Android Html Class Tag Support

The Android *Html* class supports the following HTML tags and properties. The *HtmlCompat* class (AndroidX version 1.2.1) calls through to the framework version of *Html.fromHtml()*, so *HtmlCompat* supports the same tags as *Html* but ignores the *flags* argument for API versions below 24. The following is based upon an examination of the *Html* classes found in SDKs for API 23 and API 30.

Supported Tags

Tag	Block-level?[1]	Limited Style?[2]	Notes
<a>			Supports the *href* tag.
<b>			
<big>			
<blockquote>	Y		
<cite>			
<del>			Supported API 24+.
<dfn>			
	Y		
<font>			Supports the *color* and *face*[3] attributes.
<h1> … <h6>	Y		
<i>			
<img>			Supports the *src* tag with *Html.ImageGetter*.
	Y	Y	Supported API 24+.
<p>	Y	Y	
<s>			Supported API 24+.
<small>			
		Y	Supported API 24+.
<strike>			Supported API 24+.
<sub>			
<sup>			
<tt>			
<u>			
	Y		Supported API 24+.
Other tags			Supported *with Html.TagHandler*.

[1] Block-level elements support the *text-align* style property. The supported values for text-align are: *start*, *center* and *end*. (*justify* is not supported.)

[2] "Limited style" indicates that the tag supports the *color*, *background[-color]* and *text-decoration* properties. The only supported value *for text-decoration* is *line-through*. See below for details on color support.

[3] *face* can be any typeface name supported by the *TypefaceSpan* class.

Android Html Class Tag Support

Html.fromHtml() Flags

Values for the *flags* argument of *Html.fromHtml()* are:

FROM_HTML_SEPARATOR_LINE_BREAK_BLOCKQUOTE
FROM_HTML_SEPARATOR_LINE_BREAK_DIV
FROM_HTML_SEPARATOR_LINE_BREAK_HEADING
FROM_HTML_SEPARATOR_LINE_BREAK_LIST
FROM_HTML_SEPARATOR_LINE_BREAK_LIST_ITEM
FROM_HTML_SEPARATOR_LINE_BREAK_PARAGRAPH

Each of the preceding flags specifies that the HTML processor should add a single newline after each named block-level element. If the flag is not set then the processor adds two newlines which is the legacy behavior.

For example: Setting FROM_HTML_SEPARATOR_LINE_BREAK_HEADING will add one newline after a heading (<h1>, <h2>, etc.)

FROM_HTML_MODE_LEGACY: If this flag is set, then two newlines will be added after each block-level element. Setting this flag is the same as passing zero.

FROM_HTML_MODE_COMPACT: Use of this flag is the same as specifying all of the line break flags which will remove all extra newlines from block-level elements.

FROM_HTML_OPTION_USE_CSS_COLORS: For named colors, use the CSS numeric values instead of the values defined by the Android *Color* class.

For instance, if "darkgray" is specified as the color and this flag is set then the color value will be the CSS value for “darkgray” (0xFFA9A9A9) instead of the value for “darkgray” defined in the *Color* class (0xFF444444). If this flag is not set then the value will be the value from the *Color* class.

Android Html Class Tag Support

CSS Colors

Colors defined in the Color class:

```
aqua:           0xFF00FFFF
black:          BLACK
darkgray:       DKGRAY
blue:           BLUE
cyan:           CYAN
darkgrey:       DKGRAY
fuchsia:        0xFFFF00FF
gray:           GRAY
green:          GREEN
grey:           GRAY
lightgray:      LTGRAY
lightgrey:      LTGRAY
lime:           0xFF00FF00
magenta:        MAGENTA
maroon:         0xFF800000
navy:           0xFF000080
olive:          0xFF808000
purple:         0xFF800080
red:            RED
silver:         0xFFC0C0C0
teal:           0xFF008080
white:          WHITE
yellow:         YELLOW

(Constants used above)
BLACK:          0xFF000000
BLUE:           0xFF0000FF
CYAN:           0xFF00FFFF
DKGRAY:         0xFF444444
GRAY:           0xFF888888
GREEN:          0xFF00FF00
LTGRAY:         0xFFCCCCCC
MAGENTA:        0xFFFF00FF
RED:            0xFFFF0000
TRANSPARENT:    0
WHITE:          0xFFFFFFFF
YELLOW:         0xFFFFFF00
```

CSS colors that differ from Android Color class colors are:

```
darkgray:       0xFFA9A9A9
gray:           0xFF808080
lightgray:      0xFFD3D3D3
darkgrey:       0xFFA9A9A9
grey:           0xFF808080
lightgrey:      0xFFD3D3D3
green:          0xFF008000
```

These values are only available if the ***Html.FROM_HTML_OPTION_USE_CSS_COLORS*** flag is set.

Although "white" is defined as a valid color in the Color class, its value (0xFFFFFFFF) causes processing to ignore the color altogether. This is because the value returned for "white" by the Color class (0xFFFFFFFF) is interpreted as a "not found" condition (-1).

One work-around is to specify 0xFFFFFF for the color "white" and let the processing add the leading "FF".

Android Native Support for HTML Tags

For simple text formatting, Android natively supports a subset of HTML tags. These tags can be specified in a string resource and will be interpreted by the framework without resorting to an explicit HTML conversion. To retrieve a styled string from a string resource that is tagged with any of these tags, call either *"CharSequence getText (int id, CharSequence def)"* or *"CharSequence getText (int id)"*. HTML tags that are specified in a string resource that are not named below will not cause an error but will be ignored.

Simple Tags

- <b> Bold *(StyleSpan(Typeface.BOLD) is not supported.)*
- <big> 25% larger text *(RelativeSizeSpan (1.25f))*
- <i> Italic *(StyleSpan(Typeface.ITALIC) <cite> and <dfn> are not supported.)*
- Bullet points *(BulletSpan(10) is not supported. See note below.)*
- <marquee> Marquee *(See note below.)*
- <small> 20% smaller text *(RelativeSizeSpan(0.8f))*
- <strike> Strikethrough *(StrikethroughSpan() <s> and <del> are not supported.)*
- <sub> Subscript *(SubscriptSpan())*
- <sup> Superscript *(SuperscriptSpan())*
- <tt> Set a monospace font family *(TypefaceSpan("monospace"))*
- <u> Underline *(UnderlineSpan())*

Tags with Attributes

- Font: <font> *("int" is an integer value, For "color", see the notes.)*
    - bgcolor="color" Applies a BackgroundColorSpan to the text.
    - color=" color" Applies a ForegroundColorSpan to the text.
    - face="font_family" "font_family" can be any font acceptable to TypefaceSpan.
    - fgcolor="color" Applies a ForegroundColorSpan to the text.

  - height=" int" Applies an internally defined "Height" span that forces the text to be a specific height.
  - size=" int" Applies an AbsoluteSizeSpan to the text.
- Link: <a>
  - href="url"
- Annotation: <annotation>
  - key="value" Any number of key/value pairs where key can be any reasonable string. Each key/value pair creates a new Annotation span. Because of the way that Android encodes styles, "value" cannot contain a semicolon (";").

Tags that are not Supported

The following tags are mentioned as supported in the documentation but do not seem to be natively supported by Android.

- Line breaks:
 (Use "\n" instead.)
- CSS style:
- Paragraphs: <p dir="rtl | ltr" style="…">
- Division:

Notes on Native Android Support of HTML Tags

Colors supported are of the format #AARRGGBB (or #RRGGBB) or "@" followed by an Android resource color. For instance, "red" can be represented as "#FF0000", "#FFFF0000" or "@red" which is interpreted as "@android:color/red".

The unordered list tag, <ul>, is ignored and the <li> tag can stand on its own. The <li> tag does not introduce a line break as might be expected. The radius of the bullet created is 4px, the gap width is 10px and the bullet color is black (0xFF000000). These values cannot be changed.

For <marquee>, Android creates a "TextUtils.TruncateAt" span for the enclosed text. Other than to tag the text as "marquee" for some use later, it is unclear if Android does anything else with this tag.

For details on what is documented as supported, see
https://developer.android.com/guide/topics/resources/string-resource#StylingWithHTML

For supported tags, reference is made to the following methods in StringBlock.java from the AOSP:

applyStyles()
https://cs.android.com/android/platform/superproject/+/master:frameworks/base/core/java/android/content/res/StringBlock.java;drc=master;l=228

Android Html Class Tag Support

getColor()
https://cs.android.com/android/platform/superproject/+/master:frameworks/base/core/java/android/content/res/StringBlock.java;drc=master;l=387

String resource to test native support:

```
<string name="html_test"><b>bold</b><em> emphasis</em>\n
    <i>italic</i><cite> cite</cite><dfn> dfn</dfn>\n
    <big>big</big><small> small</small><tt> monospace</tt>\n
    <s>s</s><strike> strike</strike><del> del</del>\n
    <u>underline</u><sup> superscript</sup><sub> subscript</sub>\n
    <marquee>marquee</marquee>\n
    <li>Bullet1</li>\n<li>Bullet2</li>\n<li>Bullet3</li>\n
    <p style="fgcolor:#ff0000">Paragraph</p>\n
    <font bgcolor="#ff000000" color="@red" face="serif" height="30"
size="50">Font height</font>\n
    <a href="https://google.com">Link</a>\n
    <span style="color:#FF0000 background_color: #00FF00">span</span>
<div>div</div>
</string>
```

And the associated Kotlin code:

```
var s = getText(R.string.html_test)
var spans = (s as Spanned).getSpans(0, s.length, Any::class.java)
binding.textView1.setText(s)
binding.textView1.movementMethod = LinkMovementMethod.getInstance()
```

###