DSC 20
Discussion Section 3

ARDA CANKAT BATI

Piazza Poll

- Let's look at the piazza poll for today's coverage

Today's Plan

A. File Operations
B. Advanced Argument Passing
C. Discussing HW03 Questions

File Operations

open (**filename**, **mode**='r', **buffering**=-1, **encoding**=None, **errors**=None, **newline**=None, **closefd**=True, **opener**=None)

with open ("hello.txt", r) **as** file: #Read only mode
with open ("hello.txt", rb) **as** file: #Read only binary mode

The same case with w, wb

File Operations

```
with open ("hello.txt", a) as file: #append mode
with open ("hello.txt", ab) as file: #append binary mode
```

- For the append mode, the pointer is places at the end of the file, instead of the beginning as in write and read modes.
- In Python 3 mode 'r' opens the files in the text encoding provided by the user (if not provided the default encoding). read() will give you strings.
- mode 'rb' opens the files iin binary format (just bytes). read() will give a bytes object.
- Usually 'rb' and 'wb' are used for non-text files, like images, videos etc.

File Operations, some tips

```
# Reading file as a whole
with open(filepath, "r") as f:
    text = f.read()
    tex.split('\n')

# Reading file line by line
with open(filepath, "r") as f:
    lines = f.readlines()
    for i in range(len(lines)):
        print(lines[i])

# Same as above, alternative way with enumerate
with open(filepath, "r") as f:
    lines = f.readlines()
    for indx,line in enumerate(lines):
        print(line)
```

File Operations, some tips

```
line = "Sunny day in SD."

print(line)
print()
print(line.split(' '))
print()

line = "Sunny \nday \nin \nSD."
print(line.split('n'))
print()

line = "Sunny \nday \nin \nSD."
print(line.split('\n'))
```

File Operations, some tips

```
line = "Sunny day in SD."

print(line)
print()
print(line.split(' '))
print()

line = "Sunny \nday \nin \nSD."
print(line.split('n'))
print()

line = "Sunny \nday \nin \nSD."
print(line.split('\n'))
```

```
Sunny day in SD.

['Sunny', 'day', 'in', 'SD.']

['Su', '', 'y \nday \ni', ' \nSD.']

['Sunny ', 'day ', 'in ', 'SD.']
```

File Operations, some tips

```
a = '\n'
print('a = ', a, 'len(a) = ', len(a))

a = '\\n'
print('a = ', a, 'len(a) = ', len(a))
```

File Operations, some tips

```
a = '\n'
print('a = ', a, 'len(a) = ', len(a))

a = '\\n'
print('a = ', a, 'len(a) = ', len(a))
```

```
a =
 len(a) =  1
a =  \n len(a) =  2
```

File Operations, some tips

```
filepath = 'dummy.txt'

# Write Mode
with open(filepath, "w") as f:
    f.write("First line of file.\n")

# Append Mode
with open(filepath, "a") as f:
    f.write("Second line of wile.\n")
    f.write("Not overwriting anything.\n")

# Read Mode
with open(filepath, "r") as f:
    text = f.read()
    print(text)
```

```
First line of file.
Second line of wile.
Not overwriting anything.
```

Advanced Argument Passing

```
def fun(x,y=0,z=0):
    return "{0} {1} {2}".format(x, y, z)

def fun(x,y=0,z=0):
    return f"{x} {y} {z}"

print(fun(1, 2, 3))
print(fun(4, y = 5, z = 6))
```

Advanced Argument Passing

```
def fun(x,y=0,z=0):
    return "{0} {1} {2}".format(x, y, z)

def fun(x,y=0,z=0):
    return f"{x} {y} {z}"

print(fun(1, 2, 3))
print(fun(4, y = 5, z = 6))
```

```
1 2 3
4 5 6
```

Advanced Argument Passing

```
print(fun(y = 3,z = 2, 3))
```

```
print(fun(u= 5, y = 3,z = 2))
```

Advanced Argument Passing

```
print(fun(y = 3,z = 2, 3))
```

```
  File "<ipython-input-79-a25b4bd899c5>", line 1
    print(fun(y = 3,z = 2, 3))
                           ^
SyntaxError: positional argument follows keyword argument
```

```
print(fun(u= 5, y = 3,z = 2))
```

```
---------------------------------------------------------------------------
TypeError                                 Traceback (most recent call last)
<ipython-input-44-14a2053f4445> in <module>
----> 1 print(fun(u= 5, y = 3,z = 2))

TypeError: fun() got an unexpected keyword argument 'u'
```

Advanced Argument Passing

```
def func2(*args, **kwargs):

    print(type(args), args)
    res = []
    for arg in args:
        res.append(arg)

    print(type(kwargs), kwargs)
    for key in kwargs:
        res.append((key, kwargs[key])) #Appending as tuple

    return res

func2(1, 2, 3, 4, x = 5, y = 9.43, z = 'hello')
```

Advanced Argument Passing

```
def func2(*args, **kwargs):

    print(type(args), args)
    res = []
    for arg in args:
        res.append(arg)

    print(type(kwargs), kwargs)
    for key in kwargs:
        res.append((key, kwargs[key])) #Appending as tuple

    return res

func2(1, 2, 3, 4, x = 5, y = 9.43, z = 'hello')
```

```
<class 'tuple'> (1, 2, 3, 4)
<class 'dict'> {'x': 5, 'y': 9.43, 'z': 'hello'}

[1, 2, 3, 4, ('x', 5), ('y', 9.43), ('z', 'hello')]
```

HW03 Discussion

Which Questions you want to focus on from the HW?

A) Question 1: order_scores
B) Question 2: counting_spaces
C) Question 3: create_trigrams
D) Question 4: newton_sqrt
E) Question 5: list_to_pixel & pixel_to_list