

# GWT - RADIOBUTTON WIDGET

[http://www.tutorialspoint.com/gwt/gwt\\_radiobutton\\_widget.htm](http://www.tutorialspoint.com/gwt/gwt_radiobutton_widget.htm)

Copyright © tutorialspoint.com

## Introduction

The **RadioButton** widget represents a mutually exclusive selection radio button.

## Class declaration

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

```
public class RadioButton
    extends CheckBox
```

## CSS style rules

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

```
.gwt-RadioButton {}
```

## Class constructors

### S.N. Constructor & Description

- RadioButton***java.lang.Stringname*  
Creates a new radio associated with a particular group name.
- RadioButton***java.lang.Stringname, java.lang.Stringlabel*  
Creates a new radio associated with a particular group, and initialized with the given HTML label.
- RadioButton***java.lang.Stringname, java.lang.Stringlabel, booleanasHTML*  
Creates a new radio button associated with a particular group, and initialized with the given label *optionallytreatedasHTML*.

## Class methods

### S.N. Function name & Description

- void setName***java.lang.Stringname*  
Change the group name of this radio button.

## 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
- com.google.gwt.user.client.ui.CheckBox
- java.lang.Object

## RadioButton Widget Example

This example will take you through simple steps to show usage of a RadioButton 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;
}
.gwt-RadioButton{
  color:green;
}
```

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>RadioButton 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 RadioButton widget.

```

package com.tutorialspoint.client;

import com.google.gwt.core.client.EntryPoint;
import com.google.gwt.user.client.ui.RadioButton;
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 some radio buttons, all in one group 'radioGroup'.
        RadioButton radioButton1 = new RadioButton("radioGroup", "First");
        RadioButton radioButton2 = new RadioButton("radioGroup", "Second");
        RadioButton radioButton3 = new RadioButton("radioGroup", "Third");

        // Check 'First' by default.
        radioButton1.setValue(true);

        //disable 'Second' radio button
        radioButton2.setEnabled(false);

        // Add toggle button to the root panel.
        VerticalPanel panel = new VerticalPanel();
        panel.setSpacing(10);
        panel.add(radioButton1);
        panel.add(radioButton2);
        panel.add(radioButton3);

        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:



- First
- Second
- Third

Loading [Mathjax]/jax/output/HTML-CSS/jax.js