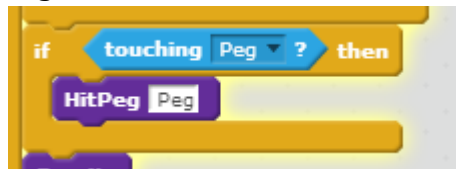


1. Log on to Scratch <http://scratch.mit.edu/>
2. Select "Try it out."
3. Upload your file from the previous exercise.
4. **Create a Round Peg for your ball to bounce off of.**
 - 4.1. Make a new Sprite called **Peg** and give it a round costume centered about its origin.
 - 4.2. Select the **Ball** sprite and make a new **Block** called **HitPeg** with text argument for thePeg.
 - 4.2.1. Select **More Blocks**.
 - 4.2.2. Make a **Block** called **HitPeg**.
 - 4.2.3. Select **Options**.
 - 4.2.4. Add string input and rename it **thePeg**.
 - 4.2.5. Select **Run without screen refresh** and press **OK**.
 - 4.3. Add an IF-THEN control to your forever loop that will detect when the ball hits the Peg.
 - 4.4. Add **HitPeg** to this IF-THEN and type the name of your peg in as its argument.



5. **Define** the angle that we bounce off. In contrast to the Kicker where the ball goes straight away, we will calculate a proper reflection angle for the ball to bounce off of the peg. In order for this to work both the ball and peg must be round and centered!
 - 5.1. Create two local variables: **myAngle** and **yourAngle**. We will use these to store the angle the ball is traveling, and the angle from the ball to the peg.
 - 5.2. In the HitPeg Definition block save the ball's current **direction** in **myAngle** (set myAngle to **Direction**).
 - 5.3. To find which direction the Peg is, **Point towards thePeg**. **thePeg** is at the top of your define block – Don't use the lookup in point towards!
 - 5.4. Save this direction in **yourAngle**. (**Set yourAngle to Direction**)
 - 5.5. Point your ball sprite in the direction of **2*yourAngle - myAngle - 180**.

- 5.6. Test your code. Make sure that the ball bounces correctly off the pin from all different angles.
- 5.7. You should notice strange behavior that gets worse at higher speeds. Try to figure out what might be going wrong. The next two sections provide solutions that will prevent your ball from getting stuck on a peg or other object.
6. Write a function called **backOff** with an argument called **theObject**. This function should move your ball backwards until it is no longer touching the object (the ball's direction should be the same after the function as it was before).
 - 6.1. Add a call to this new function to the beginning of your **HitPeg** function block and see if it fixes the problem you observed in 5.7.
7. Also write a function called **moveOn** with an argument called **theObject**. This function should move your ball forward a small amount and then keep moving until it is no longer touching the object.
 - 7.1. Add this new function to the end of your **HitPeg** function block.
8. **Make several more Pegs**
 - 8.1. Copy your **Peg** sprite several times and give each a new name.
 - 8.2. Add another **IF** statement to your forever loop for each **Peg**. Change the names in the If and Call statements to match the new pegs (Peg1, Peg2, etc..).
 - 8.3. Test and debug your code.
9. Save your work (Pinball #4).