**Rock Paper Scissors!**

1342292% of 544452 of5,244[brunolm](http://www.codewars.com/users/brunolm" \o "This kata's Sensei)

C#

* [TRAIN AGAIN](http://www.codewars.com/kata/rock-paper-scissors/train/csharp)
* [NEXT KATA](http://www.codewars.com/trainer/csharp)

Details

[Solutions](http://www.codewars.com/kata/rock-paper-scissors/solutions/csharp)

[Forks (6)](http://www.codewars.com/kata/rock-paper-scissors/forks/csharp)

[Discourse (27)](http://www.codewars.com/kata/rock-paper-scissors/discuss/csharp)

* Add to Collection
* |
* Share this kata:

**Rock Paper Scissors**

Let's play! You have to return which player won! In case of a draw return Draw!.

Examples:

rps('scissors','paper') // Player 1 won!

rps('scissors','rock') // Player 2 won!

rps('paper','paper') // Draw!

<http://www.codewars.com/kata/rock-paper-scissors/csharp>

public string Rps(string p1, string p2)

{

//'scissors','paper') // Player 1 won!

//rps('scissors','rock') // Player 2 won!

//rps('paper'

if (p1 == "scissors" && p2 == "paper")

{

return "Player 1 won!";

}

else if (p1 == "scissors" && p2 == "rock")

{

return "Player 2 won!";

}

else if (p1 == "scissors" && p2 == "scissors")

{

return "Draw!";

}

if (p1 == "paper" && p2 == "paper")

{

return "Draw!";

}

else if (p1 == "paper" && p2 == "rock")

{

return "Player 1 won!";

}

else if (p1 == "paper" && p2 == "scissors")

{

return "Player 2 won!";

}

if (p1 == "rock" && p2 == "paper")

{

return "Player 2 won!";

}

else if (p1 == "rock" && p2 == "rock")

{

return "Draw!";

}

else if (p1 == "rock" && p2 == "scissors")

{

return "Player 1 won!";

}

return "";

}

-----------otras soluciones---------------

public class Kata

{

public string Rps(string p1, string p2)

{

if (p1 == p2) return "Draw!";

int p = (p1 + p2) == "scissorspaper" || (p1 + p2) == "rockscissors" || (p1 + p2) == "paperrock" ? 1 : 2;

return "Player " + p + " won!";

}

}

------------------------

public class Kata

{

public string Rps(string p1, string p2)

{

if (p1 == p2)

return "Draw!";

if (((p1 == "rock") && (p2 == "scissors")) ||

((p1 == "scissors") && (p2 == "paper")) ||

((p1 == "paper") && (p2 == "rock")))

{

return "Player 1 won!";

}

else

{

return "Player 2 won!";

}

}

}

**using System;**

**using System.Collections.Generic;**

**public class Kata**

**{**

**private static readonly IDictionary<Tuple<string, string>, string> Games =**

**new Dictionary<Tuple<string, string>, string> {**

**{ Tuple.Create("rock", "rock"), "Draw!" },**

**{ Tuple.Create("rock", "paper"), "Player 2 won!" },**

**{ Tuple.Create("rock", "scissors"), "Player 1 won!" },**

**{ Tuple.Create("paper", "rock"), "Player 1 won!" },**

**{ Tuple.Create("paper", "paper"), "Draw!" },**

**{ Tuple.Create("paper", "scissors"), "Player 2 won!" },**

**{ Tuple.Create("scissors", "rock"), "Player 2 won!" },**

**{ Tuple.Create("scissors", "paper"), "Player 1 won!" },**

**{ Tuple.Create("scissors", "scissors"), "Draw!" },**

**};**

**public string Rps(string p1, string p2)**

**{**

**return Games[Tuple.Create(p1, p2)];**

**}**

**}**

public string Rps(string p1, string p2)

{

return p1 == p2 ? "Draw!" :

(p1 == "rock" && p2 == "scissors" || p1 == "scissors" && p2 == "paper" || p1 == "paper" && p2 == "rock")

? "Player 1 won!" : "Player 2 won!";

}