Outline

Boolean Operators
Exercises
If Statements and Graphics
Organization of a Graphics Program
Measuring Distances
Graphics: Obstacles
Eliza
Boolean Operators
Boolean Operators

And, Or, Not
Boolean Operators

And, Or, Not

True and False are Python values!
if expression1 and expression2:
    statement
    statement
    statement
else:
    statement
    statement
    statement
if expression1 and expression2:
    statement
    statement
    statement
else:
    statement
    statement
    statement
if expression1 and expression2:
    statement
    statement
    statement
else:
    statement
    statement
    statement
if expression1 and expression2:
  statement
  statement
  statement
else:
  statement
  statement
  statement
  statement
```python
if expression1 and expression2:
    statement
    statement
    statement
else:
    statement
    statement
    statement
```
if expression1 and expression2:
    statement
    statement
    statement
else:
    statement
    statement
    statement
    statement
if \textit{expression1} \textbf{and} \textit{expression2}:

\texttt{statement}

\texttt{statement}

\texttt{statement}

\texttt{else:}

\texttt{statement}

\texttt{statement}

\texttt{statement}

\texttt{statement}
if `expression1 and expression2`:

statement  
statement  
statement  

else:

statement  
statement  
statement  
statement
if `expression1` and `expression2`:
  statement
  statement
  statement
else:
  statement
  statement
  statement
if expression1 or expression2:
  statement
  statement
  statement
else:
  statement
  statement
  statement

True  False
False  True
True  True

False  False

Or
if not expression:
    statement
    statement
    statement
else:
    statement
    statement
    statement

Not
else is not always used...

```python
if no20s == 1:
    print( no20s, "$20-bill" )
else:
    print( no20s, "$20-bills" )
```
else is not always used...

caption = "$20-bill"
if no20s != 1:
    caption = caption + "s"
print( no20s, caption )
Outline

Boolean Operators

Exercises

If Statements and Graphics

Organization of a Graphics Program

Measuring Distances

Graphics: Obstacles

Eliza
If-Statements and Graphics
Where are Graphic Objects Defined?

Zelle's Graphics.py for Python 3

--D. Thiebaut (talk) 11:12, 8 March 2015 (EDT)

The file below, copyrighted by John Zelle, was downloaded from http://mcsp.wartburg.edu/zelle/python/graphics.py on 3/8/15, and mirrored here for convenience.

```
# graphics.py
"""Simple object oriented graphics library

The library is designed to make it very easy for novice programmers to experiment with computer graphics in an object oriented fashion. It is written by John Zelle for use with the book "Python Programming: An Introduction to Computer Science" (Franklin, Beedle & Associates).

LICENSE: This is open-source software released under the terms of the GPL (http://www.gnu.org/licenses/gpl.html).

PLATFORMS: The package is a wrapper around Tkinter and should run on any platform where Tkinter is available.

INSTALLATION: Put this file somewhere where Python can see it.

OVERVIEW: There are two kinds of objects in the library. The GraphWin class implements a window where drawing can be done and various GraphicsObjects are provided that can be drawn into a GraphWin. As a simple example, here is a complete program to draw a circle of radius 10 centered in a 100x100 window:
```

http://cs.smith.edu/dftwiki/index.php/Zelle%27s_Graphics.py_for_Python_3
Every element is an OBJECT

Examples

Organization of a graphic program

Something completely different...
p = Point( 50, 150 )

point1 = Point( 50, 150 )

x = point1.getX()
y = point1.getY()

if x <= 0 or y <= 0:
    # the point is outside the window
    ...

D. Thiebaut, Computer Science, Smith College
point1 = Point( 50, 150 )
circ1 = Circle( point1, 30 )

center1 = circ1.getCenter()
x = center1.getX()
y = center1.getY()
if x <= 0 or y <= 0:
    # the center is outside the window
    ...

point1 = Point( 50, 150 )
circ1 = Circle( point1, 30 )

center1 = circ1.getCenter()
x = center1.getX()
y = center1.getY()
if x <= 0 or y <= 0:
    # the center is outside the window
    ...

x = circ1.getCenter().getX()
y = circ1.getCenter().getY()
if x <= 0 or y <= 0:
    #
Rectangle Object

gETCH1()
gETCH2()

r = Rectangle( Point( 50, 150 ),
              Point( 150, 150 ) )

r.move( dx, dy )

x1 = r.getP1().getX()
y1 = r.getP1().getY()
x2 = r.getP2().getX()
y2 = r.getP2().getY()

mouseP = win.checkMouse()
if mouseP != None:
    x = mouseP.getX()
y = mouseP.getY()
    if x1 <= x <= x2 and ...
Every element is an OBJECT

Examples

Organization of a graphic program

Something completely different...
Outline

Boolean Operators

Exercises

If Statements and Graphics

Organization of a Graphics Program

Measuring Distances

Graphics: Obstacles

Eliza
Organization of a Graphic Program
def main():
    # open the graphic window

    # define and initialize the graphic objects

    # start animation loop. Stop on specific user interaction
    while win.checkMouse() == None:
        # move/update each object according to its speed
        # and direction

    # Loop is over.
    # close the graphic window
def main():
    # open the graphic window
    win = GraphWin( "Demo", 600, 400 )

    # define and initialize the graphic objects

    # start animation loop. Stop on specific user interaction

        # move/update each object according to its speed
        # and direction

    # Loop is over.
    # close the graphic window
def main():
    # open the graphic window
    win = GraphWin( "Demo", 600, 400 )
    circ = Circle( Point( 100, 100 ), 30 )
    circ.setFill( 'red' )
    circ.draw( win )
    dx, dy = 3, 2
    # start animation loop. Stop on specific user interaction
    # move/update each object according to its speed
    # and direction

    # Loop is over.
    # close the graphic window
def main():
    # open the graphic window
    win = GraphWin( "Demo", 600, 400 )

    # define and initialize the graphic objects
    circ = Circle( Point( 100, 100 ), 30 )
    circ.setFill( 'red' )
    circ.draw( win )
    dx, dy = 3, 2

    # start animation loop. Stop on specific user interaction
    while win.checkMouse() == None:
        # move/update each object according to its speed
        # and direction

    # Loop is over.
    # close the graphic window
def main():
    # open the graphic window
    win = GraphWin( "Demo", 600, 400 )

    # define and initialize the graphic objects
    circ = Circle( Point( 100, 100 ), 30 )
    circ.setFill( 'red' )
    circ.draw( win )
    dx, dy = 3, 2

    # start animation loop. Stop on specific user interaction
    while win.checkMouse() == None:
        # move/update each object according to its speed
        # and direction
        circ.move( dx, dy )

    # Loop is over.
    # close the graphic window
def main():
    # open the graphic window
    win = GraphWin( "Demo", 600, 400 )

    # define and initialize the graphic objects
    circ = Circle( Point( 100, 100 ), 30 )
    circ.setFill( 'red' )
    circ.draw( win )
    dx, dy = 3, 2

    # start animation loop. Stop on specific user interaction
    while win.checkMouse() == None:
        # move/update each object according to its speed
        # and direction
        circ.move( dx, dy )

    # Loop is over.
    # close the graphic window
def main():
    # open the graphic window
    win = GraphWin( "Demo", 600, 400 )

    # define and initialize the graphic objects
    circ = Circle( Point( 100, 100 ), 30 )
    circ.setFill( 'red' )
    circ.draw( win )
    dx, dy = 3, 2

    # start animation loop. Stop on specific user interaction
    while win.checkMouse() == None:
        # move/update each object according to its speed and direction
        circ.move( dx, dy )

    # Loop is over.
    # close the graphic window
def main():
    # open the graphic window
    win = GraphWin("Demo", 600, 400)

    # define and initialize the graphic objects
    circ = Circle(Point(100, 100), 30)
    circ.setFill('red')
    circ.draw(win)
    dx, dy = 3, 2

    # start animation loop. Stop on specific user interaction
    while win.checkMouse() == None:
        # move/update each object according to its speed
        # and direction
        circ.move(dx, dy)

    # Loop is over.
    # close the graphic window
def main():
    # open the graphic window
    win = GraphWin( "Demo", 600, 400 )

    # define and initialize the graphic objects
    circ = Circle( Point( 100, 100 ), 30 )
    circ.setFill( 'red' )
    circ.draw( win )
    dx, dy = 3, 2

    # start animation loop. Stop on specific user interaction
    while win.checkMouse() == None:
        # move/update each object according to its speed
        # and direction
        circ.move( dx, dy )

    # Loop is over.
    # close the graphic window
def main():
    # open the graphic window
    win = GraphWin( "Demo", 600, 400 )

    # define and initialize the graphic objects
    circ = Circle( Point( 100, 100 ), 30 )
    circ.setFill( 'red' )
    circ.draw( win )
    dx, dy = 3, 2

    # start animation loop. Stop on specific user interaction
    while win.checkMouse() == None:

        # move/update each object according to its speed
        # and direction
        circ.move( dx, dy )

    # Loop is over.
    # close the graphic window
    win.close()
def main():
    # open the graphic window
    win = GraphWin( "Demo", 600, 400 )

    # define and initialize the graphic objects
    circ = Circle( Point( 100, 100 ), 30 )
    circ.setFill( 'red' )
    circ.draw( win )
    dx, dy = 3, 2

    # start animation loop. Stop on specific user interaction
    while win.checkMouse() == None:

        # move/update each object according to its speed
        # and direction
        circ.move( dx, dy )

    # Loop is over.
    # close the graphic window
    win.close()
Outline

Boolean Operators

Exercises

If Statements and Graphics

Organization of a Graphics Program

Measuring Distances

Graphics: Obstacles

Eliza
Measuring Distances
\[ d^2 = a^2 + b^2 \]
\[ d^2 = a^2 + b^2 \]
\[ d^2 = a^2 + b^2 \]
d^2 = a^2 + b^2
\[ d = \sqrt{ (y_2 - y_1)(y_2 - y_1) + (x_2 - x_1)(x_2 - x_1) } \]
from math import *

def distance(x1, y1, x2, y2):
    return sqrt((x1-x2)*(x1-x2) + (y1-y2)*(y1-y2))
```python
from math import *

def distance( x1, y1, x2, y2 ):
    return sqrt( (x1-x2)*(x1-x2) + (y1-y2)*(y1-y2) )

def distanceP( p1, p2 ):
    x1, y1 = p1.getX(), p1.getY()
    x2, y2 = p2.getX(), p2.getY()
    return distance( x1, y1, x2, y2 )
```
Testing
# distanceDemo.py
# D. Thiebaut

```python
from math import *
from graphics import *

def distance( x1, y1, x2, y2 ):
    return sqrt((x1-x2)*(x1-x2) + (y1-y2)*(y1-y2))

def distanceP( p1, p2 ):
    x1, y1 = p1.getX(), p1.getY()
    x2, y2 = p2.getX(), p2.getY()
    return distance( x1, y1, x2, y2 )

def main():
    point1 = Point( 3, 5 )
    point2 = Point( 7, 8 )
    d = distanceP( point1, point2 )
    print( "distance =", d )

main()
```
Graphics: Detecting Obstacles
Exercise: Obstacle

Take the graphic program moving a circle around, and create an obstacle.
Outline

Boolean Operators

Exercises

If Statements and Graphics

Organization of a Graphics Program

Measuring Distances

Graphics: Obstacles

Eliza
Turing
The Imitation Game
The Turing Test

Turing 1912-1954

Dr Suilin Lavelle
University of Edinburgh
Blade Runner
• Example of Natural Language Processing (NLP)

• MIT, 1964, Joseph Weizenbaum

• One of the first chat-bots (Amazon Alexa)

• Emulates a Rogerian psychotherapist

• Example of dialogs: https://web.stanford.edu/group/SHR/4-2/text/dialogues.html
# Eliza.py
# D. Thiebaut
# A very short beginning program for Eliza
# just print the string to the console
# will be transformed to something better later...
def myprint( string ):
    print( string )

def sayGoodBye( name ):
    myprint( "Good bye " + name )

def isGoodBye( userAnswer ):
    if userAnswer.lower().strip() in [ "bye", "goodbye", "ciao" ]:
        return True
    else:
        return False
def greetings():
    myprint( "Hello there!" )
    myprint( "What is your name?" )
    name = input( "> " )
    myprint( "Welcome " + name )
    return name

def main():
    userName = greetings()
    for i in range( 1000 ):
        userAnswer = input( "> " )
        if isGoodBye( userAnswer ) == True:
            break
        myprint( "Please tell me more..." )
    sayGoodBye( userName )
Adding Randomness To Eliza's Dialogs

• Use the random library

```python
from random import *
...

prompts = ["Please go on...", "Please tell me more...",
            "Interesting... Go on, please!",
            "Yes? Really? Go on",
            "Weird... I'm not sure what to think of that..."
]
...

print( random.choice( prompts ) )
```
Looking for String Patterns
The Problem

**User types:** "I had a HUGE fight with my brother"

**Program knows:** ['mother', 'father', 'brother', 'sister']
"I had a HUGE fight with my brother"

```python
split()
[
    "I",
    "had",
    "a",
    "huge",
    "fight",
    "with",
    "my",
    "brother"
]
```

in
```python
["mother", "father", "brother", "sister"]
```
"I had a HUGE fight with my brother"

```
[ "I", "had", "a", "huge", "fight", "with", "my", "brother" ]
```

Option 1
"I had a HUGE fight with my brother"
"I had a HUGE fight with my brother"
"I had a HUGE fight with my brother"

```python
family = ["mother", "father", "brother", "sister"]

userInput = input("> "
words = userInput.lower().split()

familyMatter = False
for word in words:
    if word in family:
        familyMatter = True
        break

if familyMatter == True:
    doSomething()
```

Option 1
"I had a HUGE fight with my brother"

Option 2

[ "mother",
  "father",
  "brother",
  "sister" ]
"I had a HUGE fight with my brother"

```javascript
.find(['mother', 'father', 'brother', 'sister'])
```
"I had a HUGE fight with my brother"

```
Option 2

D. Thiebaut, Computer Science, Smith College
```
"I had a HUGE fight with my brother"

```
[ "mother",
  "father",
  "brother",
  "sister"
]
```
"I had a HUGE fight with my brother"

```python
family = ['mother', 'father', 'brother', 'sister']

userInput = input( "> " ).lower()

familyMatter = False
for word in family:
    if userInput.find( word ) != -1:
        familyMatter = True

if familyMatter == True:
    doSomething()
```

Option 2

```python
[ "mother",
  "father",
  "brother",
  "sister"
]```
"I had a HUGE fight with my brother"

family = ["mother", "father", 
           "brother", "sister"]

userInput = input( "> " ).lower()

familyMatter = False
for word in family:
    if userInput.find( word ) != -1:
        familyMatter = True
        break

if familyMatter == True:
    doSomething()