course_3_assessment_1

April 23, 2021

Q1) The variable nested contains a nested list. Assign 'snake' to the variable output using indexing.

[4]:
```
nested = [['dog', 'cat', 'horse'],
          ['frog', 'turtle', 'snake', 'gecko'],
          ['hamster', 'gerbil', 'rat', 'ferret']]
print(len(nested)) #to get how many items in the outer list
output = nested[1][2]
```

```
3
```

Q2) Below, a list of lists is provided. Use in and not in tests to create variables with Boolean values. See comments for further instructions.

[5]:
```
lst = [['apple', 'orange', 'banana'],
       [5, 6, 7, 8, 9.9, 10],
       ['green', 'yellow', 'purple', 'red']]

#Test to see if 'yellow' is in the third list of lst. Save to variable␣
 ↪``yellow``
yellow = 'yellow' in lst[2]

#Test to see if 4 is in the second list of lst. Save to variable ``four``
four = 4 in lst[1]

#Test to see if 'orange' is in the first element of lst. Save to variable␣
 ↪``orange``
orange = 'orange' in lst[0]
```

Q3) Below, we've provided a list of lists. Use in statements to create variables with Boolean values - see the ActiveCode window for further directions.

[6]:
```
L = [[5, 8, 7], ['hello', 'hi', 'hola'], [6.6, 1.54, 3.99], ['small', 'large']]

# Test if 'hola' is in the list L. Save to variable name test1
test1 = 'hola' in L
# Test if [5, 8, 7] is in the list L. Save to variable name test2
test2 = [5,8,7] in L
# Test if 6.6 is in the third element of list L. Save to variable name test3
test3 = 6.6 in L[2]
```

Q4) Provided is a nested data structure. Follow the instructions in the comments below. Do not hard code.

[4]:
```
nested = {
    'data': ['finding', 23, ['exercises', 'hangout', 34]],
     'window': ['part', 'whole', [], 'sum',
                     ['math', 'calculus', 'algebra', 'geometry',
                      'statistics',['physics', 'chemistry', 'biology']]]
         }
print(nested.keys())
# Check to see if the string data is a key in nested, if it is,
#assign True to the variable data, otherwise assign False.
data = 'data' in list(nested.keys())
# Check to see if the integer 24 is in the value of the key data,
#if it is then assign to the variable twentyfour the value of True, otherwise␣
 ↪False.
twentyfour = 24 in nested['data']
# Check to see that the string 'whole' is not in the value of the key window.␣
 ↪If it's not, then assign to the variable whole the value of True, otherwise␣
 ↪False.
whole = 'whole' not in nested['window']
# Check to see if the string 'physics' is a key in the dictionary nested. If it␣
 ↪is, assign to the variable physics, the value of True, otherwise False.
physics = 'physics' in list(nested.keys())
```

```
dict_keys(['data', 'window'])
```

Q5) The variable nested_d contains a nested dictionary with the gold medal counts for the top four countries in the past three Olympics. Assign the value of Great Britain's gold medal count from the London Olympics to the variable london_gold. Use indexing. Do not hardcode.

[5]:
```
nested_d = {'Beijing':{'China':51, 'USA':36, 'Russia':22, 'Great Britain':19},␣
 ↪'London':{'USA':46, 'China':38, 'Great Britain':29, 'Russia':22}, 'Rio':
 ↪{'USA':35, 'Great Britain':22, 'China':20, 'Germany':13}}

for k , v  in nested_d.items():
    print( k , v )
london_gold = nested_d['London']['Great Britain']
```

```
Beijing {'China': 51, 'USA': 36, 'Russia': 22, 'Great Britain': 19}
London {'USA': 46, 'China': 38, 'Great Britain': 29, 'Russia': 22}
Rio {'USA': 35, 'Great Britain': 22, 'China': 20, 'Germany': 13}
```

Q6) Below, we have provided a nested dictionary. Index into the dictionary to create variables that we have listed in the ActiveCode window.

[6]:

```
sports = {'swimming': ['butterfly', 'breaststroke', 'backstroke', 'freestyle'],␣
 ↪'diving': ['springboard', 'platform', 'synchronized'], 'track': ['sprint',␣
 ↪'distance', 'jumps', 'throws'], 'gymnastics': {'women':['vault', 'floor',␣
 ↪'uneven bars', 'balance beam'], 'men': ['vault', 'parallel bars', 'floor',␣
 ↪'rings']}}
for k , v in sports.items():
    print( k , v )
# Assign the string 'backstroke' to the name v1
v1 = sports['swimming'][2]
# Assign the string 'platform' to the name v2
v2 = sports['diving'][1]
# Assign the list ['vault', 'floor', 'uneven bars', 'balance beam'] to the name␣
 ↪v3
v3 = sports['gymnastics']['women']
# Assign the string 'rings' to the name v4
v4 = sports['gymnastics']['men'][3]
```

```
swimming ['butterfly', 'breaststroke', 'backstroke', 'freestyle']
diving ['springboard', 'platform', 'synchronized']
track ['sprint', 'distance', 'jumps', 'throws']
gymnastics {'women': ['vault', 'floor', 'uneven bars', 'balance beam'], 'men':
['vault', 'parallel bars', 'floor', 'rings']}
```

Q7) Given the dictionary, nested_d, save the medal count for the USA from all three Olympics in the dictionary to the list US_count.

[7]:

```
nested_d = {'Beijing':{'China':51, 'USA':36, 'Russia':22, 'Great Britain':19},␣
 ↪'London':{'USA':46, 'China':38, 'Great Britain':29, 'Russia':22}, 'Rio':
 ↪{'USA':35, 'Great Britain':22, 'China':20, 'Germany':13}}
for k , v in nested_d.items():
    print( k , v )
US_count = []

for inner_k1 in nested_d: #first iteration
    #print(inner_k1)
    for inner_k2 in nested_d[inner_k1]: #second iteration
        print(inner_k2)
        if inner_k2 == 'USA':
            US_count.append(nested_d[inner_k1][inner_k2])
```

```
Beijing {'China': 51, 'USA': 36, 'Russia': 22, 'Great Britain': 19}
London {'USA': 46, 'China': 38, 'Great Britain': 29, 'Russia': 22}
Rio {'USA': 35, 'Great Britain': 22, 'China': 20, 'Germany': 13}
China
USA
Russia
Great Britain
```

```
USA
China
Great Britain
Russia
USA
Great Britain
China
Germany
```

Q8) Iterate through the contents of l_of_l and assign the third element of sublist to a new list called third.

[8]:

```
l_of_l = [['purple', 'mauve', 'blue'],
          ['red', 'maroon', 'blood orange', 'crimson'],
          ['sea green', 'cornflower', 'lavender', 'indigo'],
          ['yellow', 'amarillo', 'mac n cheese', 'golden rod']]

print(len(l_of_l)) #it has 4 inner lists
third = []
for inner_lst1 in l_of_l:
    third.append(inner_lst1[2])
```

```
4
```

Q9) Given below is a list of lists of athletes. Create a list, t, that saves only the athlete's name if it contains the letter "t". If it does not contain the letter "t", save the athlete name into list other.

[9]:

```
athletes = [['Phelps', 'Lochte', 'Schooling', 'Ledecky', 'Franklin'],
            ['Felix', 'Bolt', 'Gardner', 'Eaton'],
            ['Biles', 'Douglas', 'Hamm', 'Raisman', 'Mikulak', 'Dalton']]

print(len(athletes)) # the inner list dont have the same structure
t = []
other = []

for inner_l1 in athletes:
    for name in inner_l1:
        if 't' in name:
            t.append(name)
        else:
            other.append(name)
```

```
3
```

[ ]: