How does a computer get drunk?

It takes screenshots!

Source: https://www.rd.com/jokes/computer/

Thank you Anant!
Last Lecture

• Practice Exam #4
• Hand-in your in-class activity #4 at the end of this lecture
Review – From last lecture

1. How do we create a Turtle object, given the turtle module?
2. What is the keyword necessary to make a while loop?
3. What is the keyword necessary to create a function?
4. What is the difference between function parameters and function arguments?
5. Why do we place our functions at the top of our program?
Today’s Menu

• Continue learning about **turtle**
• Continue creating animation programs using
  • **Turtle**
  • **Functions**
• Introduce **tuples**
Last Lecture

Problem Statement:

- Write a program that draws a blue square with a Turtle

---

```python
# DrawSquare.py
#
# Description: Draws a square with a Turtle given a colour and the size of the square's sides.
#
# Anne Lavergne
# Date: Feb. 14 2024
#
# Import the turtle library
import turtle

def drawSquare(aColour, aSideSize):
    """Draw a square of colour "aColor" and of side "sideSize".""

    # Set tt's pen (tail) to blue
    tt.color(aColour)

    # Tell tt to draw four sides
    # starting with tt faces east at (0,0)
    for side in range(4):  # 0,1,2,3
        tt.forward(aSideSize)
        tt.left(90)

    return

# ***Main part of the program
#
# Creates a graphics window "canvas"
canvas = turtle.Screen()

# Give a background colour to "canvas"
canvas.bgcolor("yellow")

# Create a turtle named "tt"
tt = turtle.Turtle()

# Set the colour of the square.
theColour = "hotpink"

# Set the size of the square.
theSizeOfSide = 150

# Tell tt to draw a square of "theColour" and of "theSizeOfSide"
drawSquare(theColour, theSizeOfSide)

# Clock on canvas to exit
canvas.exitonclick()
```
How about this!

• **Step 1 - Problem statement:**
  - Write a program that draws a chocolate chip cookie with our Turtle

• **Step 2 – Design**
  - Let’s create our algorithm
  - As a software developers, we do not need to be told to create a function 😊
  - We know that the functionality of drawing a cookie needs to be implemented as a function

• **Step 3 – Implementation**
  - Let’s transform this algorithm into Python code

• **Step 4 – Testing**
  - Does our program work i.e., solve the problem?

We shall implement our Cookie program **incrementally:** implement some code, test, repeat!

**Incremental development:** let’s draw the cookie first then the chocolate chips! 😊
# Tell tt to draw a chocolate chip cookie
# Draw the contour of the cookie
    tt.penup()
    tt.goto(-140,-120)
    tt.pendown()
    tt.circle(30)
    tt.penup()

# Draw a chocolate chip in the middle of the cookie
# Draw a chocolate chip in the top left area of the cookie
# Draw a chocolate chip in the bottom left area of the cookie
# Draw a chocolate chip in the bottom right area of the cookie
# Draw a chocolate chip in the top right area of the cookie
# Click on the canvas to exit
canvas.exitonclick()
What if I want many cookies?

• To solve the “many cookies” problem ...

• Call `drawCookie()` from the *Main part of my program* using a `for` loop!
What if I want many cookies?

• Hum... calling `drawCookie()` in a `for` loop from the main part of my program would draw all the cookies in the same spot

• Solution:
  • Add parameters to our `drawCookie()` function definition
  • Use these parameters in the `body` of our `drawCookie()` function
  • And when we call our `drawCookie()` function, we pass arguments to it and these arguments can have different values!

What kinds of parameters should we add to our function?
How to draw the chocolate chips?

• Solution:
  • Could we copy and modify the code many times, each instance of the code would be drawing a chocolate chip?

• Hum... This solution would lead to a lot of repeating code, which is not a good idea!

Why?
# Tell `tt` to draw a chocolate chip cookie
# Draw the contour of the cookie
`tt.penup()`
`tt.goto(-140,-120)`
`tt.pendown()`
`tt.circle(30)`
`tt.penup()`

# Draw a chocolate chip in the middle of the cookie
# Draw a chocolate chip in the top left area of the cookie
# Draw a chocolate chip in the bottom left area of the cookie
# Draw a chocolate chip in the bottom right area of the cookie
# Draw a chocolate chip in the top right area of the cookie
# Click on the canvas to exit
`canvas.exitonclick()`
Next Lecture

• Differentiate
  • variables \textbf{local} to a function – local scope versus
  • variables used outside of a function

• Introduce a new kind of algorithm: \textit{recursion}