

Taking inputs, Joining List, File Operations

Python Lecture 3

1

Agenda

- Taking inputs
- Converting type
- While loop
- Joining list from string
- Splitting string into list
- File operations (Read, Append, Write)

2

Taking Inputs

3

```
name = input("What is your name? ")  
print("Hello " + name)
```

4

The inputs are always string. We need to convert inputs into our data type.

5

Convert type

- `int(x)`
- `float(x)`
- `str(x)`
- `bool(x)`
- `list(x)`

6

An example of taking input and convert it into integer.

```
age = input("What is your age? ")
```

```
if int(age) >= 18:
```

```
    print("You can drink.")
```

```
else:
```

```
    print("Please don't drink.")
```

```
print("Good bye.")
```

7

While-loop

8

While loop

```
x = 0
while x < 10:
    print(x)
    x += 1
```

9

Comparing to for-loop

```
for x in range(10):
    print(x)
```

10

Take inputs until quit

```
guests = []

while True:
    value = input("Please input a guest name, or
'q' to quit: ")
    if value == "q":
        break
    guests.append(value)

print(guests)
```

11

Using while-loop

- When we don't know where the loop end.
- Beware of infinite loop.

12

Join and Split

13

Join string together

We can join the list into string for better readable text.

```
"glue".join(list)
```

14

```
", ".join(guests)
```

15

Join list together into string

```
sample_list = ["Peter", "Tom", "Viena", "John"]  
result = "The students are: {}".format( "  
".join(sample_list) )  
print( result )
```

16

Split string into list

```
sample_string = "Peter, Tom, Viena, John"  
result_list = sample_string.split(', ')  
print( result_list )
```

17

Split multiple lines

```
sample_string = """This is a sample paragraph with  
multiple lines.  
So that we can split the string by line endings.  
Good for getting list from a plain text file with  
mutliple lines."""  
result_list = sample_string.splitlines()  
print(result_list)
```

18

File Operations

19

Reading Plain Text File with `readlines`

```
with open("guests.txt", "r") as file_obj:  
    result = file_obj.readlines()  
  
print(result)
```

20

Reading Plain Text File with `read` and `splitlines`

```
with open("guests.txt", "r") as file_obj:  
    result = file_obj.read().splitlines()  
  
print(result)
```

21

Appending to Existing File

```
with open("guests2.txt", "a") as file_obj:  
    file_obj.write("New Name Here\n")
```

22

Writing to File

```
with open("guests2.txt", "w") as file_obj:  
    file_obj.write("New Name Here\n")
```

23

Writing a list into file

```
list = ["Susan", "Chris", "Anthony", "Joana"]  
with open("guests2.txt", "a") as file_obj:  
    file_obj.writelines(list)
```

24

Writing a list into file with line endings

```
list = ["Susan", "Chris", "Anthony", "Joana"]
with open("guests3.txt", "a") as file_obj:
    file_obj.write( "\n".join(list) )
```

25

Guests Example: Write Result to File

```
guests = []

while True:
    value = input("Please input a guest name, or
'q' to quit: ")
    if value == "q":
        break
    if len(value) > 0:
        guests.append(value)

with open("guests.txt", "a") as file_obj:
```

26

Guests Example: What if we let user choose where to save the list?

```
list_name = input("Please enter a list name: ")
```

27

Guests Example: Save to the given list name

```
with open(f"{list_name}.txt", "a") as file_obj:
    file_obj.write("\n".join(items))
    file_obj.write("\n")

print(f"List saved to {list_name}.txt.")
```

28

Further more: Divide logic into functions

```
def ask_for_list_name():  
    '''Ask the user for list name to save.'''  
    list_name = input("Please enter a list name: ")  
    return list_name  
  
def get_list_items_input(list_name):  
    '''Ask user to input a collection of list  
items until type 'q'. '''  
    items = []  
    while True:  
        value = input(f"Please input an item for
```