Put True (T) or False (F) in every cell, based on characteristics of each type.

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>list</td>
<td>T</td>
<td>T</td>
<td>T</td>
<td>F</td>
<td>F</td>
</tr>
<tr>
<td>tuple</td>
<td>F</td>
<td>T</td>
<td>T</td>
<td>F</td>
<td>F</td>
</tr>
<tr>
<td>namedtuple</td>
<td>F</td>
<td>T</td>
<td>F</td>
<td>T</td>
<td>T</td>
</tr>
</tbody>
</table>

1. 

$\text{list}$

2. 

```
x = [1, 2, 3]
y = [1, 2, 3]
z = x
z.append(4)
```

3. 

```
nums1 = [1, 2]
nums2 = nums1
x = nums2.pop(1)
```

4. 

```
x = [1, 2]
y = [3]
z = x + y
y.append(4)  # only appends 4 to the object referenced by y
```

5. 

```
people = {"alice":30, "bob":25}
x = people
y = people["bob"]
x["alice"] = 31
y = 26  # only modifies y's reference. Does not change dict object instance.
```

6. 

```
def f(items):
    return items.pop(0)
nums = [1,2,3]
nums.append(f(nums))
```

Final list: $[2, 3, 1]$
Remember to import copy for these in Python Tutor!

```python
def biggest(items):
    items = copy.copy(items)
    items.sort()
    return items[-1]
```

```python
nums = [3, 9, 6]
x = biggest(nums)
```

```python
team1 = [
    {'name': 'A', 'age': 7}
]
team2 = copy.copy(team1)
team2.append(
    {'name': 'B', 'age': 9}
)
team2[0]['age'] = 8
x = team1[0]['age']
```

```python
# Same as above, but with copy.deepcopy(...) instead of copy.copy(...).
orig = [1, [2, [3, 4]]]
x = orig
y = copy.copy(orig)
z = copy.deepcopy(orig)
```