course_3_assessment_1

Due: 2018-11-25 01:35:00

Description: Asssement for the Nested Data and Nested Iteration lesson.

Score: 0 of 9 = 0.0%

Questions

Not yet graded

The variable `nested` contains a nested list. Assign ‘snake’ to the variable `output` using indexing.

Save & Run

11/10/2020, 12:34:42 AM - 2 of 2

Show in CodeLens

```

nested = [['dog', 'cat', 'horse'], ['frog', 'turtle', 'snake', 'gecko'], ['hamster', 'gerb
output = nested[1][2]

```

ActiveCode (ac17_6_1)

Result	Actual Value	Expected Value	Notes
Pass	'snake'	'snake'	Testing that output is assigned to correct value

You passed: 100.0% of the tests

Not yet graded

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

Save & Run 11/10/2020, 12:36:23 AM - 2 of 2 Show in CodeLens

```

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

#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]


```

ActiveCode (ac17_6_2)

Result	Actual Value	Expected Value	Notes
Pass	True	True	Testing that yellow is assigned to correct value
Pass	False	False	Testing that four is assigned to correct value
Pass	True	True	Testing that orange is assigned to correct value

You passed: 100.0% of the tests

Not yet graded

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

Save & Run 11/10/2020, 12:37:36 AM - 2 of 2 Show in CodeLens

```

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]


```

ActiveCode (ac17_6_3)

Result	Actual Value	Expected Value	Notes
Pass	False	False	Testing that test1 has the correct value.
Pass	True	True	Testing that test2 has the correct value.
Pass	True	True	Testing that test3 has the correct value.

You passed: 100.0% of the tests

Not yet graded

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

Save & Run   11/10/2020, 12:41:13 AM - 3 of 3   Show in CodeLens

```

nested = {'data': ['finding', 23, ['exercises', 'hangout', 34]], 'window': ['part', 'whe

# Check to see if the string data is a key in nested, if it is, assign True to the varia
data = 'data' in nested

# Check to see if the integer 24 is in the value of the key data, if it is then assign t
twentyfour = 24 in nested

# Check to see that the string 'whole' is not in the value of the key window. If it's no
whole = not 'whole' in nested['window']

# Check to see if the string 'physics' is a key in the dictionary nested. If it is, assi
physics = 'physics' in nested
```

ActiveCode (ac17_6_4)

Result	Actual Value	Expected Value	Notes
Pass	False	False	Testing that physics has the correct value.
Pass	True	True	Testing that data has the correct value.
Pass	False	False	Testing that whole has the correct value.
Pass	False	False	Testing that twentyfour has the correct value.

You passed: 100.0% of the tests

Not yet graded

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.

Save & Run 11/10/2020, 12:43:24 AM - 3 of 3 Show in CodeLens

```

nested_d = {'Beijing':{'China':51, 'USA':36, 'Russia':22, 'Great Britain':19}, 'London':{'
london_gold = nested_d['London']['Great Britain']

```

ActiveCode (ac17_6_5)

Result	Actual Value	Expected Value	Notes
Pass	29	29	Testing that london_gold is assigned to correct value

You passed: 100.0% of the tests

Not yet graded

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

Save & Run 11/10/2020, 12:45:53 AM - 2 of 2 Show in CodeLens

```

sports = {'swimming': ['butterfly', 'breaststroke', 'backstroke', 'freestyle'], 'diving
```

```

# 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

```

ActiveCode (ac17_6_6)

Result	Actual Value	Expected Value	Notes	
Pass	'backstroke'	'backstroke'	Testing that v1 was created correctly.	
Pass	"v1 = ...roke'"	'\nspor... [-1]\n'	Testing your code (Don't worry about actual and expected values).	Expand Differences
Pass	'v1 = ...roke"'	'\nspor... [-1]\n'	Testing your code (Don't worry about actual and expected values).	Expand Differences
Pass	'platform'	'platform'	Testing that v2 was created correctly.	
Pass	'v2 = "platform"'	'\nspor... [-1]\n'	Testing your code (Don't worry about actual and expected values).	Expand Differences
Pass	"v2 = 'platform'"	'\nspor... [-1]\n'	Testing your code (Don't worry about actual and expected values).	Expand Differences
Pass	['vau...eam']	['vau...eam']	Testing that v3 was created correctly.	Expand Differences
Pass	"v3 = ...eam']"	'\nspor... [-1]\n'	Testing your code (Don't worry about actual and expected values).	Expand Differences
Pass	'rings'	'rings'	Testing that v4 was created correctly.	
Pass	"v4 = 'rings'"	'\nspor... [-1]\n'	Testing your code (Don't worry about actual and expected values).	Expand Differences
Pass	'v4 = "rings"'	'\nspor... [-1]\n'	Testing your code (Don't worry about actual and expected values).	Expand Differences

You passed: 100.0% of the tests

Not yet graded

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

Save & Run 11/10/2020, 12:47:04 AM - 2 of 2 Show in CodeLens

```

nested_d = {'Beijing':{'China':51, 'USA':36, 'Russia':22, 'Great Britain':19}, 'London':{'

US_count = [nested_d[k]['USA'] for k in nested_d]



```

ActiveCode (ac17_6_7)

Result	Actual Value	Expected Value	Notes
Pass	[35, 36, 46]	[35, 36, 46]	Testing that US_count is assigned to correct values.

You passed: 100.0% of the tests

Not yet graded

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

Save & Run 11/10/2020, 12:48:17 AM - 2 of 2 Show in CodeLens

```

l_of_l = [['purple', 'mauve', 'blue'], ['red', 'maroon', 'blood orange', 'crimson'], ['sea

third = list(map(lambda x: x[2], l_of_l))
```

ActiveCode (ac17_6_8)

Result	Actual Value	Expected Value	Notes	
Pass	['blu...ese']	['blu...ese']	Testing that third has the correct list assigned to it.	Expand Differences

You passed: 100.0% of the tests

Not yet graded

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`.

Save & Run   11/10/2020, 12:53:36 AM - 7 of 7   Show in CodeLens

```

athletes = [['Phelps', 'Lochte', 'Schooling', 'Ledecky', 'Franklin'], ['Felix', 'Bolt', 'G
athletes = sum(athletes, [])
t = list(filter(lambda x: ('t' in x), athletes))
other = list(filter(lambda x: (not 't' in x), athletes))

```

ActiveCode (ac17_6_9)

Result	Actual Value	Expected Value	Notes	
Pass	['Loc...ton']	['Loc...ton']	Testing that t is assigned to correct values.	Expand Differences
Pass	['Phe...lak']	['Phe...lak']	Testing that other is assigned to correct values.	Expand Differences

You passed: 100.0% of the tests

Score Me