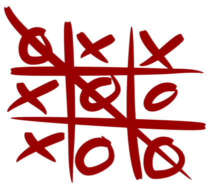
More Tic Tac Toe!



Part 7: Whose turn is it anyway?

We know Tic Tac Toe is a turn-based game, so we have to be able to see whose turn it is! We can do this using variables and if statements at the end of each turn. We can also program the computer to automatically remember the players' symbols!

Task 7.1

1. Delete the `input_symbol` code from **Task 5.1**.
2. Create a variable at the start of your code (outside the loop) called `current_player` and set it to `player0` or `playerX`. (who do you think should go first??)
3. On the next line, create another variable `current_symbol` and set it to `current_player`'s symbol.
4. Use an `if` statement to change the current player at the end of the turn. If the current player is `player0`, change it to `playerX`, and vice versa.
5. Remember to change `current_symbol` to an 0 or an X!

Part 8: That one is already taken!

Users always do the unexpected, and they might choose a square that has already been used. What happens if we try to ask for a square that has already been used?

Goal: Check if the selected square is available before we accept their choice.

Task 8.1

1. Underneath where you asked the user for the square in task 5.3, write some code to check if the input square is actually an available square.
2. Place the code you just wrote in a `while` loop so if the square's not valid we keep asking the same player for input until they enter something valid.

You can use:

- `not in` to check if the input square number is one of the numbers in `grid`
- `current_player` to tell the player if the need to try again
- Print the grid again to help them chose a better square!

Part 9: And the winner is…

Well, actually, it’s kind of hard to tell at the moment. Our program will make you play all nine turns, even if someone has already won and it still won't tell you who won in the end. That’s no fun!!

Goal: Figure out all the ways someone can win the game

Task 9.1

There are 8 different ways to win the game, fill out the remaining grids below!

```
  Pattern 1              Pattern 2              Pattern 3              Pattern 4
-------------          -------------          -------------          -------------
| X | X | X |    OR    |   |   |   |    OR    |   |   |   |    OR    |   |   |   |
-------------          -------------          -------------          -------------
|   |   |   |          | X | X | X |          |   |   |   |          |   |   |   |
-------------          -------------          -------------          -------------
|   |   |   |          |   |   |   |          |   |   |   |          |   |   |   |
-------------          -------------          -------------          -------------

  Pattern 5              Pattern 6              Pattern 7              Pattern 8
-------------          -------------          -------------          -------------
|   |   |   |    OR    |   |   |   |    OR    |   |   |   |    OR    |   |   |   |
-------------          -------------          -------------          -------------
|   |   |   |          |   |   |   |          |   |   |   |          |   |   |   |
-------------          -------------          -------------          -------------
|   |   |   |          |   |   |   |          |   |   |   |          |   |   |   |
-------------          -------------          -------------          -------------
```

Goal: Create a variable to keep track of the winner

Task 9.2

1. At the start of the game, create a new variable and call it `winning_player`. Set it to `"no one"`

Goal: Check if someone wins with Pattern 1

Task 9.3

Write an if statement to check in grid list to see if square 1, 2 and 3 are all the same.
If someone has won update `winning_player = current_player`

Add this code inside your while loop after you print out the grid.

You need to:

- Look items up in `grid`
- Use an `if` statement
- Add `and` to your if statement

Goal: Workout the code for the rest of the patterns!

Task 9.4

Write code to check if any of the other winning patterns are there!

1. After your code to check for pattern check for pattern 1, use `elif` a bunch of times to check for the 7 other patterns.
2. Inside each of the patterns make sure to update the `winning_player`

3. **Bonus:** Instead of LOTS of `elifs`, put all you lines of code in one if statement. Join them to your pattern 1 if statement using `or` (you can use multiple lines to make it neater if you put brackets around you line them all in brackets

```
if (Pattern 1
    or Pattern 2
    or ......
    or ...... ):
```

Goal: At the end of each turn, check all the possible ways of winning to see if the player has won.

Task 9.1

2. At the start of the game, create a new variable and call it `winning_player`. Set it to `"no one"`
3. Use `if` and `elif` statements to check if the same symbol is in each of the rows, columns and diagonals. (Hint: there are 8 ways of winning so you will need 8 statements)
4. Inside all the if statements, if it's true, then set the winning player to the current player `winning_player = current_player`.
5. Inside the loop, after you record the winner, use the `break` function to end the game `while` loop.)

Bonus!

1. Instead of using `if` and `elif` statements, use `and` and `or` to create one `if` statement that checks all possible ways you can win.

Part 10: Announcing the winner!

What use is knowing the winner if we don't tell anyone!
Now for the best part! Telling the user who won!

Goal: Announcing the winner!

Task 10.2

1. At the end of your code, outside of the game `while` loop, create an `if` statement that checks if `winning_player` equals `player0` or `playerX`. If so, prints a line congratulating the `winning_player`!

But not every game of Tic-Tac-Toe ends with a winner! More often than not, it's a tie. So we need to let the users know.

Goal: If there are no moves available, and no one won, tell the players that it's a tie!

Task 10.3

1. After your loop, check to see if the game was won by someone by checking if `winning_player` is still no one.
2. If it is still `"no one"`, announce that the game was a tie
3. If it is not `"no one"`, print out "and the winner is" and then the name of the winning_player

Are you finished with all of this?

Let a tutor know. We've got more stuff for you!