10 - Dictionaries Examples

September 30, 2022

0.1 Dictionary Examples

Ordinals

Make a dictionary that maps numbers to their ordinal words (first, second, third, etc.) for the numbers `1` through `3`. The keys should be integers and the values should be strings.

[4]:
```
#RUN THIS BLOCK
ordinals = {1 : "first", 2 : "second", 3 : "third"}
```

Print the ordinal for 3.

[7]:
```
print(ordinals[3])
```

```
third
```

Change the existing values for the keys `1`, `2`, `3` to `"1st"`, `"2nd"`, `"3rd"`.

[10]:
```
ordinals[1] = "1st"
ordinals[2] = "2nd"
ordinals[3] = "3rd"
```

Add an entry for `4`.

[11]:
```
ordinals[4] = "4th"

print(ordinals)
```

```
{1: '1st', 2: '2nd', 3: '3rd', 4: '4th'}
```

Bank accounts

For the next part, use the dictionary representing bank account balances below.

[3]:
```
#RUN THIS BLOCK
accounts = {"Sofia": 525, "Dave": 200, "Charlie": 315, "Doug": 600}
```

Write a function `has_account` that takes a person's name (a string) as a parameter and checks if that person has an account in the `accounts` dictionary.

[5]:
```
def has_account(s):
    return s in accounts

print(has_account("Sofia"))
print(has_account("Jimmy"))
```

```
True
False
```

Write a function `get_balance` that takes a person's name (a string) as a parameter and prints the balance for that person.

If the person does not have an account, it should print `"No account found."`

[6]:
```
def get_balance(s):
    if has_account(s):
        print(s + " has an account balace of " + str(accounts[s]))
    else:
        print("No account found.")

get_balance("Sofia")
get_balance("Jimmy")
```

```
Sofia has an account balace of 525
No account found.
```

Write two functions `withdraw` and `deposit` that take a person's name (a string) as a parameter as well as an integer value. `withdraw` should decrease the person's balance by the given amount and print the new balance, while `deposit` should increase the person's balance and print the new balance.

Note: these functions should change the entry in the dictionary (this is called a side-effect of the function.)

If the person does not have an account, it should print `"No account found."`

[9]:
```
def withdraw(s, n):
    if has_account(s):
        accounts[s] -= n
        get_balance(s)
    else:
        print("No account found.")

withdraw("Sofia", 100)
withdraw("Jimmy", 100)


def deposit(s, n):
    if has_account(s):
        accounts[s] += n
```

```
        get_balance(s)
    else:
        print("No account found.")

deposit("Sofia", 100)
deposit("Jimmy", 50)
```

```
Sofia has an account balace of 425
No account found.
Sofia has an account balace of 525
No account found.
```

0.2 Interactive Phone Book

- Ask user for a name, then a phone number, and add them to a dictionary.
- When the user enters an empty string for a name, stop and return the dictionary
  - It should not enter that empty string into the dictionary

[ ]: