Sudoku

Rules & Checking
Objects and References

Each row, column, and 3x3 sub-square must contain each of the digits 1..9

To check a row ...

<table>
<thead>
<tr>
<th>Symbols</th>
<th>Board</th>
</tr>
</thead>
<tbody>
<tr>
<td>1 -</td>
<td>9 8 7 6 5 4 3 2 9</td>
</tr>
<tr>
<td>2 -</td>
<td>2 4 6 1 7 3 9 8 5</td>
</tr>
<tr>
<td>3 -</td>
<td>3 5 1 9 2 8 7 4 6</td>
</tr>
<tr>
<td>4 -</td>
<td>1 2 8 5 3 7 6 9 4</td>
</tr>
<tr>
<td>5 -</td>
<td>6 3 4 8 9 2 1 5 7</td>
</tr>
<tr>
<td>6 -</td>
<td>5 1 9 2 8 6 4 7 3</td>
</tr>
<tr>
<td>7 -</td>
<td>7 9 5 4 6 1 8 3 2</td>
</tr>
<tr>
<td>8 -</td>
<td>4 7 2 3 1 9 5 6 8</td>
</tr>
<tr>
<td>9 -</td>
<td>8 6 3 7 4 5 2 1 9</td>
</tr>
</tbody>
</table>

To check a row ...

<table>
<thead>
<tr>
<th>Symbols</th>
<th>Board</th>
</tr>
</thead>
<tbody>
<tr>
<td>1 -</td>
<td>9 8 7 6 5 4 3 2 9</td>
</tr>
<tr>
<td>2 -</td>
<td>2 4 6 1 7 3 9 8 5</td>
</tr>
<tr>
<td>3 -</td>
<td>3 5 1 9 2 8 7 4 6</td>
</tr>
<tr>
<td>4 -</td>
<td>1 2 8 5 3 7 6 9 4</td>
</tr>
<tr>
<td>5 ✔</td>
<td>6 3 4 8 9 2 1 5 7</td>
</tr>
<tr>
<td>6 ✔</td>
<td>5 1 9 2 8 6 4 7 3</td>
</tr>
<tr>
<td>7 ✔</td>
<td>7 9 5 4 6 1 8 3 2</td>
</tr>
<tr>
<td>8 ✔</td>
<td>4 7 2 3 1 9 5 6 8</td>
</tr>
<tr>
<td>9 ✔</td>
<td>8 6 3 7 4 5 2 1 9</td>
</tr>
</tbody>
</table>
Oops.

<table>
<thead>
<tr>
<th>Symbols</th>
<th>Board</th>
</tr>
</thead>
<tbody>
<tr>
<td>1 -</td>
<td>9 8 7 6 5 4 3 2 9</td>
</tr>
<tr>
<td>2 ✔</td>
<td>2 4 6 1 7 3 9 8 5</td>
</tr>
<tr>
<td>3 ✔</td>
<td>3 5 1 9 2 8 7 4 6</td>
</tr>
<tr>
<td>4 ✔</td>
<td>1 2 8 5 3 7 6 9 4</td>
</tr>
<tr>
<td>5 ✔</td>
<td>6 3 4 8 9 2 1 5 7</td>
</tr>
<tr>
<td>6 ✔</td>
<td>5 1 9 2 8 6 4 7 3</td>
</tr>
<tr>
<td>7 ✔</td>
<td>7 9 5 4 6 1 8 3 2</td>
</tr>
<tr>
<td>8 ✔</td>
<td>4 7 2 3 1 9 5 6 8</td>
</tr>
<tr>
<td>9 ✔</td>
<td>8 6 3 7 4 5 2 1 9</td>
</tr>
</tbody>
</table>

Sets of symbols found in a row

We could start with the empty set and add symbols as they are found.

My choice: Start with all available symbols, cross out (remove) the ones we have used.

Python has a “set” type

```python
SYMBOLS = {"1", "2", "3", "4", "5", "6", "7", "8", "9"}

available = set(SYMBOLS)
available.remove("6")
print(available)


Note I removed “6” from a copy of the SYMBOLS set. Guess why?

>>> symbols = {‘a’, ‘b’, ‘c’, ‘d’}
>>> same = symbols
>>> another = set(symbols)
>>> same.remove(‘a’)
>>> same.remove(‘b’)
>>> another.remove(‘c’)
>>> another.remove(‘d’)
>>> same
{‘c’, ‘d’}
>>> another
{‘a’, ‘b’}
>>> symbols
{‘c’, ‘d’}

What’s going on here?
>>> symbols = {'a', 'b', 'c', 'd'}
>>> same = symbols
>>> another = set(symbols)

```
symbols
\[ 'a' \, 'b' \, 'c' \, 'd' \]

symbols
\[ 'a' \, 'b' \, 'c' \, 'd' \]

another
\[ 'a' \, 'b' \, 'c' \, 'd' \]
```

```python
>>> symbols
\{'a', 'b', 'c', 'd'\}
```

```python
>>> symbols = {'a', 'b', 'c', 'd'}
>>> same = symbols
>>> another = set(symbols)

>>> same.remove('a')
>>> same.remove('b')

>>> another.remove('c')
>>> another.remove('d')

>>> same
\{'c', 'd'\}
>>> another
\{'a', 'b'\}

symbols
\{'c', 'd'\}
```

```python
set(\ldots ) \text{ constructs a copy of a collection}
```

I want to avoid writing the “check unique entries” code three times.
Once looping through elements in a row
Once looping through elements in a column
Once looping through elements in a square

DRY: “Don’t Repeat Yourself”

How can I factor that code out?
I want to avoid writing the “check unique entries” code three times
   Once looping through elements in a row
   Once looping through elements in a column
   Once looping through elements in a square

DRY: “Don’t Repeat Yourself”

How can I factor that code out?

Write it just once, for a list of 9 elements

Board provides the same tiles in different lists, by row, column, and square

---

From checker to solver

For checker:
   Check uniqueness of symbols in each row
   Check uniqueness of symbols in each column
   Check uniqueness of symbols in square

For solver:
   Allowed symbols in a tile is intersection of
      symbols not yet used in row
      symbols not yet used in column
      symbols not yet used in square

---

Board contains only one copy of each tile, but each tile appears in three different lists.