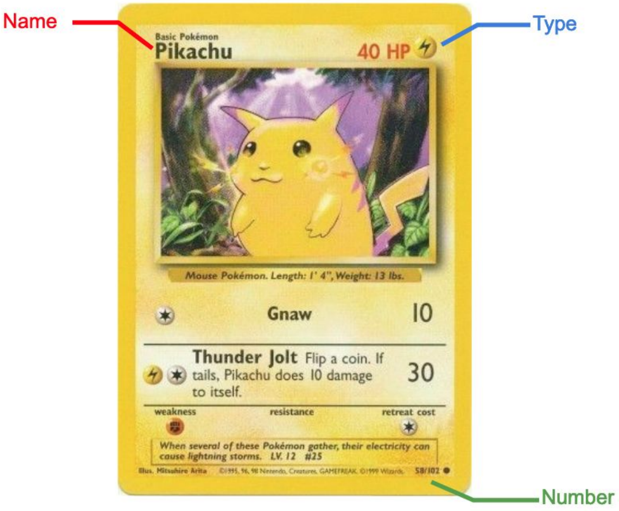
Problem Solving Session

- The remainder of today's class will comprise the ***problem solving session*** (***PSS***).
- Your instructor will divide you into ***teams of 3 or 4 students***.
- Each team will ***work together*** to solve the following problems over the course of ***20-30 minutes***.
  - You may work on paper, a white board, or digitally as determined by your instructor.
  - You will submit your solution by pushing it to GitHub before the end of class.
- Your instructor will go over the solution before the end of class.
- If there is any time remaining, you will begin work on your homework assignment.

Class participation is a significant part of your grade (20%). This includes in class activities and the problem solving session.

Your Course Assistants will grade your participation by verifying that you pushed your solutions before the end of the class period each day.

Problem Solving 1



Name
Type
Basic Pokémon
Pikachu
40 HP
Mouse Pokémon. Length: 1' 4", Weight: 13 lbs.
Gnaw 10
Thunder Jolt Flip a coin. If tails, Pikachu does 10 damage to itself. 30
weakness
resistance
retreat cost
When several of these Pokémon gather, their electricity can cause lightning storms. LV. 12 #25
Illus. Mitsuhiro Arita
58/102
Number

Every Pokémon card has three basic attributes: a number, name, and type. For example: `number=58`, `name="Pikachu"`, `type="Electric"`. Use this information to answer the questions to the right.

How will you represent a Pokémon card in Python?

A tuple

Write the Python code to make a card with the values given to the left.

```
pokemon_card = (58, “Pikachu”, “Electric”)
```

Assuming that you need to quickly find a card using only its number, what data structure will you use to store them?

A dictionary

Write the code to create your data structure and add three different Pokémon cards to it. Use whatever numbers, names, and types you'd like.

```
collector_deck = {}
collector_deck[58] = (“Pikachu”, “Electric”)
collector_deck[1] = (“Charmander”, “Fire”)
collector_deck[2] = (“Squirtle”, “Water”)
```

Problem Solving 2

Pokémon cards are sold in packs of 10. For the purposes of this exercise we will assume that a pack of cards does not contain more than one of each card and that the cards may be in arbitrary order in the pack.

What data structure will you use to represent a pack of Pokémon cards?

List that stores tuples if cards can be duplicates.
If cards may not duplicate, a set that contains tuples.

Write a function named `make_pack` **that declares a parameter for a** `database` **of Pokémon cards - the database is a dictionary where each key is a card number and the corresponding value is a card. Your function should create and return a "pack" of 10 unique cards randomly selected from the database.**

```
import random

def make_pack(database):
        pack = set()

        return pack
```

Problem Solving 3

The goal of any serious Pokémon card collector is to acquire a complete set of cards. Some collector's keep their set in a binder. As they buy packs of cards, they add any missing cards to the binder. Duplicates can be discarded or traded later.

A collector's binder contains one of each card that they have obtained so far. Which data structure is best to use to keep track of the cards that have been collected?

A dictionary

Define a function named `build_collection`that declares a parameter for a `database` of cards and uses a loop to "buy" ten packs of cards and add them to the collection. Hint: use your `make_pack` function to create packs to buy.

```
def build_collection(database):
```

Problem Solving 4

The function that you wrote in the previous activity only keeps track of which cards are in the collection, but not *how many* of each have been collected. A typical collector gets *lots* of duplicate cards before completing a set. For example, they may end up with 10 Pikachu cards before getting the last Charizard that needed to complete the collection.

Write a modified version of the function that keeps track of not only which cards have been acquired, but how many of each card the collector has after opening the 10 packs. What data structure will you use?