

## Student

### Attributes:

- Group: the group where the student is
- Name (str): student's name
- Partner\_ratings (lst of str): list of partner names from how the student prefers them
- Partner: student's current partner
- To\_propose (lst of str): name of partners to propose

### Methods:

- `__init__(self, group, name, partner_ratings, partner) = none`
- `__str__(self) = student and partner`
- `has_partner(self) = checks if student has a partner`
- `get_rating_of_name(self, name) = returns how much the student wants to work with the name given`
- `break_partnership(self) = break partnership between the partners`
- `propose_to_top_choice(self) = propose partnership with current top choice`

## Group

### Attributes:

- `students_a(lst of student): students in group a`
- `students_b(lst of student): students in group b`
- `all_students(lst of student): all students`

### Methods:

- `__init__(self, names_a, names_b) = initializes group`
- `get_student_by_name(self, name) = finds a student in the group by name, returns None if not found`
- `set_ratings(self, ratings) = sets preferences for each student based on name`
- `break_all_partnerships(self) = break all partnerships within group`
- `make_gale_shapely_partnerships(self) = forms partnerships Using the Gale Shapely algorithm`
- `print_student_information(self) = prints information about all students in the group`