Specification of htmlTool

1 Summary *ostream (publ) class htmlTool*

Declaration:
ostream (publ) class **htmlTool**;

Based on: iostream.h and cgi/Defs.H

Properties: htmlTool is a class designed for creating html-document easily. It can be used as any ostream.

2 Purpose

The htmlTool class purpose is to make it easier to create HTML-documents, which very often are necessary when you create CGI-programs. The class also contains some extensions for **any** ostream when you use the cgi-library, which are incredibly useful. (Take my word).

3 Description

3.1 Operations

3.1.1 Basic Methods

constructor **htmlTool** ()
Description: Initializes the class to use *stdout* for output.

constructor **htmlTool** (ostream &dest)
Description: Initializes the class to use *dest* for output.

destructor ~**htmlTool** (void)
Description: Closes the outputstream used

3.1.2 Methods which prints to the htmlTool

htmlTool& function **gotoURL** (**const** char *redirectUrl)

htmlTool& function **headerStatus** (int status, **const** char *statusMessage)

htmlTool& function **printHeader** (void)
Description: Should be used initially if you plan to write an HTML-document

htmlTool& function **printPlainHeader** (void)

htmlTool& function **headerContentType** (**const** char *mimeType)

htmlTool& function **html** (void)
Description: First time called, it prints an opening-tag, and second time called gives you a closing-tag. Most (if not all) methods which correspond to an HTML-element and takes no argument works this way. (e.g title just below)

htmlTool& function **title** (void)

htmlTool& function **title** (**const** char *content)
Description: As with most other methods, this method prints the content surrounded by it's HTML-element. Here it is TITLE.

htmlTool& function **head** (void)

htmlTool& function **body** (**const** char *bodyInfo=NULL)

htmlTool& function **foot** (void)

htmlTool& function **address** (void)

htmlTool& function **address** (**const** char *content)

htmlTool& function **comment** (void)
htmlTool& function **comment** (**const** char *content)
htmlTool& function **put** (char *text)
htmlTool& function **put** (**const** char *text)
htmlTool& function **put** (int i)
htmlTool& function **put** (long l)
htmlTool& function **put** (float f)
htmlTool& function **put** (double d)
htmlTool& function **ul** (void)
htmlTool& function **ol** (void)
htmlTool& function **li** (void)
htmlTool& function **dl** (void)
htmlTool& function **dd** (void)
htmlTool& function **dt** (void)
htmlTool& function **link** (**const** char *location, **const** char *description)
htmlTool& function **image** (**const** char *imageFile, **const** char *altText=NULL)
htmlTool& function **br** (void)
htmlTool& function **hr** (void)
htmlTool& function **hr** (int percent)
htmlTool& function **hr** (int percent, int size)
htmlTool& function **lf** (void)
htmlTool& function **p** (void)
htmlTool& function **p1** (void)
htmlTool& function **h1** (void)
htmlTool& function **h1** (**const** char *content)
htmlTool& function **h2** (void)
htmlTool& function **h2** (**const** char *content)
htmlTool& function **h3** (void)
htmlTool& function **h3** (**const** char *content)
htmlTool& function **h4** (void)
htmlTool& function **h4** (**const** char *content)
htmlTool& function **h5** (void)
htmlTool& function **h5** (**const** char *content)
htmlTool& function **h6** (void)
htmlTool& function **h6** (**const** char *content)
htmlTool& function **b** (void)
htmlTool& function **b** (**const** char *content)
htmlTool& function **i** (void)
htmlTool& function **i** (**const** char *content)
htmlTool& function **fixed** (void)
htmlTool& function **fixed** (**const** char *content)
htmlTool& function **underline** (void)

htmlTool& function **underline** (**const** char *content)

htmlTool& function **define** (void)

htmlTool& function **define** (**const** char *content)

htmlTool& function **em** (void)

htmlTool& function **em** (**const** char *content)

htmlTool& function **cite** (void)

htmlTool& function **cite** (**const** char *content)

htmlTool& function **code** (void)

htmlTool& function **code** (**const** char *content)

htmlTool& function **kbd** (void)

htmlTool& function **kbd** (**const** char *content)

htmlTool& function **samp** (void)

htmlTool& function **samp** (**const** char *content)

htmlTool& function **var** (void)

htmlTool& function **var** (**const** char *content)

htmlTool& function **strong** (void)

htmlTool& function **strong** (**const** char *content)

htmlTool& function **strike** (void)

htmlTool& function **strike** (**const** char *content)

htmlTool& function **block** (void)

htmlTool& function **block** (**const** char *content)

htmlTool& function **pre** (void)

htmlTool& function **pre** (**const** char *content)

htmlTool& function **center** (void)

htmlTool& function **center** (**const** char *content)

htmlTool& function **big** (void)

htmlTool& function **big** (**const** char *content)

htmlTool& function **small** (void)

htmlTool& function **small** (**const** char *content)

htmlTool& function **sub** (void)

htmlTool& function **sub** (**const** char *content)

htmlTool& function **sup** (void)

htmlTool& function **sup** (**const** char *content)

htmlTool& function **subscript** (void)

htmlTool& function **subscript** (**const** char *content)

htmlTool& function **superscript** (void)

htmlTool& function **superscript** (**const** char *content)

htmlTool& function **font** (char action, int =1)

ostream & function **br** (ostream &outs)

Description: A genial construction. Lets you use **br** where you would use **endl** in an ostream. Producing a lineshift-HTML-tag and flushing it. The same goes for those below. Check also *test.cpp* for several examples.

ostream & function **hr** (ostream &outs)

ostream & function **p** (ostream &outs)

ostream & function **li** (ostream &outs)

ostream & function **dd** (ostream &outs)

3.1.3 Protected Methods

ostream & function **outputtag** (int &type, **const** char *value)
Description: For internal use

htmlTool& function **outputtag** (**const** char *value, **const** char *type)
Description: For internal use

procedure **initvars** (void)
Description: For internal use

Document created with SDS v0.20