Contents

Revision History

Date	Version	Description	Author(s)
04.12.2017	1.0	Initial Version	Emily WUU, Deniz Sarici, Anthony Kalbermatten, Andrea Meier
17.12	1.1	Anchor, List, External Links and Code Generator Visitor Update	Anthony Kalbermatten
17.12	1.2	Text, Snippet, Table	Deniz
18.12	1.3	Image and Paragraph Test Update	Emily WUU

Public API Overview

1.1 Public classes to be tested

html_text

html_image

html_snippet

html_external_link

html_ordered_list

html_unordered_list

html_table

html_paragraph

html_anchor_manager

html_anchor

html_title

html_code_generator_visitor

1.2 Public Routines

html_text.make()

html_image.make()

html_snippet.make()

html_external_link.make()

html_unordered_list.make()

html_unordered_list.add_row()

html_table.make()

html_table.add_row()

html_paragraph.make()

html_paragraph.new_element()

html_anchor_manager.make()

html_anchor_manager.create_anchor()

code_generator_visitor.visit_anchor

code_generator_visitor.visit_ext_link

code_generator_visitor.visit_title

code_generator_visitor.visit_image

code_generator_visitor.visit_int_link

code_generator_visitor.visit_ordered_list

code_generator_visitor.visit_paragraph

code_generator_visitor.visit_snippet

code_generator_visitor.visit_table

code_generator_visitor.visit_text

code_generator_visitor.visit_unordered_list

2 Test Suite Description

2.1 HTML_Text Tests

2.1.1 <HTMLText Test 1>

Test, that an HTMLText object is created, when provided a correct string.

- **Routine(s) or other functionality under test:** make()
- **Set-up if any:** text1: TEXT
- **Tear-down if any:** None
- **Test data:** text1.make("I am a string")
- **Any other resources:** None
- **Oracle:**

An instance of a text is created with content "I am a string". All style attributes are set to false.

2.1.2 <HTMLText Test 2>

Test, that style attributes (bold, italic and underlined) are set to True. Testing each style feature. The text instance is assumed to be created with a valid string input.

- **Routine(s) or other functionality under test:** bold, italic, underlined
- **Set-up if any:** text1: TEXT
  text1.make("I am a string")
- **Tear-down if any:** None
- **Test data:** Set text1 bold, italic and underlined
- **Any other resources:** None
- **Oracle:**

An instance of text is created with content "I am a string". The style attributes is_bold, is_italic and is_underlined are set to True.

2.1.3 <HTMLText Test 3>

Test that text set to be bold.

- **Routine(s) or other functionality under test:** bold, italic, underlined
- **Set-up if any:** text1: TEXT
  text1.make("I am a string")
- **Tear-down if any:** None
- **Test data** Set text1 to bold
- **Any other resources:** None
- **Oracle:**

An instance of text is created with content "I am a string". Only the style attribute is_bold is set to True, the others are set to False.

2.1.4 <HTMLText Test 4>

Test that text set to be bold.

- **Routine(s) or other functionality under test:** bold, italic, underlined
- **Set-up if any:** text1: TEXT
  text1.make("I am a string")
- **Tear-down if any:** None
- **Test data** Set text1 to italic
- **Any other resources:** None
- **Oracle:**

An instance of text is created with content "I am a string". Only the style attribute is_italic is set to True, the others are set to False.

2.1.5 <HTMLText Test 5>

Test that text set to be bold.

- **Routine(s) or other functionality under test:** bold, italic, underlined
- **Set-up if any:** text1: TEXT
  text1.make("I am a string")
- **Tear-down if any:** None
- **Test data** Set text1 to underlined
- **Any other resources:** None
- **Oracle:**

An instance of text is created with content "I am a string". Only the style attribute is_underlined is set to True, the others are set to False.

2.1.6 <HTMLText Test 6>

Test that text remains bold if bold feature was called again.

- **Routine(s) or other functionality under test:** bold
- **Set-up if any:** text1: TEXT
  text1.make("I am a string")
  text1.bold
- **Tear-down if any:** None
- **Test data:** Set text1 to bold (again)
- **Any other resources:** None
- **Oracle:**

An instance of text is created with content “I am a string”. The style attribute is_bold is remains set to True. The other style attributes remain unchanged as well.

2.1.7 <HTMLText Test 7>

Test that an exception is caught if the precondition is violated by creating an instance with an empty string.

- **Routine(s) or other functionality under test:** make
- **Set-up if any:** text1: TEXT
- **Tear-down if any:** None
- **Test data:** text is created with an empty string
- **Any other resources:** None
- **Oracle:**

Precondition is violated and the exception caught by the test.

2.2 HTMLTitle Tests

2.2.1 <Title Test 1>

Test, that an Title object is created, when provided a correct input.

- **Routine(s) or other functionality under test:** make()
- **Set-up if any:** title1: TITLE
- **Tear-down if any:** None
- **Test data:** String1: “I am a title”
  Integer1: random number from 1 to 6 (e.g. 5)
- **Any other resources:** None
- **Oracle:**

Creates an title object “title1” in which the string “I am a header” is attached to the variable text and the size integer1 is attached to the variable size.

2.2.2 <Title Test 2>

Test, that no Title object is created, when the second input is not within the scope of integers 1-6.

- **Routine(s) or other functionality under test:** make()
- **Set-up if any:** title2: TITLE
- **Tear-down if any:** None
- **Test data:** String1: “I am a title”
  Integer1: random integer except the integers 1-6, eg.7.
- **Any other resources:** None

Oracle:

Precondition is violated and the exception caught by the test.

2.3 HTMLImage Tests

2.3.1 <HTMLImage Test 1>

Test, that an HTMLImage is created.

- **Routine(s) or other functionality under test:** make()
- **Set-up if any:** image1: IMAGE

- **Tear-down if any:** None
- **Test data:** src: “test_image.jpg”
  alt: “alternative text”
- **Any other resources:** None
- **Oracle:**
  An object “image1” of type HTMLImage is created and the variable src has the value “test_image1” and the variable althas the value “alternative text”

2.3.2 <HTMLImage Test 2>

Test, that an HTMLImage not created if the input of source of image is empty

- **Routine(s) or other functionality under test:** make()
- **Set-up if any:** image1: IMAGE
- **Tear-down if any:** None
- **Test data:** src: “”
  alt: “alternative text”
- **Any other resources:** None
- **Oracle:**
  Precondition is violated and the exception caught by the test.

2.3.3 <HTMLImage Test 3>

Test, that an HTMLImage not created if the input of alt of image is empty

- **Routine(s) or other functionality under test:** make()
- **Set-up if any:** image1: IMAGE
- **Tear-down if any:** None
- **Test data:** src: “test_image.jpg”
  alt: “”
- **Any other resources:** None
- **Oracle:**
  Precondition is violated and the exception caught by the test.

2.4 HTML_Snippets Tests

2.4.1 <HTMLSnippets Test 1>

Test, that a HTMLSnippet is created

- **Routine(s) or other functionality under test:** make()
- **Set-up if any:** Snippet1: SNIPPET
- **Tear-down if any:** None
- **Test data:** String1: “I am a string”
- **Any other resources:** None
- **Oracle:**
  An HTML is created within String is stored

2.4.2 <HTMLSnippets Test 2>

Test that an exception is caught if the precondition is violated by creating an instance with an empty string.

- **Routine(s) or other functionality under test:** make
- **Set-up if any:** snippet1: SNIPPET
- **Tear-down if any:** None
- **Test data:** snippet is created with an empty string

- **Any other resources:** None
- **Oracle:**

Precondition is violated and the exception caught by the test.

2.5 HTMLExternal_Link Tests

2.5.1 <HTMLExternal_Link Test 1>

Test, that an HTMLExternal_Link object is created.

- **Routine(s) or other functionality under test:** make()
- **Set-up if any:** Text1: HTMLText
- **Tear-down if any:** None
- **Test data:** url: “https://www.google.ch/”
text: Text1
- **Any other resources:** None
- **Oracle:**

  Creates an an htmlexternal_link object within Text1 and url are stored.

2.5.2 <HTMLExternal_Link Test 2>

Test, that no HTMLExternal_Link object is created when url is no string.

- **Routine(s) or other functionality under test:** make()
- **Set-up if any:** Text1: HTMLText
- **Tear-down if any:** None
- **Test data:** url: any data type except of string
text: Text1
- **Any other resources:** None
- **Oracle:**

  Program reports a failure in the input.

2.5.3 <HTMLExternal_Link Test 3>

Test, that no HTMLExternal_Link object is created when text is not of type HTMLText.

- **Routine(s) or other functionality under test:** make()
- **Set-up if any:** Object1: Any object except of HTMLTextObjects
- **Tear-down if any:** None
- **Test data:** url: “https://www.google.ch/”
text: Object1
- **Any other resources:** None
- **Oracle:**

  Program reports a failure in the input.

2.5.4 <HTMLExternal_Link Test 4>

Test, that no HTMLExternal_Link object is created when text object is not existing.

- **Routine(s) or other functionality under test:** make()
- **Set-up if any:** Text1: HTMLText
- **Tear-down if any:** None
- **Test data:** url: “https://www.google.ch/”
text: Text2
- **Any other resources:** None
- **Oracle:**

  Program reports a failure in the input.

2.6 Unordered-List Tests

2.6.1 <Unordered_List Test 1>

Test, that an unordered_list object is created

- **Routine(s) or other functionality under test:** make
- **Set-up if any:** list: HTML_UNORDERED_LIST
- **Tear-down if any:** None
- **Test data:** None
- **Any other resources:** None
- **Oracle:**
  An ordered_list object is created with an empty row list.

2.6.2 <Unordered_List Test 2>

Test, that content can be added to an unordered_list object

- **Routine(s) or other functionality under test:** add_row()
- **Set-up if any:** list: HTML_UNORDERED_LIST
- **Tear-down if any:** None
- **Test data:** Content1: any class that is an HTMLComponent
  Object: UList1
- **Any other resources:** None
- **Oracle:**
  Content1 is added to UList1.

2.7 HTML_Table Tests

2.7.1 <Table Test 1>

Test, that a table object is created with specified number of columns.

- **Routine(s) or other functionality under test:** make()
- **Set-up if any:** table1: TABLE
- **Tear-down if any:** None
- **Test data:** column1: integer >= 1
- **Any other resources:** None
- **Oracle:**
  A table object is created with column1 columns. The list of rows is empty.

2.7.2 <Table Test 2>

Test, that the precondition is violated, when provided a wrong input for column

- **Routine(s) or other functionality under test:** make()
- **Set-up if any:** table1: TABLE
- **Tear-down if any:** None
- **Test data:** column1: any data types except integer or 0
- **Any other resources:** None
- **Oracle:**
  Program reports a failure in the input.

2.7.3 <Table Test 3>

Test, that a row with a single text instance is added to a table object. Column of table is set to 1.

- **Routine(s) or other functionality under test:** add_row
- **Set-up if any:** table1: TABLE
  row1: LINKED_LIST
  text1: TEXT
- **Tear-down if any:** None
- **Test data:** table1 created with column=1
  row1 contains text1, text1 has content “I am a string”
- **Any other resources:** None
- **Oracle:**

Table with one row and one cell containing an instance of text. Content of text is “I am a string”.

2.7.4 <Table Test 4>

Test, that a row cannot be added to a table object if the list contains a wrong number of objects

- **Routine(s) or other functionality under test:** add_row
- **Set-up if any:** table1: TABLE with 1 column
  text1: TEXT
- **Tear-down if any:** None
- **Test data:** row1: LINKED_LIST with 2 entries of text1
- **Any other resources:** None
- **Oracle:**

Precondition violation raised.

2.8 Paragraph Tests

2.8.1 <Paragraph Test 1>

Test, that an paragraph object is created

- **Routine(s) or other functionality under test:** make()
- **Set-up if any:** paragraph1: PARAGRAPH
- **Tear-down if any:** None
- **Test data:** None
- **Any other resources:** None
- **Oracle:**

A paragraph(paragraph1) gets created.

2.8.2 <Paragraph Test 2>

Test, that an object can be added to an paragraph

- **Routine(s) or other functionality under test:** new_element()
- **Set-up if any:** dummy_object1: HTML_TEXT
- **Tear-down if any:** None
- **Test data:** dummy_object1
- **Any other resources:** None
- **Oracle:**

The element “dummy_object1” is stored at the first place in the array “word_arr” in the paragraph object “paragraph1”.

2.8.3 <Paragraph Test 3>

Test, that a second object can be added to an paragraph

- **Routine(s) or other functionality under test:** new_element()
- **Set-up if any:** dummy_object2: HTML_TEXT
- **Tear-down if any:** None

- **Test data:** dummy_object2
- **Any other resources:** None
- **Oracle:**
  The element “dummy_object2” is stored at the second place in the array “word_arr” in the paragraph object “paragraph1” while the first element “dummy_object1” is still located at the first position.

2.9 HTML_Anchor and HTML_anchor_manager Tests

2.9.1 <Anchor Test 1>

Test, that an anchor_manager is created

- **Routine(s) or other functionality under test:** HTML_ANCHOR_MANAGER.make()
- **Set-up if any:** create manager.make (Type ANCHOR MANAGER)
  create cgv.make (Type HTML_CODE_GENERATOR_VISITOR)
  create exception_asserter
- **Tear-down if any:** None
- **Test data:** manager.make()
- **Any other resources:** None
- **Oracle:**
  An anchor_manager object “manager” gets created with an id = 1

2.9.2 <Anchor Test 2>

Test, that an anchor is set/anchor object is created

- **Routine(s) or other functionality under test:** create_anchor
  HTML_ANCHOR.make
- **Set-up if any:** like in Anchor Test 1
  anchor1: HTML_ANCHOR
- **Tear-down if any:** None
- **Test data:** anchor1 = manager.create_anchor()
- **Any other resources:** None
- **Oracle:**
  An anchor object “anchor1” gets created with the id = 1

2.9.3 <Anchor Test 3>

Test, that a second anchor can be set. and its id is another then of the first anchor

- **Routine(s) or other functionality under test:** create_anchor
  HTML_ANCHOR.make()
- **Set-up if any:** like in Anchor Test 2
  anchor2: HTML_ANCHOR
- **Tear-down if any:** None
- **Test data:** anchor2 = manager.create_anchor()
- **Any other resources:** None
- **Oracle:**
  A second anchor object gets created with an id = 2

2.9.4 <Anchor Test 4>

Test, that a anchor link to the first anchor can be created

- **Routine(s) or other functionality under test:** add_link
- **Set-up if any:** like in 2
  link1 : HTML_INTERNAL_LINK

- **Tear-down if any:** None
- **Test data:** link1 = anchor1.add_link(“click me to get to an other position”)
- **Any other resources:** None
- **Oracle:**

  Creates a link object with the url and an id=1 and the text “click me to get to another position”

2.9.5 <Anchor Test 5>

Test, that two internal links from the same anchor have the same id

- **Routine(s) or other functionality under test:** add_link
  - **Set-up if any:** like in Anchor Test 4
link2: HTML_INTERNAL_LINK
  - **Tear-down if any:** None
  - **Test data:** link1.id and link2.id
  - **Any other resources:** None
  - **Oracle:**

    Creates two links with the id “00000001”

2.9.6 <Anchor Test 6>

Test, that precondition holds for empty source

- **Routine(s) or other functionality under test:** precondition source_is_not_empty
  - **Set-up if any:** like in 1
link1: HTML_INTERNAL_LINK
  - **Tear-down if any:** None
  - **Test data:** link1.make(“”, “TEXT”)
  - **Any other resources:** None
  - **Oracle:**

    Program states that a precondition violation occurred

2.9.7 <Anchor Test 7>

Test, that precondition holds for empty text

- **Routine(s) or other functionality under test:** precondition text_is_not_empty
  - **Set-up if any:** like in 1
link1: HTML_INTERNAL_LINK
  - **Tear-down if any:** None
  - **Test data:** link1.make(“TEXT”, “”)
  - **Any other resources:** None
  - **Oracle:**

    Program states that a precondition violation occurred

2.10 HTML_code_generator_visitor Tests

2.10.1 <Visitor Test 1>

Test, that correct HTML code of the anchor is generated.

- **Routine(s) or other functionality under test:** visit_anchor
- **Set-up if any:** Anchor object as described in 2.10.2
- **Tear-down if any:** None
- **Test data:** anchor1

cgv.visit_anchor(anchor1)

- **Any other resources:** None
- **Oracle:**
  The visitor creates a string of the HTML representation which is stored in variable end_string and it will look like this: "%T<a name=%"00000001%"></a>%N"

2.10.2 <Visitor Test 2>

Test, that correct HTML code of the internal link is generated.

- **Routine(s) or other functionality under test:** visit_int_link
- **Set-up if any:** Anchor object anchor1 as described in 2.10.2
- **Tear-down if any:** None
- **Test data:** internal :=anchor1.add_link(“to top”)
  cgv.visit_int_link(internal)
- **Any other resources:** None
- **Oracle:**
  The visitor creates a string of the HTML representation which is stored in variable end_string and it will look like this: "<%T<a href=%"#00000001%">to top</a>%N”

2.10.3 <Visitor Test 3>

Test, that correct HTML code of the external link is generated.

- **Routine(s) or other functionality under test:** visit_ext_link()
- **Set-up if any:** External link object as described in 2.5.1
- **Tear-down if any:** None
- **Test data:** link
  cgv.visit_ext_link(link)
- **Any other resources:** None
- **Oracle:**
  The visitor creates a string of the HTML representation which is stored in variable end_string and it will look like this: "%T<a href=%"www.google.com%">Link to Google</a>%N"

2.10.4 <Visitor Test 4>

Test, that correct HTML code of title is generated.

- **Routine(s) or other functionality under test:** visit_title
- **Set-up if any:** title:= TITLE
  title.make(“I am a title”,4)
- **Tear-down if any:** None
- **Test data:** title
  cgv.visit_title(title)
- **Any other resources:** None
- **Oracle:**
  The visitor creates a string of the HTML representation which is stored in variable end_string and it will look like this: "%T<h4>%"I am a title%"</h4>%N"

2.10.5 <Visitor Test 5>

Test, that correct HTML code of an image is generated.

- **Routine(s) or other functionality under test:** visit_image
- **Set-up if any:** image:= IMAGE
  image.make“testimage.jpg”,”testimage”)
- **Tear-down if any:** None
- **Test data:** image
  cgv.visit_image(image)

- **Any other resources:** None
- **Oracle:**
  The visitor creates a string of the HTML representation which is stored in variable end_string and it will look like this: <%T<img src=%"testimage.jpg%" alt=%"testimage%">%N"

2.10.6 <Visitor Test 6>

Test, that correct HTML code of an ordered_list is generated.

- **Routine(s) or other functionality under test:** visit_ordered_list
- **Set-up if any:** olist:= HTML_ORDERED_LIST
  otext:=HTML_TEXT
  otext.make("I am dummy text")
  olist.add_row(otext)
  olist.add_row(otext)
- **Tear-down if any:** None
- **Test data:** olist
  cgv.visit_ordered_list(olist)
- **Any other resources:** None
- **Oracle:**
  The visitor creates a string of the HTML representation which is stored in variable end_string and it will look like this:%T<ol>%N%T%T<li>I am dummy text</li>%N%T%T<li>I am dummy text</li>%N%T</ol>%N"

2.10.7 <Visitor Test 7>

Test, that correct HTML code of an oedered_list is generated.

- **Routine(s) or other functionality under test:** visit_unordered_list
- **Set-up if any:** ulist:= HTML_UNORDERED_LIST
  utext:=HTML_TEXT
  utext.make("I am dummy text")
  ulist.add_row(utext)
  ulist.add_row(utext)
- **Tear-down if any:** None
- **Test data:** ulist
  cgv.visit_unordered_list(ulist)
- **Any other resources:** None
- **Oracle:**
  The visitor creates a string of the HTML representation which is stored in variable end_string and it will look like this:"%T<ul>%N%T%T<li>I am dummy text</li>%N%T%T<li>I am dummy text</li>%N%T</ul>%N"

2.10.8 <Visitor Test 8>

Test, that correct HTML code of a paragraph is generated.

- **Routine(s) or other functionality under test:** visit_paragraph
- **Set-up if any:** para:= HTML_PARAGRAPH
  ptext:=HTML_TEXT
  ptext.make("I am dummy text")
  para.make
  para.new_element(ptext)
- **Tear-down if any:** None
- **Test data:** para
  cgv.visit_paragraph(para)
- **Any other resources:** None
- **Oracle:**

The visitor creates a string of the HTML representation which is stored in variable end_string and it will look like this:”T<p>I am dummy text</p>%N”

2.10.9 <Visitor Test 9>

Test, that correct HTML code of a snippet is generated.

- **Routine(s) or other functionality under test:** visit_snippet
- **Set-up if any:** snippet:=HTML_SNIPPET
  snippet.make(“<p>I am a snippet</p>”)
- **Tear-down if any:** None
- **Test data:** snippet
  cgv.visit_snippet(snippet)
- **Any other resources:** None
- **Oracle:**
  The visitor creates a string of the HTML representation which is stored in variable end_string and it will look like this:”%T<p>I am a snippet</p>%N”

2.10.10 <Visitor Test 10>

Test, that correct HTML code of a table is generated.

- **Routine(s) or other functionality under test:** visit_table
- **Set-up if any:** table:=HTML_TABLE
  ttext:=HTML_TEXT
  tlist:= LINKED_LIST[COMPONENT]
  ttext:=html.text(“I am dummy text”)
  table:=html.table(2)
  table.add_row(tlist)
  table.add_row(tlist)
- **Tear-down if any:** None
- **Test data:** table
  cgv.visit_table(table)
- **Any other resources:** None
- **Oracle:**
  The visitor creates a string of the HTML representation which is stored in variable end_string and it will look like this:”%T<table>%N%T%T<tr>%N%T%T%T<th>I am dummy text</th>%N%T%T%T<th>I am dummy text</th>%N%T%T</tr>%N%T%T<tr>%N%T%T%T<th>I am dummy text</th>%N%T%T%T<th>I am dummy text</th>%N%T%T</tr>%N%T</table>%N”

2.10.11 <Visitor Test 11>

Test, that correct HTML code of a text is generated.

- **Routine(s) or other functionality under test:** visit_text
- **Set-up if any:** text:=HTML_TEXT
  text.make(“I am dummy text”)
  text.bold
  text.underline
  text.cursiv
- **Tear-down if any:** None
- **Test data:** text
  cgv.visit_text(text)
- **Any other resources:** None
- **Oracle:**
  The visitor creates a string of the HTML representation which is stored in variable end_string and it will look like this:”<b><i><u>I am dummy text</u></i></b>”

2.11 End-to-End Test

2.11.1 <End-to-End Test 1>

General program test. Should be set up separately.

```
html : HTML_FACTORY

text1: TEXT
text2: TEXT
text3: TEXT
text4: TEXT

title1: TITLE
title2: TITLE

image1: IMAGE

table1:TABLES

row1: LINKED_LIST[COMPONENT]

link1:LINK
link2:LINK
link3:LINK

anchor1:ANCHOR

ordered_list1: LISTS
unordered_list1: LISTS

paragraph1:PARAGRAPH

snippet1:SNIPPET

manager: HTML_ANCHOR_MANAGER


title1:=html.title(“Main title”,1)
list.extend(title1)

manager.make
anchor1:= manager.create_anchor
list.extend(anchor1)

text1:=html.text(“I am bold”)
text1.bold

link1:=html.external_link(“www.google.com”, “Link to Google”)
```

```
paragraph1:= html.paragraph()
paragraph1.new_element(text1)
paragraph1.new_element(link1)
list.extend(paragraph1)

text2:=html.text(“I am cursive”)
text2.cursive

text3:=html.text(“I am normal”)

image1:=html.image(“testimage.jpg”,”This is a test image”)

unordered_list1:=html.unordered_list()
unordered_list1.add_row(text1)
unordered_list1.add_row(text2)

table1:=html.table(2)
table1.add_row([text2,image1])
table1.add_row([unordered_list1,text3])
list.extend(table1)

title2:=html.title(“Subtitle”,3)
list.extend(title2)

link2:=html.link(“C://…”,”go to second page”)

snippet1:=html.snippet(“<b>I am a snippet</b>”)

text4:=html.text(“I am bold and cursive and underlined”)
text4.bold()
text4.cursive()
text4.underline()

ordered_list1:=html.ordered_list()
ordered_list1.add_row(link2)
ordered_list1.add_row(snippet1)
ordered_list1.add_row(text4)
list.extend(ordered_list1)

link:=html.internal_link(anchor1, “Go to top”)
list.extend(link)
```

Oracle: The final HTML representation should then look like:

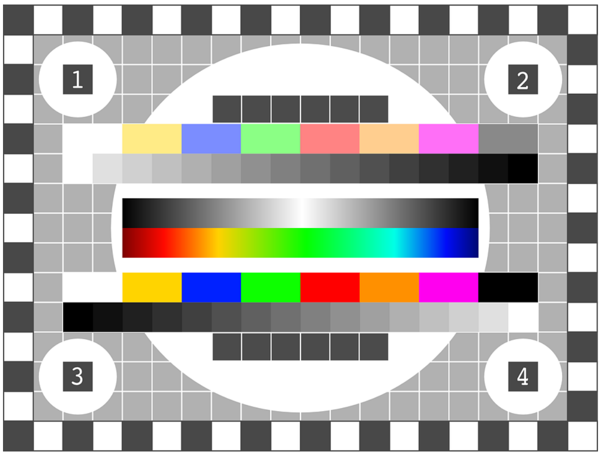
```
<!DOCTYPE html>
<html lang="en">
<head>
    <META CHARSET="UTF-8">
    <title>A document created with a HTML Generator</title>
</head>
<body>
```

```
    <h1>Main title</h1>
    <a name="00000001"></a>
    <p><b>I am bold</b><a href="www.google.com">Link to Google</a></p>
    <table>
                            <tr>
                            <th><i>I am cursive</i></th>
                            <th><img src="testimage.jpg" alt="This is a test image"></th>
                            </tr>
                            <tr>
                            <th><ul>
    <li><b>I am bold</b></li>
    <li><i>I am cursive</i></li>
</ul></th>
                            <th>I am normal</th>
                            </tr>
    </table>
    <h3>Subtitle</h3>
    <ol>
                            <li><a href="C://...">go to second page</a></li>
                            <li><b>I am a snippet</b></li>
                            <li><b><i><u>I am bold and cursive and
underlined</u></i></b></li>
    </ol>
    <a href="#00000001">Go to top</a>

</body>
</html>
```

Main Title

I am bold [Link to Google](www.google.com)

I am cursiv	
• **I am bold**	I am normal

• *I am cursiv*	

Subtitle

1. Go to second page (does not work in word)
2. **I am a snippet**
3. ***I am bold and cursiv and underlined***

Go to top (does not work here either)

3 Expected Coverage

The estimated coverage over the whole program is 71%