

Class 14 - Worksheet

Upcoming Schedule

Since we didn't get as far in class as expect, [Project 3](#) is extended to **Monday, 29 February**. Before submitting Project 3 you must have complete the Orange Belt (either by getting a "Gold star" on [Project 2](#) or completing the promotion requirements from [Class 13 notes](#)).

Slides

Nested Lists

```
def list_copy(lst):
    """Returns a shallow copy of the input lst."""
    if not lst:
        return []
    else:
        return [lst[0]] + list_copy(lst[1:])

def is_list(obj):
    """Returns True iff obj is a list."""
    return isinstance(obj, list)

def list_deep_copy(lst):
    """Returns a deep copy of the input list."""
    if not lst:
        return []
    elif is_list(lst[0]):
        return [list_deep_copy(lst[0])] + list_deep_copy(lst[1:])
    else:
        return [lst[0]] + list_deep_copy(lst[1:])
```

Demonstrate the difference between `list_copy` and `list_deep_copy` with some Python code that shows how they behave differently.

```
def list_deep_map(fn, lst):
    """
    Returns a new list with the same structure as lst, but with each element
    replaced by the result of applying fn to that element.
    """
    if not lst:
        return []
    elif is_list(lst[0]):
        # finish this code
        #
    else:
        #
        #

def list_accumulate(fn, lst, start):

def list_sum(lst):
    def
    #
    return list_accumulate(_____, lst, _____)

def list_length(lst):
    def
    #
    return list_accumulate(_____, lst, _____)

def list_and(lst):

def list_or(lst):

def list_biggest(lst):

def list_deep_accumulate(lst, fn, base):
```