PYTHON IN 10 VIDEOS

Assignment - 4

1. A lambda function that adds 10 to the number passed in as an argument, and print the result:

https://www.w3resource.com/python-exercises/lambda/python-lambda-exercise-1.php

2. Add Two Lists Using map and lambda

```
list1= [1,2,3,4,5]

List2 = [11,12,13,14,15]
```

Ans:
https://www.w3resource.com/python-exercises/lambda/python-lambda-exercise-15.php#:~:text=%23%20Create%20two%20lists%20'nums1','%20%23%20Use%20the%20'map(')

3. Define a function that takes a list of mixed data types (integers, strings, and floats) and returns three lists: one containing all the integers, one containing all the strings, and one containing all the floats. Test with different inputs.

[ ]:
```python
def separate_types(lst):
    ints, strs, floats = [], [], []
    for item in lst:
        if isinstance(item, int):
            ints.append(item)
        elif isinstance(item, str):
            strs.append(item)
        elif isinstance(item, float):
            floats.append(item)
    return ints, strs, floats

# Test
print(separate_types([1, 'a', 2.5, 3, 'b', 4.0, 'c']))  # ([1, 3], ['a', 'b', 'c'], [2.5, 4.0])
```

4. Define a function that returns another function. The returned function should take an integer and return its square. Test the returned function with different inputs.

```python
def outer_function():
    def inner_function(x):
        return x ** 2
    return inner_function

# Test
square = outer_function()
print(square(2))  # 4
print(square(5))  # 25
```

5. Filter the array, and return a new array with only the values equal to or above 18:

https://www.w3schools.com/python/ref_func_filter.asp

6. Define a function that takes two arguments, a and b, where b is a dictionary with a default value of an empty dictionary. The function should add a new key-value pair to the dictionary and return it. Test the function with different inputs.

```
def add_to_dict(a, b=None):
    if b is None:
        b = {}
    b[a] = a**2
    return b

# Test
print(add_to_dict(2))  # {2: 4}
print(add_to_dict(3, {1: 1}))  # {1: 1, 3: 9}
```