

```

class PollRecord:

    def __init__(self, name: str, options: list[str]):
        self.name = name          # Set up a field, assign a value to it
        self.options = options
        self.votes = list()      # Make a new field with an empty list

    def cast_vote(self, for_option):
        if for_option in self.options:  # is for_option contained in self.options
            self.votes.append(for_option)
        else:
            raise ValueError("Wrong option " + for_option)

    def count_votes(self):
        return len(self.votes)

    def count_votes_for(self, for_option: str) -> int:
        total = 0

        # "for each loop"
        for v in self.votes:
            if v == for_option:
                total += 1

        return total

    # Like Java's toString
    def __str__(self) -> str:
        return "Poll for question " + self.name + ", choices: " + str(self.options)

option_list = ["red", "green", "blue"]
p = PollRecord("What is your favorite color?", option_list)
p.cast_vote("red")
p.cast_vote("green")

```

Another way

```
def cast_vote(poll, option_picked: str):
    if option_picked not in poll["options"]:
        raise ValueError("Invalid option")

    poll["votes"].append(option_picked)

def total_votes(poll):
    return len(poll["votes"])

def count_votes_for(poll, for_option):
    total = 0

    for v in poll["votes"]:
        if v == for_option:
            total += 1

    return total

some_poll = {
    "name": "what is your favorite color?",
    "options": ["red", "green", "blue"],
    "votes": [],
}

cast_vote(some_poll, "red")
cast_vote(some_poll, "green")
print(total_votes(some_poll))
```

Bonus: list comprehensions

Map-like list comprehension

Transform ["8", "10", "-5.5"] to [8, 10, 5.5]

Transform ["hello", "WORLD", "pYthon"] to ["hello", "world", "python"]

Filter-like list comprehension

Transform [0, 5, -2, -4, 7] to [5, 7]

Combine map and filter

Transform ["pvd", "boston", "wos", "newport"] to ["PVD", "WOS"]

Combine map, filter, and enumerate

Transform [8, 3, 5, 9, 0, 2] to [0, 4, 5] (indices of the even numbers)