

Introduction

The **Anchor** widget that represents a simple <a> element.

Class declaration

Following is the declaration for **com.google.gwt.user.client.ui.Anchor** class:

```
public class Anchor
    extends FocusWidget
    implements HasHorizontalAlignment, HasName,
        HasHTML, HasWordWrap, HasDirection
```

CSS style rules

Following default CSS Style rule will be applied to all the Image widget. You can override it as per your requirements.

```
.gwt-Anchor { }
```

Class constructors

S.N.	Constructor & Description
------	---------------------------

- | | |
|---|--|
| 1 | Anchor
Creates an empty anchor. |
| 2 | protected AnchorElementelement
This constructor may be used by subclasses to explicitly use an existing element. |
| 3 | Anchorjava.lang.Stringtext
Creates an anchor for scripting. |
| 4 | Anchorjava.lang.Stringtext, booleanasHtml
Creates an anchor for scripting. |
| 5 | Anchorjava.lang.Stringtext, booleanasHTML, java.lang.Stringhref
Creates an anchor with its text and href <i>targetURL</i> specified. |
| 6 | Anchorjava.lang.Stringtext, booleanasHtml, java.lang.Stringhref, java.lang.Stringtarget
Creates a source anchor <i>linktoURI</i> . |
| 7 | |

Anchor*java. lang. Stringtext, java. lang. Stringhref*

Creates an anchor with its text and href *targetURL* specified.

8

Anchor*java. lang. Stringtext, java. lang. Stringhref, java. lang. Stringtarget*

Creates a source anchor with a frame target.

Class methods

S.N.	Function name & Description
------	-----------------------------

1	
---	--

	HasDirection.Direction getDirection
--	--

	Gets the directionality of the widget.
--	--

2	
---	--

	HasHorizontalAlignment.HorizontalAlignmentConstant getHorizontalAlignment
--	--

	Gets the horizontal alignment.
--	--------------------------------

3	
---	--

	java.lang.String getHref
--	---------------------------------

	Gets the anchor's href <i>theurltowhichitlinks</i> .
--	--

4	
---	--

	java.lang.String getHTML
--	---------------------------------

	Gets this object's contents as HTML.
--	--------------------------------------

5	
---	--

	java.lang.String getName
--	---------------------------------

	Gets the widget's name.
--	-------------------------

6	
---	--

	int getTabIndex
--	------------------------

	Gets the widget's position in the tab index.
--	--

7	
---	--

	java.lang.String getTarget
--	-----------------------------------

	Gets the anchor's target frame <i>theframeinwhichnavigationwilloccurwhenthelinkisselected</i> .
--	---

8	
---	--

	java.lang.String getText
--	---------------------------------

	Gets this object's text.
--	--------------------------

9	
---	--

	boolean getWordWrap
--	----------------------------

	Gets whether word-wrapping is enabled.
--	--

- 10 **void setAccessKey***charkey*
Sets the widget's 'access key'.
- 11 **void setFocus***booleanfocused*
Explicitly focus/unfocus this widget.
- 12 **void setHorizontalAlignment***HasHorizontalAlignment. HorizontalAlignmentConstantalign*
Sets the horizontal alignment.
- 13 **void setHref***java. lang. Stringhref*
Sets the anchor's href *theurltowhichitlinks*.
- 14 **void setHTML***java. lang. Stringhtml*
Sets this object's contents via HTML.
- 15 **void setName***java. lang. Stringname*
Sets the widget's name.
- 16 **void setTabIndex***intindex*
Sets the widget's position in the tab index.
- 17 **void setText***java. lang. Stringtext*
Sets this object's text.
- 18 **void setWordWrap***booleanwrap*
Sets whether word-wrapping is enabled.
- 19 **static Anchor wrap***Elementelement*
Creates an Anchor widget that wraps an existing <a> element.

Methods inherited

This class inherits methods from the following classes:

- com.google.gwt.user.client.ui.UIObject
- com.google.gwt.user.client.ui.Widget

- `com.google.gwt.user.client.ui.FocusWidget`

Anchor Widget Example

This example will take you through simple steps to show usage of a Anchor Widget in GWT. Follow the following steps to update the GWT application we created in *GWT - Create Application* chapter:

Step	Description
1	Create a project with a name <i>HelloWorld</i> under a package <i>com.tutorialspoint</i> as explained in the <i>GWT - Create Application</i> chapter.
2	Modify <i>HelloWorld.gwt.xml</i> , <i>HelloWorld.css</i> , <i>HelloWorld.html</i> and <i>HelloWorld.java</i> as explained below. Keep rest of the files unchanged.
3	Compile and run the application to verify the result of the implemented logic.

Following is the content of the modified module descriptor **src/com.tutorialspoint/HelloWorld.gwt.xml**.

```
<?xml version="1.0" encoding="UTF-8"?>
<module rename-to='helloworld'>
  <!-- Inherit the core Web Toolkit stuff.                        -->
  <inherits name='com.google.gwt.user.User'/>

  <!-- Inherit the default GWT style sheet.                      -->
  <inherits name='com.google.gwt.user.theme.clean.Clean'/>

  <!-- Specify the app entry point class.                        -->
  <entry-point class='com.tutorialspoint.client.HelloWorld'/>

  <!-- Specify the paths for translatable code                  -->
  <source path='client'/>
  <source path='shared'/>

</module>
```

Following is the content of the modified Style Sheet file **war/HelloWorld.css**.

```
body{
    text-align: center;
    font-family: verdana, sans-serif;
}
h1{
    font-size: 2em;
    font-weight: bold;
    color: #777777;
    margin: 40px 0px 70px;
    text-align: center;
}
```

Following is the content of the modified HTML host file **war/HelloWorld.html**.

```
<html>
<head>
<title>Hello World</title>
    <link rel="stylesheet" href="HelloWorld.css"/>
    <script language="javascript" src="helloworld/helloworld.nocache.js">
    </script>
</head>
<body>

<h1>Anchor Widget Demonstration</h1>
<div ></div>
```

```
</body>
</html>
```

Let us have following content of Java file **src/com.tutorialspoint/HelloWorld.java** which will demonstrate use of Anchor widget.

```
package com.tutorialspoint.client;

import com.google.gwt.core.client.EntryPoint;
import com.google.gwt.user.client.ui.Anchor;
import com.google.gwt.user.client.ui.RootPanel;
import com.google.gwt.user.client.ui.VerticalPanel;

public class HelloWorld implements EntryPoint {
    public void onModuleLoad() {
        // Create a Anchor widget,
        // pass text as Tutorialspoint
        // set asHtml as false,
        // href as www.tutorialspoint.com,
        // target as _blank
        Anchor anchor = new Anchor("Tutorialspoint",
                                   false,
                                   "http://www.tutorialspoint.com",
                                   "_blank");

        // Add anchor to the root panel.
        VerticalPanel panel = new VerticalPanel();
        panel.add(anchor);

        RootPanel.get("gwtContainer").add(panel);
    }
}
```

Once you are ready with all the changes done, let us compile and run the application in development mode as we did in [GWT - Create Application](#) chapter. If everything is fine with your application, this will produce following result:

