TKINTER
REGULAR EXPRESSIONS
Remaining schedule

• May 7: regular session
• May 14: first test (weeks 1-5)
• May 21: second test (weeks 6-9)
• May 28: project presentations
• June 4, 11, 18: exam
What is the type?

["key1", "value1", "key2", value2"]

A. List
B. Dictionary
C. Set
D. Tuple
E. Something else
What is the type?

["key1", "value1", "key2", value2"]

A. List
B. Dictionary
C. Set
D. Tuple
E. Something else
What is the type?

([1, 2, 3], [4, 5, 6], [7, 8, 9])

A. List
B. Dictionary
C. Set
D. Tuple
E. Something else
What is the type?

([1, 2, 3], [4, 5, 6], [7, 8, 9])

A. List
B. Dictionary
C. Set
D. Tuple
E. Something else

✓ D. Tuple
What is the type?

\{0: 1, 2: 3, 4: 5, 6: 7, 8: 9\}

A. List
B. Dictionary
C. Set
D. Tuple
E. Something else
What is the type?

{0: 1, 2: 3, 4: 5, 6: 7, 8: 9}

A. List
B. Dictionary
C. Set
D. Tuple
E. Something else

B. Dictionary
What is the type?

{"height: 162", "weight: 51", "age: 23"}

A. List
B. Dictionary
C. Set
D. Tuple
E. Something else
What is the type?

A. List
B. Dictionary
C. Set  ✔️
D. Tuple
E. Something else

{"height: 162", "weight: 51", "age: 23"}
Tkinter

Tkinter is a portable GUI library for Python.

```python
# Import the module
from tkinter import *

# Create main window
window = Tk()
window.title("Canvas")

# Add widgets

# Enter the main event loop
window.mainloop()
```
Widgets are GUI elements

- button
- canvas
- checkbutton
- entry
- frame
- label
- listbox
- menu
- menubutton
- message
- panedwindow
- radiobutton
- scale
- scrollerbar
- spinbox
- text
- tkmessagebox
- toplevel
Widgets are GUI elements

```python
from tkinter import *
from tkinter import messagebox

def sayHello():
    say = "Hello " + name.get()
    messagebox.showinfo(message = say)

window = Tk()
window.title("Say Hello")
window.geometry("300x100")

# Label
labelName = Label(window, text="Name")
labelName.place(x=5, y=5)

# Text field
name = Entry(window)
name.place(x=70, y=5, width=150)

# Button; function sayHello is associated with a button
but = Button(window, text="Say Hello!", command=sayHello)
but.place(x=70, y=40, width=150)

window.mainloop()
```
Canvas and shapes

```python
window = Tk()
area = Canvas(window, width=600, height=600)

• area.create_line(x1, y1, x2, y2, x3, y3, ...)
• area.create_rectangle(x1, y1, x2, y2)
• area.create_polygon(x1, y1, x2, y2, x3, y3, ...)
• area.create_oval(x1, y1, x2, y2)
• area.create_text(x, y, text=t)
• area.create_image(x, y, image=i)
```
Regular expressions

^ Matches the **beginning** of a line

$ Matches the **end** of the line

. Matches **any** character

\s Matches **whitespace**

\S Matches any **non-whitespace** character

* **Repeats** a character zero or more times

*? **Repeats** a character zero or more times (non-greedy)

+ **Repeats** a character one or more times

+? **Repeats** a character one or more times (non-greedy)

[aeiou] Matches a single character in the listed **set**

[^XYZ] Matches a single character **not in** the listed **set**

[a-z0-9] The set of characters can include a **range**

( Indicates where string **extraction** is to start

) Indicates where string **extraction** is to end
Examples

```
import re

x = 'From: X- Using the: character. My 2 favorite numbers are 19 and 42.'

print(re.findall('^h', x))
print(re.findall('h.*:', x))
print(re.findall('[0-9]+', x))
print(re.findall('\S+a\S', x))
print(re.findall('^F.+:', x))
print(re.findall('^F.+?:', x))
print(re.findall('^From.*X-([!^i]*)', x))
```
Questions
Programming tasks