International Telecommunication Union

ITU-T

TELECOMMUNICATION
STANDARDIZATION SECTOR
OF ITU

H.763.1
(09/2010)

SERIES H: AUDIOVISUAL AND MULTIMEDIA SYSTEMS

IPTV multimedia services and applications for IPTV –
IPTV multimedia application frameworks

Cascading style sheets for IPTV services

Recommendation ITU-T H.763.1

ITU-T H-SERIES RECOMMENDATIONS

AUDIOVISUAL AND MULTIMEDIA SYSTEMS

Recommendation ITU-T H.763.1

Cascading style sheets for IPTV services

Summary

Recommendation ITU-T H.763.1 describes cascading style sheets (CSS) for IPTV. The CSS profile presented in this Recommendation provides the means for describing the presentation style of documents relative to IPTV terminal devices. This profile is based on CSS 1, a part of CSS 2 and a number of extended properties required for IPTV services. CSS is human readable and writable, and expresses style in common desktop publishing terminology.

History

Edition	Recommendation	Approval	Study Group
1.0	ITU-T H.763.1	2010-09-13	16

FOREWORD

The International Telecommunication Union (ITU) is the United Nations specialized agency in the field of telecommunications, information and communication technologies (ICTs). The ITU Telecommunication Standardization Sector (ITU-T) is a permanent organ of ITU. ITU-T is responsible for studying technical, operating and tariff questions and issuing Recommendations on them with a view to standardizing telecommunications on a worldwide basis.

The World Telecommunication Standardization Assembly (WTSA), which meets every four years, establishes the topics for study by the ITU-T study groups which, in turn, produce Recommendations on these topics.

The approval of ITU-T Recommendations is covered by the procedure laid down in WTSA Resolution 1.

In some areas of information technology which fall within ITU-T's purview, the necessary standards are prepared on a collaborative basis with ISO and IEC.

NOTE

In this Recommendation, the expression "Administration" is used for conciseness to indicate both a telecommunication administration and a recognized operating agency.

Compliance with this Recommendation is voluntary. However, the Recommendation may contain certain mandatory provisions (to ensure, e.g., interoperability or applicability) and compliance with the Recommendation is achieved when all of these mandatory provisions are met. The words "shall" or some other obligatory language such as "must" and the negative equivalents are used to express requirements. The use of such words does not suggest that compliance with the Recommendation is required of any party.

INTELLECTUAL PROPERTY RIGHTS

ITU draws attention to the possibility that the practice or implementation of this Recommendation may involve the use of a claimed Intellectual Property Right. ITU takes no position concerning the evidence, validity or applicability of claimed Intellectual Property Rights, whether asserted by ITU members or others outside of the Recommendation development process.

As of the date of approval of this Recommendation, ITU had not received notice of intellectual property, protected by patents, which may be required to implement this Recommendation. However, implementers are cautioned that this may not represent the latest information and are therefore strongly urged to consult the TSB patent database at http://www.itu.int/ITU-T/ipr/.

CONTENTS

Recommendation ITU-T H.763.1

Cascading style sheets for IPTV services

1   Scope

This Recommendation describes cascading style sheets to provide interoperability and harmonization among IPTV multimedia application frameworks. To provide global standard IPTV services, it is foreseeable that a combination of different standard multimedia application frameworks will be used. Therefore, this Recommendation describes cascading style sheets, as one of those standard multimedia application frameworks, to provide interoperable use of IPTV services.

2   References

The following ITU-T Recommendations and other references contain provisions which, through reference in this text, constitute provisions of this Recommendation. At the time of publication, the editions indicated were valid. All Recommendations and other references are subject to revision; users of this Recommendation are therefore encouraged to investigate the possibility of applying the most recent edition of the Recommendations and other references listed below. A list of the currently valid ITU-T Recommendations is regularly published. The reference to a document within this Recommendation does not give it, as a stand-alone document, the status of a Recommendation.

[ITU-T H.762]   Recommendation ITU-T H.762 (2009), *Lightweight interactive multimedia environment (LIME) for IPTV services*.

[CSS1]   World Wide Web Consortium (W3C) (1996), *Cascading Style Sheets, level 1*.

[CSS2]   World Wide Web Consortium (W3C) (1998), *Cascading Style Sheets, level 2 CSS2 Specification*.

3   Definitions

3.1   Terms defined elsewhere

None.

3.2   Terms defined in this Recommendation

None.

4   Abbreviations and acronyms

This Recommendation uses the following abbreviations and acronyms:

BML   Broadcasting Markup Language

CLUT   Colour Look-Up Table

CSS   Cascading Style Sheets

HTML   HyperText Markup Language

N/A   Not Applicable

SVG   Scalable Vector Graph

UA   User Agent

URI   Universal Resource Identifier

W3C	World Wide Web Consortium	
XML	Extensible Markup Language	

5   Introduction

CSS is a style sheet language specified by W3C that is used to describe the presentation (e.g., fonts, colours and spacing) of a document written in a markup language. CSS is human readable and writable, and expresses style in common desktop publishing terminology. CSS has been in use since October 1995 using the media type text/css without registration. A CSS can be either linked to a document through a URI or included in the document. CSS does not constitute by itself a multimedia framework, but is used with some markup languages such as the one defined in [ITU-T H.762].

This Recommendation gives the core functionalities of CSS for IPTV services.

6   Syntax and basic data types

The syntax for cascading style sheets is described in Appendix B of [CSS1]. The default style sheet describing the features of the profile specified in this Recommendation is given in Annex B herein.

7   Selectors

In CSS, selectors are patterns used in pattern matching rules that determine which style rules apply to elements in the document tree. Selectors may range from simple element names to rich contextual patterns. If all conditions in the pattern are true for a certain element, the selector matches the element.

Table 7-1 summarizes the selector syntax of this Recommendation:

Table 7-1 – The syntax of selectors

Pattern	Meaning	Described in clause
*	Matches any element.	Universal selector
E	Matches any E element (i.e., an element of type E).	Type selectors
E:active E:focus	Matches E during certain user actions.	Dynamic pseudo-classes
DIV.warning	Language specific (in HTML, the same as DIV[class~="warning"])	Class selectors
E#myid	Matches any E element with ID equal to "myid".	ID selectors

7.1   Universal selector

The universal selector, written "*", matches the name of any element type. It matches any single element in the document tree. If the universal selector is not the only component of a simple selector, the "*" may be omitted.

7.2   Type selectors

A type selector matches the name of a document language element type. A type selector matches every instance of the element type in the document tree.

7.3 Class selectors

The period (.) notation may be used for HTML as an alternative to the ~= notation when representing the class attribute. Thus, for HTML, div.value and div[class~=value] have the same meaning. The attribute value must immediately follow the "period" (.). UAs may apply selectors using the period (.) notation in XML documents if the UA has namespace specific knowledge that allows it to determine which attribute is the "class" attribute for the respective namespace.

7.4 ID selectors

An ID attribute can be used to uniquely identify its element, regardless of the document language, e.g., HTML or XML. In HTML all ID attributes are named "id"; XML applications may name ID attributes differently. In any document language, no two attributes of type ID can have the same value.

The ID attribute of a document language allows authors to assign an identifier to one element instance in the document tree. CSS ID selectors match an element instance based on its identifier. A CSS ID selector contains a "#" immediately followed by the ID value, which must be an identifier.

7.5 The dynamic pseudo-classes: ':active' and ':focus'

Interactive user agents sometimes change the rendering in response to user actions. CSS provides two pseudo-classes for common cases:

– The :active pseudo-class applies while an element is being activated by the user. For example, between the times the user presses the mouse button and releases it.

– The :focus pseudo-class applies while an element has the focus (accepts keyboard events or other forms of text input).

8 Properties

The following is the list of properties defined in this Recommendation.

@media	**padding-right**	**border-width**
left*	**height***	**display**
background-image	**font-size**	**letter-spacing**
margin	**padding-bottom**	**border-style**
top*	**z-index**	**visibility**
background-repeat	**font-weight**	**white-space**
padding-top	**padding-left**	**position**
width*	**line-height**	**overflow**
font-family	**text-align**	

NOTE – For property values with "*", the elements <input>, <object>, <div>, and <p> must have these property values. The elements <br>, <a>, <span> must not have these property values.

8.1 Assigning property values, cascading, and inheritance

8.1.1 Media types

One of the most important features of style sheets is that they specify how a document is to be presented on different media: on the screen, on paper, with a speech synthesizer, with a Braille device, etc.

Certain CSS properties are only designed for certain media (e.g., the 'page-break-before' property only applies to paged media). On occasion, however, style sheets for different media types may

share a property, but require different values for that property. For example, the 'font-size' property is useful both for screen and print media. The two media types are different enough to require different values for the common property; a document will typically need a larger font on a computer screen than on paper. Therefore, it is necessary to express that a style sheet, or a section of a style sheet, applies to certain media types.

8.1.1.1 The @media rule

An @media rule specifies the target media types (separated by commas) of a set of rules (delimited by curly braces). The @media construct allows style sheet rules for various media in the same style sheet:

```
@media print {
 body { font-size: 10pt }
 }
 @media screen {
 body { font-size: 13px }
 }
 @media screen, print {
 body { line-height: 1.2 }
 }
```

8.2 Box model

The CSS box model describes the rectangular boxes that are generated for elements in the document tree and laid out according to the visual formatting model.

8.2.1 Margin properties: 'margin'

Margin properties specify the width of the margin area of a box. The 'margin' shorthand property sets the margin for all four sides while the other margin properties only set their respective side. These properties apply to all elements, but vertical margins will not have any effect on non-replaced inline elements.

Table 8-1 – The 'margin' property

Value:	<margin-width>0
Initial:	See individual properties
Applies to:	All elements except elements with table display types other than table-caption, table and inline-table
Inherited:	No
Percentages:	Refer to width of containing block
Media:	Visual
Computed value:	See individual properties

The 'margin' property is a shorthand property for setting 'margin-top', 'margin-right', 'margin-bottom', and 'margin-left' at the same place in the style sheet.

If there is only one value, it applies to all sides. If there are two values, the top and bottom margins are set to the first value and the right and left margins are set to the second. If there are three values, the top is set to the first value, the left and right are set to the second, and the bottom is set to the third. If there are four values, they apply to the top, right, bottom, and left, respectively.

8.2.2 Padding properties: 'padding-top', 'padding-right', 'padding-bottom', 'padding-left'

The padding properties specify the width of the padding area of a box. The 'padding' shorthand property sets the padding for all four sides while the other padding properties only set their respective side.

The properties defined in this section refer to the padding-width value type, which may take one of the following values:

length

>   Specifies a fixed width.

percentage

>   The percentage is calculated with respect to the width of the generated box's containing block.

Unlike margin properties, values for padding values cannot be negative. Like margin properties, percentage values for padding properties refer to the width of the generated box's containing block.

Table 8-2 – The 'padding-top', 'padding-right', 'padding-bottom' and 'padding-left' property

Value:	<padding-width> (integer value in the unit of px, or 0.)
Initial:	0
Applies to:	All elements except table-row-group, table-header-group, table-footer-group, table-row, table-column-group and table-column. For div element, object element, form element, and img element, operated as a fixed value 0
Inherited:	No
Percentages:	Refer to width of containing block
Media:	Visual
Computed value:	The percentage as specified or the absolute length

These properties set the top, right, bottom, and left padding of a box.

8.2.3 Border properties

The border properties specify the width, colour, and style of the border area of a box. These properties apply to all elements.

8.2.3.1 Border width: 'border-width'

The border width properties specify the width of the border area.

The border's thickness has an explicit value. Explicit border widths cannot be negative.

Widths must be constant throughout a document.

Table 8-3 – The 'border-width' property

Value:	<border-width> (integer value in the unit of px, or 0.)
Initial:	0
Applies to:	All elements. All four sides are affected. For the object element and img element, operated as a fixed value 0.
Inherited:	No
Percentages:	N/A
Media:	Visual
Computed value:	See individual properties

8.2.3.2 The border-top-color, border-right-color, border-left-color, border-bottom-color properties

**Table 8-4 – The 'border-top-color, border-right-color,
border-left-color, border-bottom-color' properties**

Value:	<color>	transparent	inherit
Initial:	Transparent
Applies to:	
Inherited:	No
Percentages:	N/A
Media:	

8.2.3.3 Border style: 'border-style'

The border style properties specify the line style of a box's border (solid, double, dashed, etc.). The properties defined in this section refer to the <border-style> value type, which may take one of the following values:

none

No border; the computed border width is zero.

dotted

The border is a series of dots.

dashed

The border is a series of short line segments.

solid

The border is a single line segment.

All borders are drawn on top of the box's background.

Table 8-5 – The 'border-style' property

Value:	dotted | dashed | solid | none
Initial:	none
Applies to:	for img and object elements, the value is fixed as none in the operation
Inherited:	No
Percentages:	N/A
Media:	Visual
Computed value:	See individual properties

The 'border-style' property sets the style of the four borders. It can have from one to four values, and the values are set on the different sides as for 'border-width' above.

8.3 Visual formatting model

8.3.1 The 'display' property

Table 8-6 – The 'display' property

Value:	inline | block | none | inherit
Initial:	inline
Applies to:	div, body, input, textarea, img, object, form, and pre are fixed to block. br, span, and a are fixed to inline
Inherited:	No
Percentages:	N/A
Media:	
Computed value:	

The values of this property have the following meanings:

block

> This value causes an element to generate a block box.

inline

> This value causes an element to generate one or more inline boxes.

none

> This value causes an element to generate no boxes in the formatting structure (i.e., the element has no effect on layout). Descendant elements do not generate any boxes either; this behaviour cannot be overridden by setting the 'display' property on the descendants.

For the core profile, it is recommended that this property be constrained as follows:

This property has a different fixed value for each element as defined in the default style sheet. Elements div, body, input, textarea, img, object, form, and pre are fixed to block. Elements br, span, and a are fixed to inline. Exceptionally, for p element, the default is block. Further, the value is not inherited.

NOTE – meta, title, script, head, html, link, are fixed to none.

The string is clipped according to the value of overflow ("hidden"), when the string is larger than the area to display specified with width and height.

8.3.2 Choosing a positioning scheme: 'position' property

The 'position' property determines which of the positioning algorithms is used to calculate the position of a box.

Table 8-7 – The 'position' property

Value:	absolute | fixed
Initial:	absolute
Applies to:	p, div, textarea, input, img, object, pre, form, br, span, and all elements
Inherited:	No
Percentages:	N/A
Media:	Visual
Computed value:	As specified

The values of this property have the following meanings:

absolute

> The box's position (and possibly size) is specified with the 'top', 'right', 'bottom', and 'left' properties. These properties specify offsets with respect to the box's containing block. Absolutely positioned boxes are taken out of the normal flow. This means they have no impact on the layout of later siblings. Also, though absolutely positioned boxes have margins, they do not collapse with any other margins.

fixed

> The box's position is calculated according to the 'absolute' model, but in addition, the box is fixed with respect to some reference. As with the 'absolute' model, the box's margins do not collapse with any other margins. In the case of handheld, projection, screen, tty, and tv media types, the box is fixed with respect to the viewport and does not move when scrolled. In the case of the print media type, the box is rendered on every page, and is fixed with respect to the page box, even if the page is seen through a viewport (in the case of a print-preview, for example). For other media types, the presentation is undefined. Authors may wish to specify 'fixed' in a media-dependent way. For instance, an author may want a box to remain at the top of the viewport on the screen, but not at the top of each printed page. The two specifications may be separated by using an @media rule.

This property has a different fixed value for each element as defined in the default style sheet. Elements p, div, textarea, input, img, object, pre, and form are fixed to absolute. Elements br, span, and a are fixed to static. This property is not applied to elements other than the above.

8.3.3 Box offsets: 'top', 'left'

An element is said to be positioned if its 'position' property has a value other than 'static'. Positioned elements generate positioned boxes, laid out according to two properties: top, left.

Table 8-8 – The 'top' property

Value:	<length> (integer value in the unit of px, or 0)
Default:	0
Inherited:	No

The 'top' property specifies how far an absolutely positioned box's top margin edge is offset below the top edge of the box's containing block. For relatively positioned boxes, the offset is with respect to the top edges of the box itself (i.e., the box is given a position in the normal flow, and then offset from that position according to these properties).

NOTE 1 – For absolutely positioned elements whose containing block is based on a block-level element, this property is an offset from the padding edge of that element.

Table 8-9 – The 'left' property

Value:	\<length\> (integer value in the unit of px, or 0)
Default:	0
Inherited:	No

Like 'top', the 'left' property specifies how far a box's left margin edge is offset to the right of the left edge of the box's containing block. For relatively positioned boxes, the offset is with respect to the left edge of the box itself.

NOTE 2 – For absolutely positioned elements whose containing block is based on a block-level element, this property is an offset from the padding edge of that element.

For the two properties, *Value* has the following meaning:

length

> The offset is a fixed distance from the reference edge. Negative values are allowed.

8.3.4 Layered presentation

8.3.4.1 Specifying the stack level: the 'z-index' property

Table 8-10 – The 'z-index' property

Value:	auto

For a positioned box, the 'z-index' property specifies:

1) The stack level of the box in the current stacking context.

2) Whether the box establishes a local stacking context.

The value has the following meaning:

auto

> The stack level of the generated box in the current stacking context is the same as its parent's box. The box does not establish a new local stacking context.

8.4 Visual formatting model details

8.4.1 Content width: the 'width' property

Table 8-11 – The 'width' property

Value:	\<length\> (integer value in the unit of px, or 0)
Default:	0
Applies to:	All elements but non-replaced inline elements, table rows, and row groups
Inherited:	No

The 'width' property specifies the content width of boxes generated by block-level and replaced elements.

This property does not apply to non-replaced inline-level elements. The content width of non-replaced inline element's boxes is that of the rendered content within them (before any relative offset of children). Recall that inline boxes flow into line boxes. The width of line boxes is given by their containing block, but may be shorted by the presence of floats.

The width of a replaced element's box is intrinsic and may be scaled by the user agent if the value of this property is different than 'auto'.

Values have the following meaning:

length

> Specifies the width of the content area using a length unit.

Negative values for 'width' are illegal.

8.4.2   Content height: the 'height' property

Table 8-12 – The 'height' property

Value:	<length> (integer value in the unit of px, or 0)
Initial:	0
Applies to:	All elements but non-replaced inline elements, table columns, and column groups
Inherited:	No

This property specifies the content height of boxes generated by block-level, inline-block and replaced elements.

This property does not apply to non-replaced inline-level elements. See the section on computing heights and margins for non-replaced inline elements for the rules used instead.

Value has the following meaning:

length

> Specifies the height of the content area using a length value.

Negative values for 'height' are illegal.

8.5   Visual effects

8.5.1   Overflow: the 'overflow' property

Table 8-13 – The 'overflow' property

Value:	hidden
Initial:	hidden
Applies to:	Non-replaced block-level elements, table cells, and inline-block elements
Inherited:	No
Percentages:	N/A

This property specifies whether content of a block-level element is clipped when it overflows the element's box. It affects the clipping of all of the element's content except any descendant elements (and their respective content and descendants) whose containing block is the viewport or an ancestor of the element. Value has the following meaning:

hidden

> This value indicates that the content is clipped and that no scrolling user interface should be provided to view the content outside the clipping region.

8.5.2 Visibility: the 'visibility' property

Table 8-14 – The 'visibility' property

Value:	visible | hidden | inherit
Initial:	inherit
Applies to:	All elements
Inherited:	Yes
Percentages:	N/A

The 'visibility' property specifies whether the boxes generated by an element are rendered. Invisible boxes still affect layout (set the 'display' property to 'none' to suppress box generation altogether). Values have the following meanings:

visible

> The generated box is visible.

hidden

> The generated box is invisible (fully transparent, nothing is drawn), but still affects layout. Furthermore, descendents of the element will be visible if they have 'visibility: visible'.

This property may be used in conjunction with scripts to create dynamic effects. When the visibility is set to "hidden" in the parent element, and "visible" in the child element, only the parent is hidden. For the core profile, it is recommended that the value is fixed as "inherit" for span and a elements in the operation. Also, the value is fixed as "visible" for body element.

8.6 Colours and backgrounds

8.6.1 Colour units

8.6.1.1 Index-color

An index colour is specified using an index number <color-index> of the colour found in Annex A.

8.6.2 Colour: the 'color-index' property

This property specifies the foreground colour of an element with an index colour.

Table 8-15 – The 'color-index' property

Value:	<color-index> | inherit
Initial:	0
Applies to:	All elements
Inherited:	Yes
Percentages:	N/A
Media:	tv
Computed value:	N/A

8.6.3 The 'CLUT' property

This property specifies the resource of colour map data to be set to the CLUT using URI. If no clut property is specified explicitly, the receiver's default CLUT settings are retained. The value of this property is the URL function whose sole argument is the string for the URI referring to the location of the table.

Table 8-16 – The 'CLUT' property

Value:	url (<uri>)
Initial:	0
Applies to:	All elements
Inherited:	Yes
Percentages:	N/A
Media:	tv
Computed value:	N/A

8.6.4 Background properties: 'background-image' and 'background-repeat'

Table 8-17 – The 'background-image' property

Value:	<uri>	none
Initial:	none
Applies to:	All elements
Inherited:	No
Percentages:	N/A
Media:	Visual
Computed value:	

This property sets the background image of an element. When setting a background image, authors should also specify a background colour that will be used when the image is unavailable. When the image is available, it is rendered on top of the background colour. (Thus, the colour is visible in the transparent parts of the image). Note the encoding of the image specified by the <uri> is jpeg, and the upper left corner of the body element is the rendering origin.

Table 8-18 – The 'background-repeat' property

Value:	repeat
Initial:	repeat
Applies to:	All elements
Inherited:	No
Percentages:	N/A
Media:	Visual
Computed value:	As specified

If a background image is specified, this property specifies whether the image is repeated (tiled), and how. All tiling covers the content, padding and border areas of a box. The value has the following meaning:

repeat

> The image is repeated both horizontally and vertically.

8.6.5 The 'background-color-index' property

This property specifies the background colour of an element with a colour index.

Table 8-19 – The 'background-color-index' property

Value:	<color-index> | inherit
Initial:	0
Applies to:	All elements
Inherited:	Yes
Percentages:	N/A
Media:	tv
Computed value:	N/A

8.6.6 The 'border-top-color-index' property

This property specifies the border top colour of an element with a colour index.

Table 8-20 – The 'border-top-color-index' property

Value:	<color-index> | inherit
Initial:	0
Applies to:	All elements
Inherited:	No
Percentages:	N/A
Media:	tv
Computed value:	N/A

8.6.7 The 'border-right-color-index' property

This property specifies the border right colour of an element with a colour index.

Table 8-21 – The 'border-right-color-index' property

Value:	<color-index> | inherit
Initial:	0
Applies to:	All elements
Inherited:	No
Percentages:	N/A
Media:	tv
Computed value:	N/A

8.6.8 The 'border-bottom-color-index' property

This property specifies the border bottom colour of an element with a colour index.

Table 8-22 – The 'border-bottom-color-index' property

Value:	<color-index> | inherit
Initial:	0
Applies to:	All elements
Inherited:	No
Percentages:	N/A
Media:	tv
Computed value:	N/A

8.6.9 The 'border-left-color-index' property

This property specifies the border left colour of an element with a colour index.

Table 8-23 – The 'border-left-color-index' property

Value:	<color-index> | inherit
Initial:	0
Applies to:	All elements
Inherited:	No
Percentages:	N/A
Media:	tv
Computed value:	N/A

8.6.10 The 'grayscale-color-index' property

This property specifies the greyscale colour of an element with a colour index.

Table 8-24 – The 'grayscale-color-index' property

Value:	<color-index> <color-index> | inherit
Initial:	30 15
Applies to:	All elements
Inherited:	Yes
Percentages:	N/A
Media:	tv
Computed value:	N/A

8.7 Fonts

The exact uses of fonts with special considerations to different languages are for further study.

8.7.1 Font family: the 'font-family' property

Table 8-25 – The 'font-family' property

Value:	[[<family-name>	<generic-family>] [, <family-name>| <generic-family>]*]	inherit
Initial:	Depends on user agent
Applies to:	All elements
Inherited:	Yes
Percentages:	N/A
Media:	Visual
Computed value:	As specified

The value is a prioritized list of font family names and/or generic family names.

8.7.2 Font boldness: the 'font-weight' property

Table 8-26 – The 'font-weight' property

Value:	normal	bold	inherit
Initial:	normal
Applies to:	All elements
Inherited:	Yes
Percentages:	N/A
Media:	Visual
Computed value:	See text

The 'font-weight' property selects the weight of the font. The rendering result of "bold" depends on the receiver unit.

8.7.3 Font size: the 'font-size' property

Table 8-27 – The 'font-size' property

Value:	<size> (integer value in the unit of px)
Initial:	
Applies to:	All elements
Inherited:	Yes
Percentages:	Refer to parent element's font size
Media:	Visual
Computed value:	Absolute length

The font size corresponds to the em square, a concept used in typography. Note that certain glyphs may bleed outside their em squares.

Table 8-28 – The absolute-size mapping to HTML heading and absolute font-sizes

CSS absolute-size values	xx-small	x-small	small	medium	large	x-large	xx-large	–
HTML font sizes	1	–	2	3	4	5	6	7

Length and percentage values should not take the font size table into account when calculating the font size of the element. Negative values are not allowed.

8.8 Text

8.8.1 Alignment: the 'text-align' property

Table 8-29 – The 'text-align' property

Value:	left | right | center
Initial:	left
Applies to:	body, div, p, pre, br, span, a, form, input, textarea, object, img
Inherited:	Yes
Percentages:	N/A
Media:	Visual
Computed value:	The initial value or as specified

This property describes how inline content of a block is aligned. Values have the following meanings:

A block of text is a stack of line boxes. In the case of 'left', 'right' and 'center', this property specifies how the inline boxes within each line box align with respect to the line box's left and right sides; alignment is not with respect to the viewport. In the case of 'justify', this property specifies that the inline boxes are to be made flush with both sides of the block.

8.8.2 Letter and word spacing: the 'letter-spacing' property

Table 8-30 – The 'letter-spacing' property

Value:	normal | <length> | inherit
Initial:	normal
Applies to:	All elements
Inherited:	Yes
Percentages:	N/A
Media:	Visual
Computed value:	'normal' or absolute length

This property specifies spacing behaviour between text characters. Values have the following meanings:

normal

> The spacing is the normal spacing for the current font. This value allows the user agent to alter the space between characters in order to justify text.

length

> This value indicates inter-character space in addition to the default space between characters. Values may be negative, but there may be implementation-specific limits. User agents may not further increase or decrease the inter-character space in order to justify text.

8.8.3 Whitespace: the 'white-space' property

Table 8-31 – The 'white-space' property

Value:	normal
Initial:	normal
Applies to:	body, div, p, pre, br, span, a, form, input, textarea, object, img
Inherited:	Yes
Percentages:	N/A
Media:	Visual
Computed value:	As specified

This property declares how whitespace inside the element is handled. Value has the following meaning:

normal

> This value directs user agents to collapse sequences of whitespace, and break lines as necessary to fill line boxes.

8.8.4 The 'line-height' property

The CSS 'line-height property' is similar to leading in desktop publishing. It determines the "line height". This results in lines of text appearing closer together or further apart.

Also see the 'letter-spacing' property.

Table 8-32 – The 'line-height' property

Syntax:	line-height: <value>;
Possible values:	normal <length> – For example, 10px <number> – Multiplied with the current font size <percentage> – For example, 130% of current font size Inherit
Initial value:	normal
Applies to:	All elements
Inherited:	Yes
Media:	Visual

8.9 Extended properties for IPTV

8.9.1 The used-key-list

This property specifies the type of remote control keys to be accepted by the user agent that processes the style sheet.

Table 8-33 – The 'used-key-list' property

Value:	up	down	right	left	none
Initial:	<key-group>+	none
Applies to:	All elements
Inherited:	No

Examples of the keygroup are: basic, data-button, and numeric-tuning. As it is partially implementation dependent, the details of <key-group> are for further study.

8.9.2 Navigation attributes

There are five attributes: nav-index, nav-up, nav-down, nav-left and nav-right.

8.9.2.1 The 'nav-index' property

This property specifies an index for an element to which the focus is applied. Any element with this property set to none, is given no focus. The focus is initially applied to the element with this property set to 0.

Table 8-34 – The 'nav-index' property

Value:	<integer> | none
Initial:	none
Applies to:	All elements where the focus can be set
Inherited:	No
Percentages:	N/A
Media:	tv

8.9.2.2 The 'nav-up' property

This property specifies the value of the nav-index property describing an element to which the focus is applied when the up arrow key is pressed.

Table 8-35 – The 'nav-up' property

Value:	<integer> | none
Initial:	none
Applies to:	All elements where the focus can be set
Inherited:	No
Percentages:	N/A
Media:	tv

8.9.2.3 The 'nav-down' property

This property specifies the value of the nav-index property describing an element to which the focus is applied when the down arrow key is pressed.

Table 8-36 – The 'nav-down' property

Value:	<integer> | none
Initial:	none
Applies to:	All elements where the focus can be set
Inherited:	No
Percentages:	N/A
Media:	tv

8.9.2.4 The 'nav-right' property

This property specifies the value of the nav-index property describing an element to which the focus is applied when the right arrow key is pressed.

Table 8-37 – The 'nav-right' property

Value:	\<integer> | none
Initial:	none
Applies to:	All elements where the focus can be set
Inherited:	No
Percentages:	N/A
Media:	tv

8.9.2.5 The 'nav-left' property

This property specifies the value of the nav-index property describing an element to which the focus is applied when the left arrow key is pressed.

Table 8-38 – The 'nav-left' property

Value:	\<integer> | none
Initial:	none
Applies to:	All elements where the focus can be set
Inherited:	No
Percentages:	N/A
Media:	tv

8.9.3 The 'resolution' property

This property specifies the resolution of a text graphic plane.

Table 8-39 – The 'resolution' property

Value:	1920x1080 | 1280x720 | 960x540 | 720x480
Initial:	960x540
Applies to:	Body element
Inherited:	No
Percentages:	N/A
Media:	tv

8.9.4 The 'display-aspect-ratio' property

This property specifies the displayed aspect ratio of a text graphic plane.

Table 8-40 – The 'display-aspect-ratio' property

Value:	4v3 | 16v9
Initial:	16v9
Applies to:	Body element
Inherited:	No
Percentages:	N/A
Media:	tv

Annex A

Colour index for common colours

(This annex forms an integral part of this Recommendation)

This annex describes the common fixed colours to be used on an IPTV terminal device. There are 63 basic colours, including a half-transparent colour and a transparent colour.

The following policies are applied in deciding on the particular colours chosen:

1) First 16 colours are in accordance with the palette colours of 8-unit character codes. One colour is transparent.

2) The remaining are allocated equally in the colour space.

3) The Alpha values are also equally allocated.

4) According to the above policy, there would be 129 colours, so (R, G, B, Alpha) = (255, 255, 170, 128) is deleted to keep the table to 128 values.

5) Gamma correction is assumed.

   – allocation level of RGB: 64 colours with four R/G/B values = 0, 85, 170, 255;

   – allocation level of Alpha: Three Alpha values are used: 0, 128, 255.

In an implementation, the RGB values are expected to be converted to Y, Cb, Cr.

Table A.1 – Colour index table for common colours
(for four R/G/B values: 0, 85, 170, 255; three Alpha values: 0, 128, 255)

Index value	R	G	B	Alpha	Hexadecimal (RGB)	Name/Comments
0	0	0	0	255	#000000	Black
1	255	0	0	255	#ff0000	Red
2	0	255	0	255	#00ff00	Green
3	255	255	0	255	#ffff00	Yellow
4	0	0	255	255	#0000ff	Blue
5	255	0	255	255	#ff00ff	Magenta
6	0	255	255	255	#00ffff	Cyan
7	255	255	255	255	#ffffff	White
8	0	0	0	0	#000000	Transparent
9	170	0	0	255	#aa0000	Half Brightness Red
10	0	170	0	255	#00aa00	Half Brightness Green
11	170	170	0	255	#aaaa00	Half Brightness Yellow
12	0	0	170	255	#0000aa	Half Brightness Blue
13	170	0	170	255	#aa00aa	Half Brightness Magenta
14	0	170	170	255	#00aaaa	Half Brightness Cyan
15	170	170	170	255	#aaaaaa	Half Brightness White (Gray)
16	0	0	85	255	#000055	
17	0	85	0	255	#005500	
18	0	85	85	255	#005555	

**Table A.1 – Colour index table for common colours
(for four R/G/B values: 0, 85, 170, 255; three Alpha values: 0, 128, 255)**

Index value	R	G	B	Alpha	Hexadecimal (RGB)	Name/Comments
19	0	85	170	255	#0055aa	
20	0	85	255	255	#0055ff	
21	0	170	85	255	#00aa55	
22	0	170	255	255	#00aaff	
23	0	255	85	255	#00ff55	
24	0	255	170	255	#00ffaa	
25	85	0	0	255	#550000	
26	85	0	85	255	#550055	
27	85	0	170	255	#5500aa	
28	85	0	255	255	#5500ff	
29	85	85	0	255	#555500	
30	85	85	85	255	#555555	
31	85	85	170	255	#5555aa	
32	85	85	255	255	#5555ff	
33	85	170	0	255	#55aa00	
34	85	170	85	255	#55aa55	
35	85	170	170	255	#55aaaa	
36	85	170	255	255	#55aaff	
37	85	255	0	255	#55ff00	
38	85	255	85	255	#55ff55	
39	85	255	170	255	#55ffaa	
40	85	255	255	255	#55ffff	
41	170	0	85	255	#aa0055	
42	170	0	255	255	#aa00ff	
43	170	85	0	255	#aa5500	
44	170	85	85	255	#aa5555	
45	170	85	170	255	#aa55aa	
46	170	85	255	255	#aa55ff	
47	170	170	85	255	#aaaa55	
48	170	170	255	255	#aaaaff	
49	170	255	0	255	#aaff00	
50	170	255	85	255	#aaff55	
51	170	255	170	255	#aaffaa	
52	170	255	255	255	#aaffff	
53	255	0	85	255	#ff0055	
54	255	0	170	255	#ff00aa	
55	255	85	0	255	#ff5500	

**Table A.1 – Colour index table for common colours
(for four R/G/B values: 0, 85, 170, 255; three Alpha values: 0, 128, 255)**

Index value	R	G	B	Alpha	Hexadecimal (RGB)	Name/Comments
56	255	85	85	255	#ff5555	
57	255	85	170	255	#ff55aa	
58	255	85	255	255	#ff55ff	
59	255	170	0	255	#ffaa00	
60	255	170	85	255	#ffaa55	
61	255	170	170	255	#ffaaaa	
62	255	170	255	255	#ffaaff	
63	255	255	85	255	#ffff55	
64	255	255	170	255	#ffffaa	
65	0	0	0	128	#000000	Black
66	255	0	0	128	#ff0000	Red
67	0	255	0	128	#00ff00	Green
68	255	255	0	128	#ffff00	Yellow
69	0	0	255	128	#0000ff	Blue
70	255	0	255	128	#ff00ff	Magenta
71	0	255	255	128	#00ffff	Cyan
72	255	255	255	128	#ffffff	White
73	170	0	0	128	#aa0000	Half brightness Red
74	0	170	0	128	#00aa00	Half brightness Green
75	170	170	0	128	#aaaa00	Half brightness Yellow
76	0	0	170	128	#0000aa	Half brightness Blue
77	170	0	170	128	#aa00aa	Half brightness Magenta
78	0	170	170	128	#00aaaa	Half brightness Cyan
79	170	170	170	128	#aaaaaa	Half brightness White (Gray)
80	0	0	85	128	#000055	
81	0	85	0	128	#005500	
82	0	85	85	128	#005555	
83	0	85	170	128	#0055aa	
84	0	85	255	128	#0055ff	
85	0	170	85	128	#00aa55	
86	0	170	255	128	#00aaff	
87	0	255	85	128	#00ff55	
88	0	255	170	128	#00ffaa	
89	85	0	0	128	#550000	
90	85	255	85	128	#550055	
91	85	0	170	128	#5500aa	
92	85	0	255	128	#5500ff	

**Table A.1 – Colour index table for common colours
(for four R/G/B values: 0, 85, 170, 255; three Alpha values: 0, 128, 255)**

Index value	R	G	B	Alpha	Hexadecimal (RGB)	Name/Comments
93	85	85	0	128	#555500	
94	85	85	85	128	#555555	
95	85	85	170	128	#5555aa	
96	85	85	255	128	#5555ff	
97	85	170	0	128	#55aa00	
98	85	170	85	128	#55aa55	
99	85	170	170	128	#55aaaa	
100	85	170	255	128	#5546ff	
101	85	255	0	128	#55ff00	
102	85	255	85	128	#55ff55	
103	85	255	170	128	#55ffaa	
104	85	255	255	128	#55ffff	
105	170	0	85	128	#aa0055	
106	170	0	255	128	#aa00ff	
107	170	85	0	128	#aa5500	
108	170	85	85	128	#aa5555	
109	170	85	170	128	#aa55aa	
110	170	85	255	128	#aa55ff	
111	170	170	85	128	#aaaa55	
112	170	170	255	128	#aaaaff	
113	170	255	0	128	#aaff00	
114	170	255	85	128	#aaff55	
115	170	255	170	128	#aaffaa	
116	170	255	255	128	#aaffff	
117	255	0	85	128	#ff0055	
118	255	0	170	128	#ff00aa	
119	255	85	0	128	#ff5500	
120	255	85	85	128	#ff5555	
121	255	85	170	128	#ff55aa	
122	255	85	255	128	#ff55ff	
123	255	170	0	128	#ffaa00	
124	255	170	85	128	#ffaa55	
125	255	170	170	128	#ffaaaa	
126	255	170	255	128	#ffaaff	
127	255	255	85	128	#ffff55	

Annex B

Default style sheet

(This annex forms an integral part of this Recommendation)

```
media tv{
/* margin */
div, p, input, object {margin: 0 !important}
/* padding */
div, object {padding-top : 0 !important; padding-right : 0 !important;
padding-bottom : 0 !important; padding-left : 0 !important; }
/* border */
div, p, span, a, input {border-width : 0 }
object {border-width : 0 !important; border-style : none !important}
/* display */
meta, title, script, style, head, bml, bevent, beitem { display :
none !important }
body, div, p, object, input { display : block !important }
br, span, a { display : inline !important }
/* position */
p, div, object, input { position : absolute !important }
br, span, a { position : static !important }
/* top left width height */
p,div, input, object {top : 0; left : 0; width : 0; height : 0 }
/* z-index */
div, p, br, span, a, input, object, body { z-index : auto !important }
/* line-height */
br, span, a { line-height : inherit !important }
/* visibility */
body { visibility : visible !important }
span, a { visibility : inherit !important }
/* overflow */
p, div, input, object { overflow : hidden !important }
/* background-repeat */
body { background-repeat : repeat !important }
/* text */
p, input { font-family : "round gothic"; font-size : 24px }
span, a { font-family : inherit; font-size : inherit }
p, input { text-align : left }
/* letter-spacing */
span, a { letter-spacing : inherit !important }
/* white-space */
p, input { white-space : normal !important }
/* background-color-index */
body {background-color-index : 0}
div, p, span, a, input { background-color-index : 8 }
object { background-color-index : 0 !important}
/* grayscale-color-index */
p, input { grayscale-color-index : 30 15 }
}
```

SERIES OF ITU-T RECOMMENDATIONS

Series A	Organization of the work of ITU-T
Series D	General tariff principles
Series E	Overall network operation, telephone service, service operation and human factors
Series F	Non-telephone telecommunication services
Series G	Transmission systems and media, digital systems and networks
Series H	**Audiovisual and multimedia systems**
Series I	Integrated services digital network
Series J	Cable networks and transmission of television, sound programme and other multimedia signals
Series K	Protection against interference
Series L	Construction, installation and protection of cables and other elements of outside plant
Series M	Telecommunication management, including TMN and network maintenance
Series N	Maintenance: international sound programme and television transmission circuits
Series O	Specifications of measuring equipment
Series P	Terminals and subjective and objective assessment methods
Series Q	Switching and signalling
Series R	Telegraph transmission
Series S	Telegraph services terminal equipment
Series T	Terminals for telematic services
Series U	Telegraph switching
Series V	Data communication over the telephone network
Series X	Data networks, open system communications and security
Series Y	Global information infrastructure, Internet protocol aspects and next-generation networks
Series Z	Languages and general software aspects for telecommunication systems