table>
</div>
```

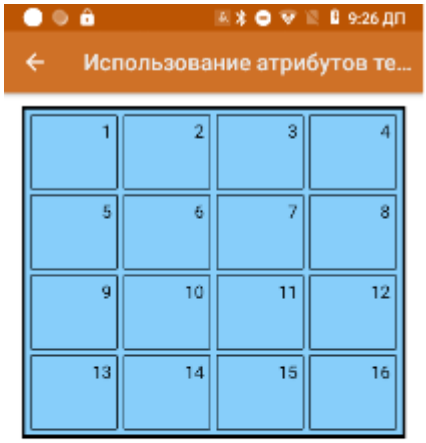
Для тега <table> доступны следующие частные атрибуты:

Атрибут
Значение
Описание

align
right, left, center
горизонтальное выравнивание

bgcolor
цвет
определяет цвет фона таблицы
border
числовое значение
определяет толщину границ таблицы, а также включает границы ячеек толщиной 1px
height
числовое значение
определяет высоту таблиц
valign
top, middle, bottom
вертикальное выравнивание содержимого ячеек
width
числовое значение
определяет ширину таблиц
cellpadding
любое целое значение в пикселах или процентах от доступного пространства
определяет отступ содержимого ячейки от границы ячейки
cellspacing
любое целое положительное число
определяет отступ между ячейками
cols
любое целое положительное число
определяет количество колонок

Пример использования несколько атрибутов сразу:




```
<div width="100%" height="50%">
<table width="100%" cols="4" align="right" valign="top" cellpadding="5dp" cellspacing="5dp" border="3dp"
bgcolor="#87cefa">
<tr>
<td height="25%">1</td>
<td>2</td>
<td>3</td>
<td>4</td>
</tr>
<tr>
<td height="25%">5</td>
<td>6</td>
<td>7</td>
<td>8</td>
</tr>
<tr>
<td height="25%">9</td>
<td>10</td>
<td>11</td>
<td>12</td>
</tr>
<tr>
<td height="25%">13</td>
<td>14</td>
<td>15</td>
<td>16</td>
</tr>
</table>
</div>
```

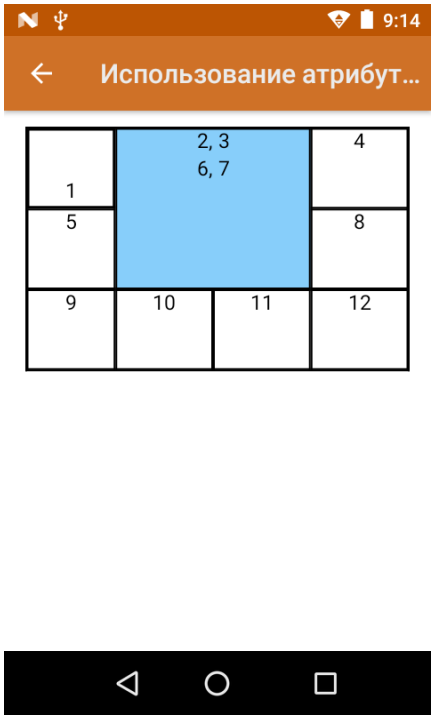
Тег <tr> используется для разметки строки таблицы. Обязательно должен иметь завершающий тег. Данный тег не имеет дополнительных атрибутов.

Тег <td> используется внутри тега <tr> и определяет ячейку таблицы. Обязательно должен иметь завершающий тег.

Для данного тега доступны следующие частные атрибуты:

Атрибут
Значение
Описание
align
right, left, center
горизонтальное выравнивание в данной ячейке
bgcolor
цвет
определяет цвет фона ячейки
border
числовое значение
определяет толщину границ ячеек
colspan
любое целое положительное число больше 1
определяет горизонтальное количество ячеек для объединения
height
числовое значение
определяет высоту ячейки
rowspan
любое целое положительное число больше 1
определяет вертикальное количество ячеек для объединения
valign
top, middle, bottom
вертикальное выравнивание содержимого ячеек
width
числовое значение
определяет ширину ячейки

Пример использования нескольких атрибутов сразу:



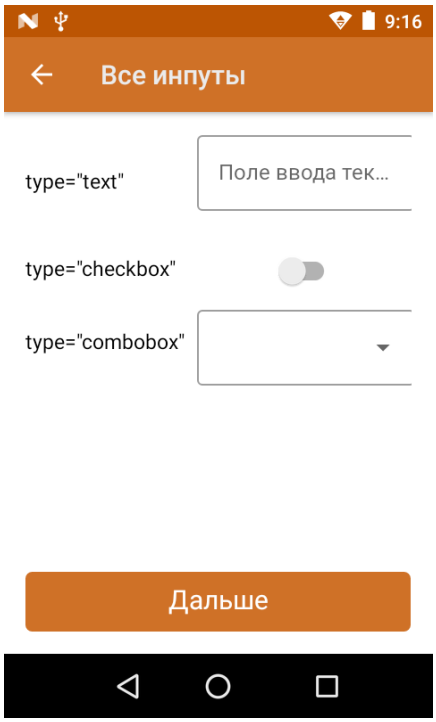

```
<div width="100%">
<table width="100%" border="1dp" align="center">
<tr>
<td height="15%" valign="bottom" border="2dp" bgcolor="#87cefa">1</td>
<td colspan="2" rowspan="2">2, 3<br />6, 7</td>
<td>4</td>
</tr>
<tr>
<td height="15%">5</td>
<td>8</td>
</tr>
<tr>
<td height="15%">9</td>
<td>10</td>
<td>11</td>
<td>12</td>
</tr>
</table>
</div>
```

Тег <input> и его частные атрибуты

Тег используется для создания полей ввода текста, чекбоксов или выпадающих списков.

У данного тега есть основной атрибут type с помощью которого определяется, как будет выглядеть поле ввода.

Рассмотрим на следующем примере:




```
<div><div>
<table col="2" width="100%"style="vertical-align:middle;">
<tr>
<td width="30%" height="10%">
type="text"
</td>
<td width="70%" align= "center">
<input type="text" width="90%" tabIndex="1" placeholder="Поле ввода текста" value="{textValue}" datatype="char"
/>
</td>
</tr>
<tr>
<td width="30%" height="10%">
type="checkbox"
</td>
<td width="70%" align= "center">
<input type="checkbox" tabIndex="2" value="{chkbxValue}" onselected ="1" />
</td>
</tr>
<tr>
<td width="30%" height="10%">
type="combobox"
</td>
<td width="70%" align= "center">
<input type="combobox" width="90%" tabIndex="3" placeholder ="пусто" source="СтрокиДляОтображения.Rows"
listItemDisplayTemplate ="{Item.Наименование}" SelectedValue="{listValue}" onselected="Возврат" />
</td>
</tr>
</table>
</div>
```

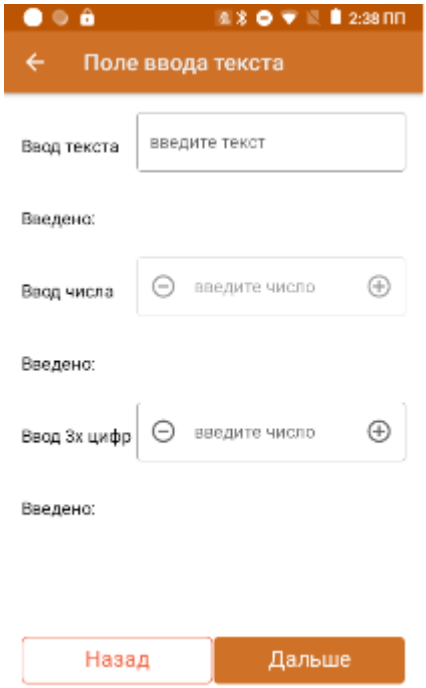
Для создания полей ввода текста доступны следующие частные атрибуты:

Атрибут
Значение
Описание

enabled
true/false, булевая переменная или выражение
отвечает за доступность элемента

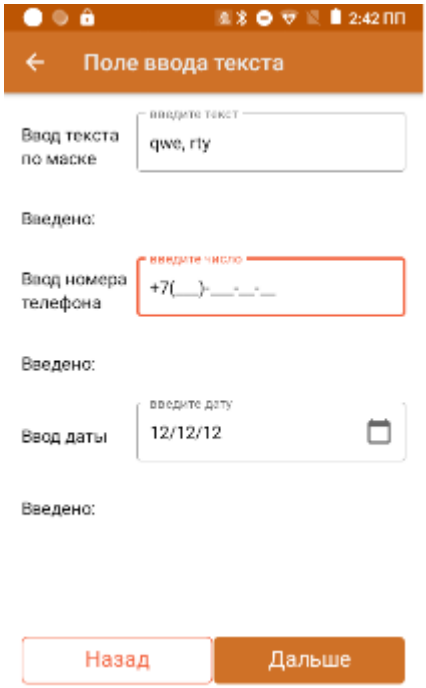
placeholder
текстовая строка, если внутри строки предполагается пробел, ее необходимо брать в двойные или одинарные кавычки
подсказка, отображаемая в тот момент, когда поле ввода пусто (android)
datatype
decimal, string, datetime
тип данных, которые можно вводить
height
числовое значение
определяет высоту поля ввода
format
типы входных данных в формате HTML
принимает regExp, по которому происходит валидация введенных данных
width
числовое значение
определяет ширину поля ввода
value
имя переменной
переменная для занесения вводимой строки value = {var}
mask
содержит маску для текстовых/числовых данных
содержит в себе шаблон, с помощью которого формируются вводимые данные Синтаксис: mask = "##.##" — для ввода чисел mask = "__.__" — для ввода букв и чисел
onBlur
имя действия
указывает действие, на которое будет выполнен переход по смене фокуса (для ОС Android)
id/class/style
см. Общие атрибуты
работают по стандартным правилам HTML

Пример использования нескольких атрибутов сразу:




```
<div width="100%">
<table cols="2" width="100%" style="vertical-align:middle;">
<tr>
<td width="30%" height="10%" >Ввод текста</td>
<td width="70%" ><input type="text" value="{textValue}" width="100%" onBlur="Возврат" placeholder="введите текст" /></td>
</tr>
<tr>
<td height="10%">Введено:</td>
<td>{textValue:(0)}</td>
</tr>
<tr>
<td>Ввод числа</td>
<td>
<input type="text" value="{decimalValue}" enabled="false" width="100%" onblur="Возврат" placeholder="введите число" datatype="decimal" /></td>
</tr>
<tr>
<td height="10%">Введено:</td>
<td>{decimalValue:(0)}</td>
</tr>
<tr>
<td>Ввод 3х цифр</td>
<td><input type="text" value="{formatValue}» width="100%" onblur="Возврат" placeholder="введите число" format="[0-9]{3}" datatype="decimal" /></td>
</tr>
<tr>
<td height="10%">Введено:</td>
<td>{formatValue:(0)}</td>
</tr>
</table>
</div>
```

Пример использования атрибутов mask и pattern:

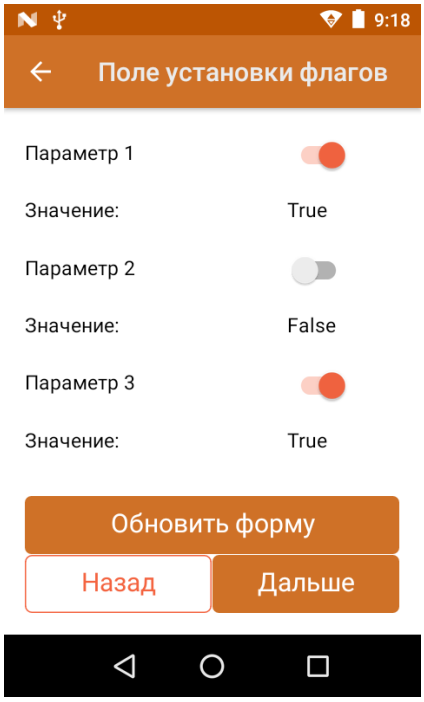



```
<div width="100%">
<table cols="2" width="100%" style="vertical-align:middle;">
<tr>
<td width="30%" height="10%" >Ввод текста по маске</td>
<td width="70%" >
<input type="text" value="{maskValue}" width="100%" onBlur="Возврат" placeholder="введите текст" mask="___,
___" /></td>
</tr>
<tr>
<td height="10%">Введено:</td>
<td>{maskValue:(0)}</td>
</tr>
<tr>
<td>Ввод номера телефона</td>
<td>
<input type="text" value="{numValue}" width="100%" onblur="Возврат" placeholder="введите число»
mask="+7(###)-###-##-##» /></td>
</tr>
<tr>
<td height="10%">Введено:</td>
<td>{numValue:(0)}</td>
</tr>
<tr>
<td>Ввод даты</td>
<td><input type="text" value="{dateValue}" width="100%" onblur="Возврат" placeholder="введите дату"
pattern="dd.MM.yy" datatype="datetime" /></td>
</tr>
<tr>
<td height="10%">Введено:</td>
<td>{dateValue:(0)}</td>
</tr>
</table>
</div>
```

Для чекбокса доступны следующие частные атрибуты:

Атрибут
Значение
Описание
value
для переключателей уникально определяет каждый элемент, с тем, чтобы клиентская или серверная программа могла однозначно установить, какой пункт выбрал пользователь
переменная для занесения вводимой строки value = {var}
id/class/style
см. Общие атрибуты
работают по стандартным правилам HTML

Пример использования нескольких атрибутов сразу:

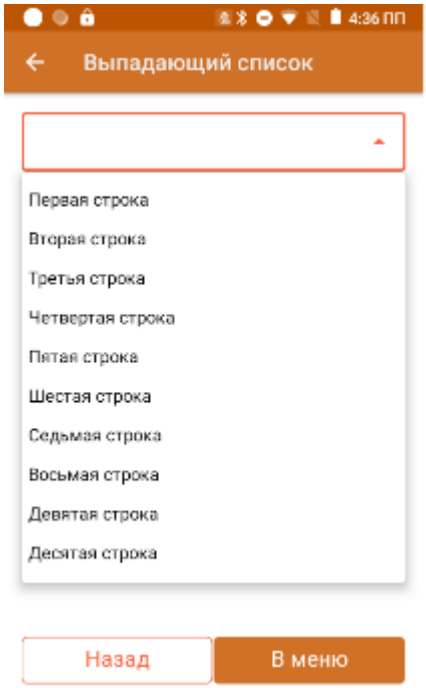



```
<div width="100%" >
<table cols="2" width="100%" style="vertical-align:middle;" >
<tr>
<td width="70%" height="10%">Параметр 1</td>
<td width="30%"><input type="checkbox" value="{par1}" onblur="Возврат" /></td>
</tr>
<tr>
<td width="70%" height="10%">Значение:</td>
<td width="30%">{par1:True;False}</td>
</tr>
<tr>
<td width="70%" height="10%">Параметр 2</td>
<td width="30%"><input type="checkbox" value="{par2}" onblur="Возврат" /></td>
</tr>
<tr>
<td width="70%" height="10%">Значение:</td>
<td width="30%">{par2:True;False}</td>
</tr>
<tr>
<td width="70%" height="10%">Параметр 3</td>
<td width="30%"><input type="checkbox" value="{par3}" onblur="Возврат" /></td>
</tr>
<tr>
<td width="70%" height="10%">Значение:</td>
<td width="30%">{par3:True;False}</td>
</tr>
</table>
</div>
```

Для комбобокса доступны следующие частные атрибуты:

Атрибут
Значение
Описание

enabled
true|false
отвечает за доступность элемента
height
числовое значение
определяет высоту поля ввода
width
числовое значение
определяет ширину поля ввода
value
является атрибутом без значения
атрибут, в котором указывается переменная для занесения выбранной строки Синтаксис: value="{var}"
onSelected
имя действия
указывает действие, на которое будет осуществлен переход после выбора элемента выпадающего списка (для ОС Android)
source
источник элементов для показа
атрибут, который используется для получения коллекции строк. Синтаксис: source = "ItemsCollection". Для того, чтобы передать строки таблицы, необходимо явно (!) указывать строки, т. е.: source = "TableName.Rows"
ListItemDisplayTemplate
если строка имеет поле ShortName, то будет отображаться оно, если нет — поле Имя
позволяет выбрать столбец коллекции для отображения. Синтаксис: ListItemDisplayTemplate = "{Item.ColumnName}"
ListItemValueTemplate
шаблон вычисления значения поля
при выборе позиции списка позволяет произвести вычисление с полем выбранной строки. Синтаксис: ListItemValueTemplate = "{Item.ColumnName = 3+2}"
id|class|style
см. Общие атрибуты
работают по стандартным правилам HTML




```
<div>
<input type="combobox" source="СтрокиДляОтображения.Rows" value ="{SelPos}" listItemDisplayTemplate ="
{Item.Наименование}" listItemValueTemplate="{Item.ВычисляемоеПоле=Item.Код+Item.Характеристика}"
onselected="Возврат" /><br />
{SelPos.Наименование:Выбрано: (0), }{SelPos.Код:(0), }{SelPos.ВычисляемоеПоле:(0)}<br />
</div>
```

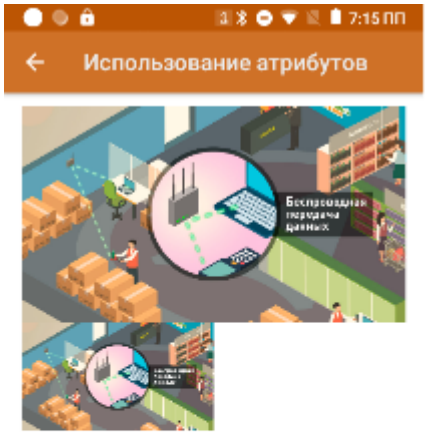
Тег <img> и его частные атрибуты

Тег используется для отображения изображения. Обязательно должен иметь завершающий тег. С помощью данного тега можно отображать изображения из файла-ресурсника, хранимого в папке «Documents» базы, по ссылке, через переменную из таблицы/по ссылке.

Для отображения изображений доступны следующие частные атрибуты:

Атрибут	Значение	Описание
width	числовое значение	определяет ширину изображения

height
числовое значение
определяет высоту изображения
size
stretch
атрибут, с помощью которого указывается поведение формирования ширины/высоты изображения. В качестве параметра можно указать только stretch (о других неизвестно). Если указано stretch — в случае если изображение больше по ширине или высоте оно будет подогнано под максимальный допустимый размер, ограниченный верстkой
tcolor
цвет
id/class/style
см. Общие атрибуты
работают по стандартным правилам HTML




```
<div width="100%">
<img size="stretch">bigPict.jpg</img>
<img width="50%" height="30%">bigPict.jpg</img>
</div>
```

Тег <button> и его частные атрибуты

Тег используется для создания кнопки. Обязательно должен иметь завершающий тег. С помощью данного тега можно создать дополнительные кнопки управления в различных частях окон (в верхней/нижней части/в списке/в самом окне/etc).

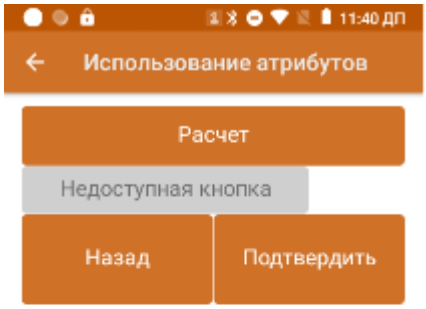
Единственное визуальное действие. в котором нельзя использовать данный тег — действие «Меню», так как оно само состоит из кнопок.

Для создания полей ввода текста доступны следующие частные атрибуты:

Атрибут
Значение
Описание

width
числовое значение
определяет ширину поля ввода
height
числовое значение
определяет высоту поля ввода
enabled
нет
указывает, доступна кнопка для нажатия или нет
direction
указывается название действия для перехода или: cancel — отмена действия finishproc — возврат на одно действие return — завершить операцию abort — прервать операцию
указывает, переход на какое действие будет совершен при нажатии кнопки
visible
отображает элемент как видимый
указывает, отображать кнопку или нет
type="submit"
кнопка для отправки данных формы на сервер
означает, что по нажатию кнопки будет обработаны введенные в поля ввода данные и произведен переход на следующее действие. Раньше, для выполнения подобного действия использовался атрибут direction="ok"
id|class|style
см. Общие атрибуты
работают по стандартным правилам HTML
command
Пример: command="x = y + 1"
позволяет производить простые вычисления без перехода на другое действие

Пример использования атрибутов кнопок:

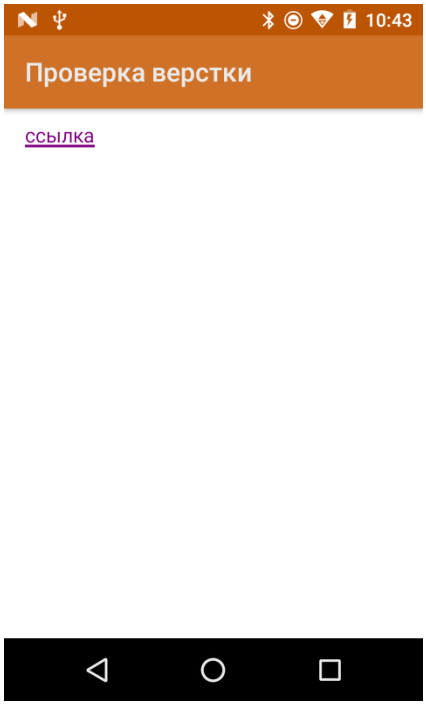



```
<div height="100%" width="100%">
<button width="100%" height="10%" command="value=2+3">Расчет</button><br />
<button width="75%" enabled="false">Недоступная кнопка</button><br />
<button width="100%" visible="false">Невидимая кнопка</button><br />
<button width="50%" height="15%" direction="finishproc">Назад</button><button width="50%" height="15%"
direction="Далее" type="submit">Подтвердить</button>
</div>
```

Тег <a> и его частные атрибуты

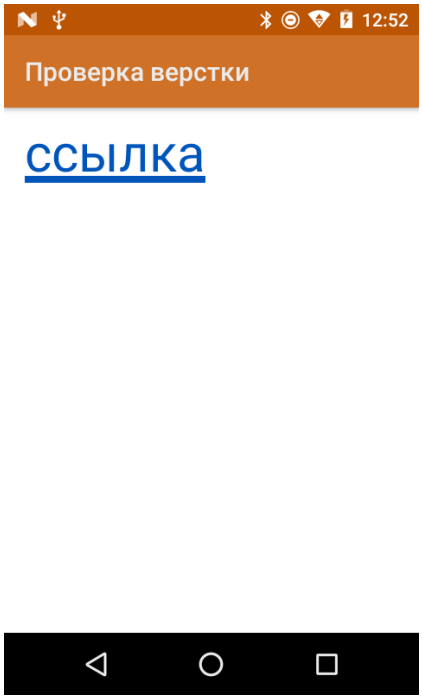
Тег используется для создания гиперссылки на другое действие в алгоритме. Обязательно должен иметь закрывающий тег. Сама ссылка определяется через атрибут href. С помощью атрибута style можно задать цвет гиперссылки.

Пример синтаксиса:



ссылка

Цвет можно задавать как через hex коды цветов, так и по названию цвета.



style="color:purple"

Атрибут size задаёт размер текста гиперссылки

ссылка

Для создания ссылки можно вместо атрибута href использовать атрибут direction

ссылка

форматирование

Не нашли что искали?

Задать вопрос в техническую поддержку

Использование классов стилей в Mobile SMARTS

Последние изменения: 2024-03-26

Mobile SMARTS поддерживает работу со многими ТСД, у которых разные размеры и расширения экрана. Для простоты настройки отображения текстов и кнопок для конкретного мобильного устройства предусмотрен стиль клиентского приложения.

> Для мобильных устройств на базе ОС Windows CE интерфейс форматируется согласно инструкции «Стиль клиентского приложения на ТСД».

Создание классов стилей

Для того, чтобы каждый раз не писать одинаковые параметры, у множества элементов необходимо использовать уже адаптированные стили.

У данного подхода несколько плюсов:

- класс стиля находится в одном месте и если необходимо изменить, например, цвет или размер шрифта, сделать это можно сразу везде без вмешательства в код верстки;
- для устройств на разных ОС (Win/ CE/ Android) можно задавать разные параметры для одних и тех же классов. Это может быть полезно когда разница в разрешениях экранов слишком велика;
- упрощение читаемости html кода.

Файлы стилей должны располагаться на сервере в папке «Documents» (приведем возможные имена файлов):

- **global.css** — общий файл стилей;
- **global.android.css** — файл стилей для Android-устройств;
- **global.win.css** — файл стилей для Win-клиента;
- **global.cf.css** — файл стилей для CE устройств.

Стили из основного файла используются, если не найдены стили из файла для конкретной ОС.

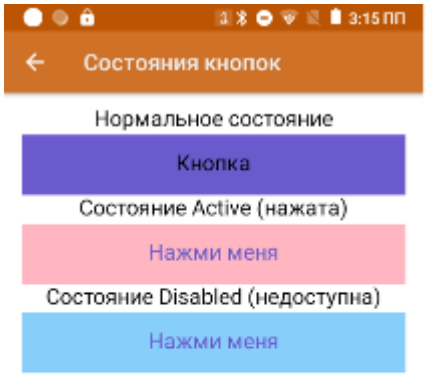
Рассмотрим код класса стиля на следующем примере:

```
.listHead — наименование класса
   {
   text-align: left; - указание, что текст будет прижат к левому краю элемента
   vertical-align: middle; - вертикально посередине элемента
   font-weight: bold; - текст жирный
   font-size: 110%; - размер текста 110%
   color: #666666; - цвет серый
   padding-bottom: 10dp; - снизу элемента отступ внутри элемента 10dp
   }
```

Список свойств достаточно большой, чтобы упомянуть их все в нашей статье, поэтому свойства можно посмотреть на сайте w3schools (поддерживаются не все свойства).

Также рекомендуется ознакомиться с CSS селекторами — это шаблоны, используемые для выбора элементов, которые вы хотите стилизовать (:active|:hover|:disabled используются для кнопок).

Пример использования селекторов для кнопок:




```
CSS:
.bttn3{
background-color: slateblue;
}
.bttn3:disabled{
background-color: lightskyblue;
color: slateblue;
}
.bttn3:active{
background-color: lightpink;
color: slateblue;
}
```

```
<h3 align="center">Нормальное состояние</r>

<button direction="Возврат» width="100%» height="10%» class="bttn3">Кнопка</button>

<h3 align="center">Состояние Active (нажата)</r>

<button direction="Возврат» width="100%» height="10%» class="bttn3">Кнопка</button>

<h3 align="center">Состояние Disabled (недоступна)</r>

<button direction="Возврат» width="100%» height="10%» enabled="false» class="bttn3">Кнопка</button>
```

Использование цветов в стилях приложения

Начиная с версии платформы Mobile SMARTS 3.3 появилась возможность использовать стили приложения.

В классах используется не конкретный цвет, а ссылка на этот цвет. Способ задания выглядит следующим образом:

```
.some_class{
   color: var(--theme-textColorPrimary);
}
```

Аналогично, можно применять прямо в верстке с помощью style.

Список использующихся стандартных цветов:

Цвет темы
Использование
--theme-colorPrimary
Отображение кнопок по умолчанию (цвет фона кнопки обмена в главном меню и кнопки с заливкой, содержимого текстовой и обведенной кнопок, рамки обведенной кнопки)
--theme-colorPrimaryOp24
Цвет фона кнопки обмена и кнопки с заливкой при нажатии
--theme-colorPrimaryOp12
Цвет фона текстовой и обведенной кнопок при нажатии
--theme-colorDivider
Цвет разделяющей черты между кнопками в действии «Меню»
--theme-colorError
Используется в стилях отображения ошибок
--theme-colorListItemBackgroundFocused
Цвет фона кнопки в фокусе в действии «Меню»
--theme-colorListItemBackgroundPushed
Цвет фона кнопки при нажатии в действии «Меню»
--theme-textColorPrimary
Цвет текста по умолчанию
--theme-textColorSecondary
Вспомогательный цвет текста, используется в стилях текста описаний и комментариев

> Пользователь не может самостоятельно сменить значения предустановленных цветов.

Не нашли что искали?

Задать вопрос в техническую поддержку

Стандартные классы стилей в Mobile SMARTS

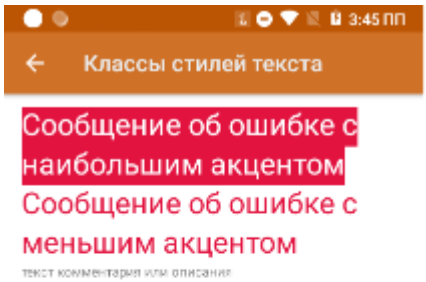
Последние изменения: 2024-03-26

Начиная с версии 3.3 платформы Mobile SMARTS в Android-клиент по умолчанию уже встроены стандартные классы стилей. Использование этих классов обязательно там, где это возможно. Не стоит писать свои аналогичные классы стилей.

Доступные стили текста:

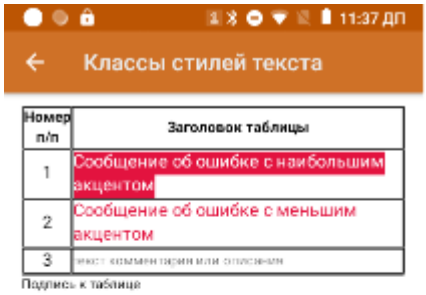
Класс
Описание
.__header_sm_text
текст для заголовков (скорее всего для заголовков колонок таблиц)
.__error_text
отображение уведомлений об ошибке
.__error_block
отображение уведомлений об ошибке в блоке с красным фоном
.__helper_text
текст для описаний, комментариев
.__caption_text
текст для подписей к таблицам, изображениям

Пример 1




```
<r class="__error_block" size="12">Сообщение об ошибке с наибольшим акцентом</r>
<r class="__error_text" size="12">Сообщение об ошибке с меньшим акцентом</r>
<r class="__helper_text" size="12">текст комментария или описания</r>
```

Пример 2



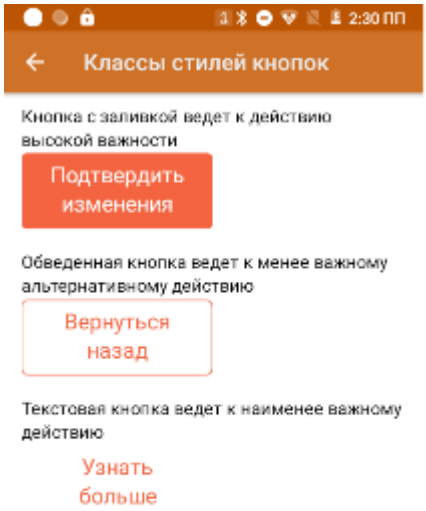

```
<div width="100%">
<table border="1dp" valign="middle">
<tr>
<td width="10%" align="center"><r class="__header_sm_text">Номер п/п</r></td>
<td align="center"><r class="__header_sm_text">Заголовок таблицы</r></td>
</tr>
<tr>
<td align="center">1</td>
<td><r class="__error_block">Сообщение об ошибке с наибольшим акцентом</r></td>
</tr>
<tr>
<td align="center">2</td>
<td><r class="__error_text">Сообщение об ошибке с меньшим акцентом</r></td>
</tr>
<tr>
<td align="center">3</td>
<td><r class="__helper_text">текст комментария или описания</r></td>
</tr>
</table>
<r class="__caption_text">Подпись к таблице</r>
</div>
```

Доступные стили кнопок:

Класс
Описание
.__contained_button
кнопки с заливкой используются для наиболее важных действий
.__outlined_button
обведенные кнопки используются для менее важных действий
.__text_button
текстовые кнопки используются для отображения наименее важных действий

.__menu_button
кнопка меню в действии «Меню»
.__exchange_button
кнопка обмена с сервером в главном меню
.__exchange_noconn_button
кнопка обмена в главном меню при отсутствии соединения с сервером

Пример

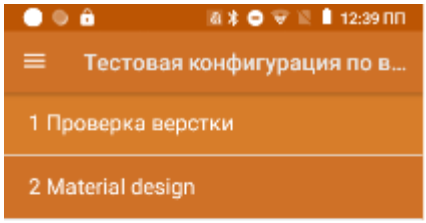


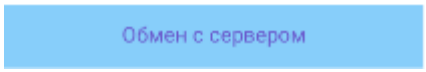
Кнопка с заливкой ведет к действию высокой важности
<button class="__contained_button" width="50%">Подтвердить изменения</button>

Обведенная кнопка ведет к менее важному альтернативному действию
<button class="__outlined_button" width="50%">Вернуться назад</button>

Текстовая кнопка ведет к наименее важному действию
<button class="__text_button" width="50%">Узнать больше</button>

Для изменения стиля кнопки обмена достаточно в файле global.css прописать свои значения атрибутам соответствующего класса, например:






```
.__exchange_button{
background-color: lightskyblue;
color: slateblue;
}
```

То же можно сделать и для других стандартных классов.

Кроме стандартных, можно использовать пользовательские классы стилей. Для их создания используется тег <style>. У каждого стиля есть имя и он хранит в себе параметры key-value.

```
<style name="MyStyle">
<item name="key">value</item>
 </style>
```

Не нашли что искали?