Data Structures | *Hashmap*

Hashmap

- Collection that stores data as key-value pairs
    - Data is located using the “key”
    - The data is the “value”
- Similar to definitions in a dictionary
- Very fast to retrieve data using the key

Example: find data

```
let mut people = HashMap::new();
people.insert("Susan", 21);
people.insert("Ed", 13);
people.insert("Will", 14);
people.insert("Cathy", 22);
people.remove("Susan");

match people.get("Ed") {
    Some(age) => println!("age = {:?}", age),
    None => println!("not found"),
}
```

Example: iterate

```
for (person, age) in people.iter() {
    println!("person = {:?}, age = {:?}", person, age);
}

for person in people.keys() {
    println!("person = {:?}", person);
}

for age in people.values() {
    println!("age = {:?}", age);
}
```

Recap

- Store information as key-value pairs
  - “Key” is used to access the “value”
- Very fast to insert & find data using the key
- Useful when you need to find information and know exactly where it is (via the key)