

# RUBY CGI SESSIONS

[http://www.tutorialspoint.com/ruby/ruby CGI\\_sessions.htm](http://www.tutorialspoint.com/ruby/ruby CGI_sessions.htm)

Copyright © tutorialspoint.com

A CGI::Session maintains a persistent state for Web users in a CGI environment. Sessions should be closed after use, as this ensures that their data is written out to the store. When you've permanently finished with a session, you should delete it.

```
#!/usr/bin/ruby

require 'cgi'
require 'cgi/session'
cgi = CGI.new("html4")

sess = CGI::Session.new( cgi, "session_key" => "a_test",
                        "prefix" => "rubysess.")

lastaccess = sess["lastaccess"].to_s
sess["lastaccess"] = Time.now
if cgi['bgcolor'][0] =~ /[a-z]/
  sess["bgcolor"] = cgi['bgcolor']
end

cgi.out{
  cgi.html {
    cgi.body ("bgcolor" => sess["bgcolor"]){
      "The background of this page" +
      "changes based on the 'bgcolor'" +
      "each user has in session." +
      "Last access time: #{lastaccess}"
    }
  }
}
```

Accessing `/cgi-bin/test.cgi?bgcolor=red` would turn the page red for a single user for each successive hit until a new "bgcolor" was specified via the URL.

Session data is stored in a temporary file for each session, and the prefix parameter assigns a string to be prepended to the filename, making your sessions easy to identify on the filesystem of the server.

CGI::Session still lacks many features, such as the capability to store objects other than Strings, session storage across multiple servers

## Class CGI::Session:

A CGI::Session maintains a persistent state for web users in a CGI environment. Sessions may be memory-resident or may be stored on disk.

## Class methods:

Ruby class *Class CGI::Session* provides a single class method to create a session:

```
CGI::Session::new( cgi[, option])
```

Starts a new CGI session and returns the corresponding CGI::Session object. option may be an option hash specifying one or more of the following:

- **session\_key:** Key name holding the session ID. Default is `_session_id`.
- **session\_id:** Unique session ID. Generated automatically
- **new\_session:** If true, create a new session id for this session. If false, use an existing session identified by `session_id`. If omitted, use an existing session if available, otherwise create a new one.

- **database\_manager:** Class to use to save sessions; may be CGI::Session::FileStore or CGI::Session::MemoryStore. Default is FileStore.
- **tmpdir:** For FileStore, directory for session files.
- **prefix:** For FileStore, prefix of session filenames.

## Instance Methods:

SN	Methods with Description
1	<b>[ ]</b> Returns the value for the given key. See example above.
2	<b>[ ]=</b> Sets the value for the given key. See example above.
3	<b>delete</b> Calls the delete method of the underlying database manager. For FileStore, deletes the physical file containing the session. For MemoryStore, removes the session from memory.
4	<b>update</b> Calls the update method of the underlying database manager. For FileStore, writes the session data out to disk. Has no effect with MemoryStore.