assignment2

May 11, 2020

Task 1:

1.1 Write a Python Program to implement your own myreduce() function which works exactly like Python’s built-in function reduce()

[15]:
```python
def myreduce(f, data, initial=None):
    r = 0
    if initial != None:
        r = f(r, d)
    for d in data:
        r = f(r, d)
    return r
```

[16]:
```python
print(myreduce(lambda x, y: x+y, [1, 2, 3, 4, 5]))
import functools
print(functools.reduce(lambda x, y: x+y, [1, 2, 3, 4, 5]))
```

```
15
15
```

1.2 Write a Python program to implement your own myfilter() function which works exactly like Python’s built-in function filter()

[17]:
```python
def myfilter(f, data):
    for d in data:
        if f == None:
            if d:
                yield d
        else:
            if f(d):
                yield d
```

[19]:
```python
list(myfilter(lambda x: x >3, [1,2,3,4,5,6]))
```

[19]:
```
[4, 5, 6]
```

2. Implement List comprehensions to produce the following lists. Write List comprehensions to produce the following Lists ['A', 'C', 'A', 'D', 'G', 'I', ’L’, ‘ D’] ['x', 'xx', 'xxx', 'xxxx', 'y', 'yy', 'yyy', 'yyyy', 'z', 'zz', 'zzz', 'zzzz'] ['x', 'y', 'z', 'xx', 'yy', 'zz', 'xx', 'yy', 'zz', 'xxxx', 'yyyy',

```
'zzzz'] <- typo [[2], [3], [4], [3], [4], [5], [4], [5], [6]] [[2, 3, 4, 5], [3,
4, 5, 6], [4, 5, 6, 7], [5, 6, 7, 8]] [(1, 1), (2, 1), (3, 1), (1, 2), (2, 2),
(3, 2), (1, 3), (2, 3), (3, 3)]
```

[46]:

```python
print([c for c in 'ACADGILD'])
print([c * i for c in 'xyz' for i in range(1, 5)])
print([c * i for i in range(1, 5) for c in 'xyz'])
print([[j] for i in range(2, 5) for j in range(i, i + 3)])
print([list(range(i, i + 4)) for i in range(2, 6)])
print([(j, i) for i in range(1, 4) for j in range(1, 4)])
```

```
['A', 'C', 'A', 'D', 'G', 'I', 'L', 'D']
['x', 'xx', 'xxx', 'xxxx', 'y', 'yy', 'yyy', 'yyyy', 'z', 'zz', 'zzz', 'zzzz']
['x', 'y', 'z', 'xx', 'yy', 'zz', 'xxx', 'yyy', 'zzz', 'xxxx', 'yyyy', 'zzzz']
[[2], [3], [4], [3], [4], [5], [4], [5], [6]]
[[2, 3, 4, 5], [3, 4, 5, 6], [4, 5, 6, 7], [5, 6, 7, 8]]
[(1, 1), (2, 1), (3, 1), (1, 2), (2, 2), (3, 2), (1, 3), (2, 3), (3, 3)]
```

3. Implement a function longestWord() that takes a list of words and returns the longest one.

[47]:

```python
def longestWord(words):

    longest_word = ""

    for w in words:
        if len(w) > len(longest_word):
            longest_word = w
    return longest_word
```

[48]:

```python
print(longestWord(['1','12345','word', 'words']))
```

```
12345
```

Task 2:

```
1.1 Write a Python Program(with class concepts) to find the area of the triangle using the bel
area = (s*(s-a)*(s-b)*(s-c)) ** 0.5
Function to take the length of the sides of triangle from user should be defined in the parent
```

[3]:

```python
class Poligon():
    def __init__(self, sides=None):
        self.sides = sides

    def read_side_values(self):
        if self.sides:
            self.values = []
            for i in range(self.sides):
                value = int(input(f"side {i}:"))
                self.values.append(value)
```

```
class Triangle(Poligon):
    sides = 3

    def __init__(self):
        pass

    def area(self):
        s = sum(self.values)/2
        print(self.values)
        a = (s * (s - self.values[0]) * (s - self.values[1]) * (s - self.
 ↪values[2])) ** 0.5
        return a
```

[6]:

```
t = Triangle()
t.read_side_values()
area = t.area()
print(area)
```

```
[5, 5, 5]
10.825317547305483
```

1.2 Write a function filter_long_words() that takes a list of words and an integer n and retur
of words that are longer than n.

[8]:

```
def filter_long_words(words, n):
    filtered_words = filter(lambda w:len(w) > n, words)
    return filtered_words
```

[10]:

```
list(filter_long_words(["a","abc", "1223", "12345"], 3))
```

[10]:

```
['1223', '12345']
```

2.1 Write a Python program using function concept that maps list of words into a list of intege
representing the lengths of the corresponding words .
Hint: If a list [ab,cde,erty] is passed on to the python function output should come as [2,3
Here 2,3 and 4 are the lengths of the words in the list.

[12]:

```
words = ["ab", "cde", "erty"]
list(map(len,words))
```

[12]:

```
[2, 3, 4]
```

2.2 Write a Python function which takes a character (i.e. a string of length 1) and returns Tr
a vowel, False otherwise.

[16]:

```
def is_vowel(vowel):
    return vowel in "aeiou"
```

[17]:

```
print(is_vowel('a'))
print(is_vowel('b'))
```

```
True
False
```