2.3 **Tutorial:** Write a function that takes in a sequence `s` and a function `fn` and returns a dictionary.

The values of the dictionary are lists of elements from `s`. Each element `e` in a list should be constructed such that `fn(e)` is the same for all elements in that list. The key for each value should be `fn(e)`. For each element `e` in `s`, check the value that calling `fn(e)` returns, and add `e` to the corresponding group.

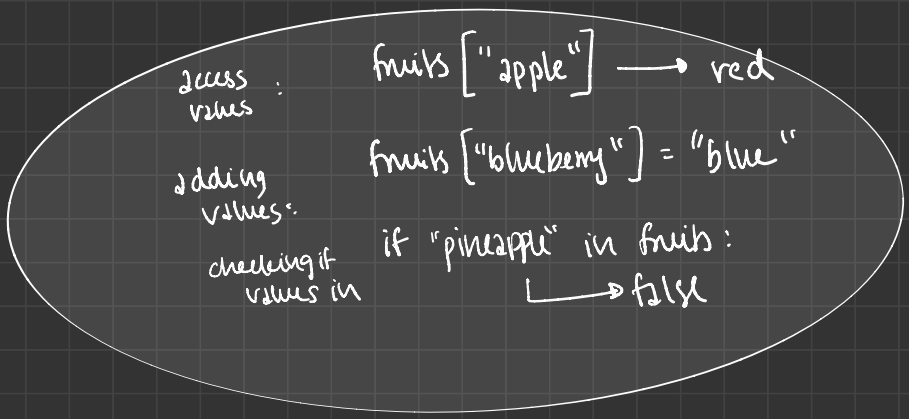
```
def group_by(s, fn):
    """
    >>> group_by([12, 23, 14, 45], lambda p: p // 10)
    {1: [12, 14], 2: [23], 4: [45]}
    >>> group_by(range(-3, 4), lambda x: x * x)
    {0: [0], 1: [-1, 1], 4: [-2, 2], 9: [-3, 3]}
    """
    grouped = {}
    for _____i in s_____:
        key = _____fn(i)_____
        if _____key in grouped_____:
            _____grouped[key].append(i)_____
        else:
            grouped[key] = _____[i]_____
    return _____grouped_____
```

dictionaries:

dict = { }

key : value

fruits = { "banana" : "yellow",
"apple" : "red"
"orange" : "orange" }



colors = { "red" : ["apples", "strawberries"]
"yellow" : ["pineapple", "banana"] }

colors ["red"].append ("pomegranate")

2.4 **Tutorial:** Write a function that takes in a value `x`, a value `el`, and a list `s` and adds as many `el`'s to the end of the list as there are `x`'s. **Make sure to modify the original list using list mutation techniques.**

```
def add_this_many(x, el, s):
    """ Adds el to the end of s the number of times x occurs
    in s.
    >>> s = [1, 2, 4, 2, 1]
    >>> add_this_many(1, 5, s)
    >>> s
    [1, 2, 4, 2, 1, 5, 5]
    >>> add_this_many(2, 2, s)
    >>> s
    [1, 2, 4, 2, 1, 5, 5, 2, 2]
    """
```