Module 2 Practice - Python Fundamentals

Lists

Student will be able to

- Create Lists
- Access items in a list
- Add Items to the end of a list
- Insert items into a list
- Delete items from a list

Create Lists

In [ ]:
```
# [ ] create and populate list called days_of_week then print it
```

In [ ]:
```
# [ ] after days_of_week is run above, print the days in the list at odd indexes 1,
```

Phone letters

 phone keys: number and letters Create a list, `phone_letters`, where the index 0 - 9 contains the letters for keys 0 - 9.

- 0 = ' ' (a space)
- 1 = '' (empty)
- 2 = 'ABC'
- 3 = 'DEF'
- etc...

In [ ]:
```
# [ ] create and populate list called phone_letters then print it
```

Access Lists

for the 2 cells below

- Use days_of_week list created above
- Run the cell above to make the list available

In [ ]:
```
# [ ] create a variable: day, assign day to "Tuesday" using days_of_week[]
# [ ] print day variable
```

In [ ]:
```
# PART 2
# [ ] assign day to days_of_week index = 5
# [ ] print day
```

Append and Insert items into a list

Endsday, Midsday, Resterday

for the exercises below

- Use days_of_week list created above
- Run the cell defining days_of_week above to make the list available

In [ ]:
```
# [ ] Make up a new day! - append an 8th day of the week to days_of_week
# [ ] print days_of_week
```

Question

- What happens if you keep running the cell above?
- How can you return to the initial state with the regular 7 days in days_of_week?

In [ ]:
```
# [ ] Make up another new day - insert a new day into the middle of days_of_week be
# [ ] print days_of_week
```

In [ ]:
```
# [ ]  Extend the weekend - insert a day between Fri & Sat in the days_of_week list
# [ ] print days_of_week
```

Delete from a list

`del` & `.pop()` some bad ideas

exercises below assume days_of_week appended/inserted 3 extra days in previous exercises

In [ ]:
```
# [ ] print days_of_week
# [ ] modified week is too long - pop() the last index of days_of_week & print .pop
# [ ] print days_of_week
```

In [ ]:
```
# [ ] print days_of_week
# [ ] delete (del) the new day added to the middle of the week
# [ ] print days_of_week
```

In [ ]:
```
# [ ] print days_of_week
# [ ] programmers choice - pop() any day in days_of week & print .pop() value
# [ ] print days_of_week
```

Program: Letter to Number Function

for the exercise below

- Use phone_letters list created above
- Run the cell above to make the list available

recall unit 1 using `in` to search for a string in a string

```
if "e" in "open":
    print("e found")
else:
   print("e not found")
```

 phone keys: number and letters

create funtion let_to_num()

- let_to_num() takes input of a single letter, space or empty string stored in an argument variable: letter
  - use `while key < 10:` to try numbers 0 - 9 as index for `phone_letters` ("key" = phone dial pad key)
  - check if `letter` variable is in the index of `phone_letters[key]`

```
key = 0
while key < 10:
    if # Create Code: determine if letter is **`in`** any of the
phone_letters[key] where key is the index 0 -9:
        return key
    else:
        key = key + 1
return "Not Found"
```

- return the number or "Not Found"
- **call** let_to_num() to test the function so it prints the argument and return value with:
  - space
  - lowercase letter
  - different letter, uppercase
  - a number

Bonus: create a special case to check if empty string (`""`) was submitted
the problem is that an empty string will be found in all strings as

```
if "" in "ABC":
```

is True, and is true for any phone_letters, but should `return 1`

In [10]:
```
phone_letters = [' ', '', 'ABC', 'DEF', 'GHI', 'JKL', 'MNO', 'PQRS', 'TUV', 'WXYZ']
def let_to_num():
```

```
    letter=input("Enter single letter, space or empty: ")
    key = 0
    while key < 10:
        if letter in phone_letters[key]:
                return key
        else:
            key = key + 1
    return ("Not found")
print(let_to_num())
```

```
Enter single letter, space or empty:
0
```

In [ ]:

Challenge: reverse a string

using

- while
- .pop()
- insert()

`pop()` the **first item** in the list and

In [4]:
```
pc_parts = ["Keyboard", "Mouse,", "Computer", "Monitor"]
rev_str = []
print("List before:", pc_parts)

while pc_parts:
    rev = pc_parts.pop(0)
    rev_str.insert(0,rev)

print("Reversed list:", rev_str)
```

```
List before: ['Keyboard', 'Mouse,', 'Computer', 'Monitor']
Reversed list: ['Monitor', 'Computer', 'Mouse,', 'Keyboard']
```

In [ ]:

Terms of use Privacy & cookies © 2017 Microsoft