

## [/home/s1211436/www\_local/wp/janken\_cookie.rhtml]

```

<%# encoding: utf-8 %>
<%
begin
  error = ""
  require 'cgi'
  @cgi = CGI.new

  def read_cookies()
    cookies = @cgi.cookies
    @win = ERB::Util.h((cookies["win"][0] || 0))
    @draw = ERB::Util.h((cookies["draw"][0] || 0))
    @lose = ERB::Util.h((cookies["lose"][0] || 0))
  end

  read_cookies
  print @cgi.header("charset" => "utf-8")

rescue => ex
  error = "<p>" + ex.message + "</p><p>" + ERB::Util.h(ex.backtrace.join("\n")) + "</p>"
  error.gsub!(/\n/, "<br>");
end
%>

<!doctype html>
<html lang="ja">
<head>
  <meta charset="utf-8" />
  <meta name="viewport" content="width=device-width, initial-scale=1">
  <title>じゃんけん</title>
</head>
<body>
  <% if !error.empty? %>
    <p><%= error %></p>
  <% end %>
  <h1>じゃんけん</h1>
  <p>現在の勝敗: <%=@win%>勝 <%=@lose%>敗 <%=@draw%>引き分け</p>
  <form action="judge_cookie.rhtml" method="post">
    <p>今度の手は?</p>
    <label><input type="radio" name="janken" value="0">グー</label>
    <label><input type="radio" name="janken" value="1">チョキ</label>
    <label><input type="radio" name="janken" value="2">パー</label>
    <input type="submit" name = "submit" value="勝負!">
    <input type="submit" name = "submit" value="勝敗をリセット">
  </form>
</body>
</html>

```

エラーメッセージを  
見やすくしました。

```

@cgi = CGI.new
@cgi['submit']

```

の値を使ってどちらのボタンがクリックされた  
かを判別します。

```
[/home/s1211436/www_local/wp/judge_cookie.rhtml]
```

```
<%# encoding: utf-8 %>
<%
begin
  error = ""
  require 'cgi'
  @cgi = CGI.new

  def check_submit
    if @cgi["submit"] == "勝敗をリセット"
      @win = 0; @draw = 0; @lose = 0;
      send_cookies
      raise "勝敗をリセット"
    end
  end

  def read_cookies
    cookies = @cgi.cookies
    @win = ERB::Util.h((cookies["win"][0] || 0)).to_i
    @draw = ERB::Util.h((cookies["draw"][0] || 0)).to_i
    @lose = ERB::Util.h((cookies["lose"][0] || 0)).to_i
  end

  def check_form
    if @cgi["janken"].empty?
      raise "手を出しましょう"
    elsif !(@cgi["janken"] == "0" || @cgi["janken"] == "1" || @cgi["janken"] == "2")
      raise "グー・チョキ・パー以外の手が出ました"
    end
  end

  def fight
    @cpu = rand(3)
    case @cgi["janken"].to_i - @cpu
    when -1, 2
      @result = "勝ち"
      @win += 1
    when 0
      @result = "引き分け"
      @draw += 1
    when -2, 1
      @result = "負け"
      @lose += 1
    end
  end

  def send_cookies
    win_cookie = CGI::Cookie.new("name" => "win", "value" => @win.to_s)
    draw_cookie = CGI::Cookie.new("name" => "draw", "value" => @draw.to_s)
    lose_cookie = CGI::Cookie.new("name" => "lose", "value" => @lose.to_s)
    print @cgi.header("type" => "text/html", "charset" => "utf-8", "cookie" => [win_cookie, draw_cookie,
lose_cookie])
  end

  def janken_dic (janken)
    case janken
    when 0
      "グー"
    when 1
      "チョキ"
    when 2
      "パー"
    end
  end
end
```

勝敗をリセットするときは、check\_submit以降の関数は実行してほしくないなので、raiseを使うとすっきり書けます。

もし、@cgi["janken"].to\_i が0から2以外にあることを条件にしようとする、to\_iメソッドは数字でない文字列を受け取ると0を返すので、URLに ?janken=hogehoge などと書いても例外が投げられません。

```

    check_submit
    read_cookies
    check_form
    fight
    send_cookies

rescue => ex
  if ex.message == "グー・チョキ・パー以外の手が出ました"
    error = ex.message
  elsif ex.message == "手を出しましょう"
    error = ex.message
  elsif ex.message == "勝敗をリセット"
    error = ex.message
  else
    error = "<p>" + ex.message + "</p><p>" + ERB::Util.h(ex.backtrace.join("\n")) + "</p>"
    error.gsub!(/\n/, "<br>");
  end
end

%>

<!doctype html>
<html lang="ja">
<head>
  <meta charset="utf-8" />
  <meta name="viewport" content="width=device-width, initial-scale=1">
  <title>ぼん！</title>
</head>

<% if error == "グー・チョキ・パー以外の手が出ました"%>
  <body>
    <h1>禁じられた手</h1>
    
    <p>現在の勝敗: <%=@win%>勝 <span id="lose"><%=@lose%></span>敗 <%=@draw%>引き分け</p>
    <p><a href="janken_cookie.rhtml">正々堂々と勝負する？</a></p>
  </body>
  <script>
    var text = document.getElementById("lose").textContent;
    var i = parseInt(text, 10);
    function penalty() {
      i += 1;
      document.getElementById("lose").innerHTML = i.toString();
      setTimeout("penalty()", 16);
    }
    penalty();
  </script>
</html>

<% elsif error == "手を出しましょう"%>
  <body>
    <h1>手を出しましょう</h1>
    <p>あなた: ㄥ( 'ω' )f</p>
    <p>現在の勝敗: <%=@win%>勝 <%=@lose%>敗 <%=@draw%>引き分け</p>
    <p><a href="janken_cookie.rhtml">今度は勝負する？</a></p>
  </body>
</html>

<% elsif error == "勝敗をリセット"%>
  <body>
    <h1>勝敗をリセット！</h1>
    <p>現在の勝敗: <%=@win%>勝 <%=@lose%>敗 <%=@draw%>引き分け</p>
    <p><a href="janken_cookie.rhtml">改めて勝負する？</a></p>
  </body>
</html>

```

?janken=3 などと入力して  
変な手を出そうとすると呪われます。

## 禁じられた手



現在の勝敗: 1勝 925敗 1引き分け

[正々堂々と勝負する？](#)

```
<% else %>
  <body>
    <% if !error.empty? %>
      <p><%= error %></p>
    <% end %>
    <h1>勝負! </h1>
    <p>あなた: <%= janken_dic(@cgi["janken"].to_i) %></p>
    <p>コンピュータ: <%= janken_dic(@cpu) %></p>
    <p><%= @result %></p>
    <p>現在の勝敗: <%=@win%>勝 <%=@lose%>敗 <%=@draw%>引き分け</p>
    <p><a href="janken_cookie.rhtml">もう一回勝負する?</a></p>
  </body>
</html>

<% end %>
```

[/home/s1211436/www\_local/wp/janken\_session.rhtml]  
以下はcookie版と比較して変更したところを太字で示します。

```
<%# encoding: utf-8 %>
<%
begin
  error = ""
  require 'cgi'
  require 'cgi/session'
  @cgi = CGI.new

  def read_session()
    session = CGI::Session.new(@cgi, "prefix" => "janken_")
    @win = ERB::Util.h((session["win"] || 0))
    @draw = ERB::Util.h((session["draw"] || 0))
    @lose = ERB::Util.h((session["lose"] || 0))
    session.close
  end

  read_session
  print @cgi.header("charset" => "utf-8")
rescue => ex
  error = "<p>" + ex.message + "</p><p>" + ERB::Util.h(ex.backtrace.join("\n")) + "</p>"
  error.gsub!(/\n/, "<br>");
end
%>

<!doctype html>
<html lang="ja">
<head>
  <meta charset="utf-8" />
  <meta name="viewport" content="width=device-width, initial-scale=1">
  <title>じゃんけん</title>
</head>
<body>
  <% if !error.empty? %>
    <p><%= error %></p>
  <% end %>
  <h1>じゃんけん</h1>
  <p>現在の勝敗: <%=@win%>勝 <%=@lose%>敗 <%=@draw%>引き分け</p>
  <form action="judge_session.rhtml" method="post">
    <p>今度の手は？</p>
    <label><input type="radio" name="janken" value="0">グー</label>
    <label><input type="radio" name="janken" value="1">チョキ</label>
    <label><input type="radio" name="janken" value="2">パー</label>
    <input type="submit" name = "submit" value="勝負！">
    <input type="submit" name = "submit" value="勝敗をリセット">
  </form>
</body>
</html>
```

```
[/home/s1211436/www_local/wp/judge_session.rhtml]
```

```
<%# encoding: utf-8 %>
<%
begin
  error = ""
  require 'cgi'
  require 'cgi/session'
  @cgi = CGI.new

  def check_submit
    if @cgi["submit"] == "勝敗をリセット"
      @win = 0; @draw = 0; @lose = 0;
      save_session
      raise "勝敗をリセット"
    end
  end

  def read_session
    session = CGI::Session.new(@cgi, "prefix" => "janken_")
    @win = ERB::Util.h(session["win"] || 0).to_i
    @draw = ERB::Util.h(session["draw"] || 0).to_i
    @lose = ERB::Util.h(session["lose"] || 0).to_i
    session.close
  end

  def check_form
    if @cgi["janken"].empty?
      raise "手を出しましょう"
    elsif !(@cgi["janken"] == "0" || @cgi["janken"] == "1" || @cgi["janken"] == "2")
      raise "グー・チョキ・パー以外の手が出ました"
    end
  end

  def fight
    @cpu = rand(3)
    case @cgi["janken"].to_i - @cpu
    when -1, 2
      @result = "勝ち"
      @win += 1
    when 0
      @result = "引き分け"
      @draw += 1
    when -2, 1
      @result = "負け"
      @lose += 1
    end
  end

  def save_session
    session = CGI::Session.new(@cgi, "prefix" => "janken_")
    session["win"] = @win
    session["draw"] = @draw
    session["lose"] = @lose
    session.close
  end
end
```

```

def janken_dic (janken)
  case janken
  when 0
    "グー"
  when 1
    "チョキ"
  when 2
    "パー"
  end
end

check_submit
read_session
check_form
fight
save_session

rescue => ex
  if ex.message == "グー・チョキ・パー以外の手が出ました"
    error = ex.message
  elsif ex.message == "手を出しましょう"
    error = ex.message
  elsif ex.message == "勝敗をリセット"
    error = ex.message
  else
    error = "<p>" + ex.message + "</p><p>" + ERB::Util.h(ex.backtrace.join("\n")) + "</p>"
    error.gsub!(/\n/, "<br>");
  end
end

%>

<!doctype html>
<html lang="ja">
<head>
  <meta charset="utf-8" />
  <meta name="viewport" content="width=device-width, initial-scale=1">
  <title>ぼん！</title>
</head>

<% if error == "グー・チョキ・パー以外の手が出ました"%>
  <body>
    <h1>禁じられた手</h1>
    
    <p>現在の勝敗: <%=@win%>勝 <span id="lose"><%=@lose%></span>敗 <%=@draw%>引き分け</p>
    <p><a href="janken_session.rhtml">正々堂々と勝負する？</a></p>
  </body>
  <script>
    var text = document.getElementById("lose").textContent;
    var i = parseInt(text, 10);
    function penalty() {
      i += 1;
      document.getElementById("lose").innerHTML = i.toString();
      setTimeout("penalty()", 16);
    }
    penalty();
  </script>
</html>

```

```

<% elsif error == "手を出しましょう"%>
  <body>
    <h1>手を出しましょう</h1>
    <p>あなた: \('ω')f</p>
    <p>現在の勝敗: <%= @win %>勝 <%= @lose %>敗 <%= @draw %>引き分け</p>
    <p><a href="janken_session.rhtml">今度は勝負する?</a></p>
  </body>
</html>

<% elsif error == "勝敗をリセット"%>
  <body>
    <h1>勝敗をリセット!</h1>
    <p>現在の勝敗: <%= @win %>勝 <%= @lose %>敗 <%= @draw %>引き分け</p>
    <p><a href="janken_session.rhtml">改めて勝負する?</a></p>
  </body>
</html>

<% else %>
  <body>
    <% if !error.empty? %>
      <p><%= error %></p>
    <% end %>
    <h1>勝負!</h1>
    <p>あなた: <%= janken_dic(@cgi["janken"].to_i) %></p>
    <p>コンピュータ: <%= janken_dic(@cpu) %></p>
    <p><%= @result %></p>
    <p>現在の勝敗: <%= @win %>勝 <%= @lose %>敗 <%= @draw %>引き分け</p>
    <p><a href="janken_session.rhtml">もう一回勝負する?</a></p>
  </body>
</html>

<% end %>

```

## 感想など

erbは楽でいいですね。

セッションもクッキーも、クッキーを保存しているブラウザからアクセスした場合でしか情報が復元されないの、データ自体はデータベースに保存して、ログイン状態を保持するためにセッションを使うのが普通かなと思いました。次回の授業でセキュリティを扱うらしいので、次回のお楽しみでしょうか。

rubyとerbで権限が違くと仰っていたと思います。

loggerが書き込むテキストが604だとPermission Deniedになりますね。606にしないとLogger.newできません。

```

Permission denied - /home/s1211436/wp_log.txt
/usr/local/lib/ruby/1.9.1/logger.rb:592:in `initialize'
/usr/local/lib/ruby/1.9.1/logger.rb:592:in `open'
/usr/local/lib/ruby/1.9.1/logger.rb:592:in `open_logfile'
/usr/local/lib/ruby/1.9.1/logger.rb:549:in `initialize'
/usr/local/lib/ruby/1.9.1/logger.rb:314:in `new'
/usr/local/lib/ruby/1.9.1/logger.rb:314:in `initialize'
/home/s1211436/www_local/wp/janken_session.rhtml:10:in `new'
/home/s1211436/www_local/wp/janken_session.rhtml:10:in `run'
/usr/local/lib/ruby/1.9.1/apache/erb-run.rb:144:in `eval'
/usr/local/lib/ruby/1.9.1/apache/erb-run.rb:144:in `run'
/usr/local/lib/ruby/1.9.1/apache/erb-run.rb:109:in `handler'

```