Introduction to Python

Fondren Library
Research Data Services

DICTIONARIES: 5 MINS

- Dictionaries can store lots of information about the object, easily:

```
patient_3 = {
    'name': 'jack',
    'last_name': 'sparrow',
    'mrn': '001234567',
    'dob': '02231845'
}
print(patient_3)
```

```
{'name': 'jack', 'last_name': 'sparrow', 'mrn': '001234567', 'dob': '0223
1845'}
```

- We can then access elements of our dictionary.

```
print("Patient_3's DOB is " + patient_3['dob'])
```

```
Patient_3's DOB is 02231845
```

- How is this format useful vs. the previous format?

```
student = {
    "name" : "Monica Dewey",
    "age" : 34,
    "courses" : ["Python", "Tableau"]
}
print(student)

print(student["name"])
print(student["age"])
```

```
book_info = {
    "title": "To Kill a Mockingbird",
    "author": "Harper Lee",
    "published_year": 1960,
    "genres": ["Fiction", "Drama"],
    "ratings": { "Goodreads": 4.28, "Amazon":
4.8 },
    "availability": { "hardcover": True,
"paperback": True, "ebook": True,
"audiobook": False }
    } # Accessing elements in the dictionary

print(f"Title : {book_info['title']}")
print(f"Goodreads Rating:
{book_info['ratings']['Goodreads']}")
```

LOOPING

- Previously when we spoke about looping through lists, we mentioned that loops can work with a few lines of data to millions of rows of data. The same applies here.

```
patient_3 = {
    'name': 'jack',
    'last_name': 'sparrow',
    'mrn': '001234567',
    'dob': '02231845'
}
```

- Looping in Python allows you to loop through a dictionary. (next slide!)

LOOPING

- First you create a **for** loop for your dictionary.
  - Your loop will contain the two variables – key and value for each pair.
  - You can choose any names you want for key and value.
- The next part includes the name of the dictionary, plus the items() method – which returns a list of key-value pairs.

```
for key, value in patient_3.items():
    print("\nKey: " + key)
    print("\nValue: " + value)
```

```
Key: name

Value: jack

Key: last_name

Value: sparrow

Key: mrn

Value: 001234567

Key: dob

Value: 02231845
```

```
patient_3 = {
    "name" : "jack" ,
    "last_name" : "sparrow" ,
    "mrn" : "001234567" ,
    "dob" : "02231845" ,
}

for key, value in patient_3.items():
    print("\nKey: " + key)
    print("\nValue: " + value)
```

```
students_grades = {
  "JC" : {"Math" : 95, "Science" :
100, "English" : 98},
  "Michelle" : {"Math" : 95,
"Science" : 100, "English" : 98},
  "Monica" : {"Math" : 95,
"Science" : 100, "English" : 98},
  "Neha" : {"Math" : 95, "Science"
: 100, "English" : 98}
}
for student, grades in
students_grades.items():
  print(f"Student: {student}")
  for subject, grades in
grades.items():
      print(f" {subject}:
{grades}")
print()
```

KEYS ONLY

- The keys() method is useful for when you only need to use the key portion of your dictionary.

```
for keys in patient_3.keys():
    print(keys)
```

```
name
last_name
mrn
dob
```

KEYS

- Lets create a new dictionary to work with keys

```
friends_ages = {
    'nick': '30',
    'amanda': '34',
    'john': '36',
    'haley':'26'
}
```

```
fav_friends = ['nick', 'amanda']
for name in friends_ages.keys():
    print(name.title())

    if name in fav_friends:
        print("Hi " + name.title() + ", I see you're " +
              friends_ages[name])
```

```
Nick
Hi Nick, I see you're 30
Amanda
Hi Amanda, I see you're 34
John
Haley
```

```
employees = {
    "John" : {"age" : 28, "department" : "HR", "salary" : 50000},
    "Alice" : {"age" : 34, "department" : "IT", "salary" : 75000},
    "Bob" : {"age" : 61, "department" : "Finance", "salary" : 120000}
}

print("Employee Names")
for name in employees.keys():
    print(name)
```

WHILE LOOPS

- While loops are used to search through data; while a certain condition is True. For example:

```
number = 1
while number <= 5:
    print(number)
    number += 1
```

```
1
2
3
4
5
```

WHILE LOOPS

- The first line we start our counting from the number = 1.
- The while loop is set to keep running while a certain condition is true. In this case, while number is <= 5.
- The code inside the loop prints the current number (print(number)) then increases the number by 1 (+= 1). += basically means: number = number + 1.
- Python repeats the loop until the condition is no longer true.

```
number = 1
while number <= 5:
    print(number)
    number += 1
```

```
1
2
3
4
5
```

USER INPUT

- Users are also able to interact with the program you are creating. Python can take an input from a user using the **input()** function.

```
message = input("Please enter your first name ")
print(message)
```

```
Please enter your first name
```

```
age = input("Please enter your age ")
print(age)
```

```
Please enter your age
```

WHILE LOOPS AND USER INPUT

- A while loop can be broken by a users input.

```
prompt = "Input a message. I will repeat it until you type 'quit'"
message = ""

while message != 'quit':
    message = input(prompt)
    print(message)
```

Strings + While Loops:

```
total_sum = 0
user_input = ""

# Loop until the user enters 'stop'
while user_input.lower() != 'stop':
    user_input = input("Enter a number to add to the sum, or type 'stop' to finish: ")
    if user_input.lower() != 'stop':
        try:
            number = float(user_input)
            total_sum += number
        except ValueError:
            print("Invalid input. Please enter a valid number:")

print(f"The total sum is: {total_sum}")
```

Dictionaries + While Loops:

```
# Dictionary to store country-capitals
country_capitals = {
    "USA" : "Washington, DC",
    "France" : "Paris",
    "Japan" : "Tokyo",
    "Germany" : "Berlin",
    "India" : "New Delhi"
    }

# Flag to control the loop
continue_game = True

print("Welcome to the Country-Capital Quiz Game!")
print("Type 'exit' to stop the game at any Time.")

while continue_game:
    country = input("Enter the name of a country: ").strip()

    if country.lower() == "exit":
        continue_game = False
        print("Thanks for Playing!")

    elif country in country_capitals:

        capital_guess = input(f"What is the capital of {country}?").strip()

    if capital_guess.lower() == country_capitals[country].lower():
        print("Correct!")

    else:
        print(f"Wrong! the capital of {country} is
```

Office Hours - 8PM to 9PM