CS65: Introduction to Computer Science

Random Number

`while` Loop

Md Alimoor Reza
Assistant Professor of Computer Science

Topics

- Assignment 1 has been released
  - Due on next Thursday, October 06
  - I also uploaded a skeleton code on Blackboard. You can start with that.
  - It is very similar to Lab 2 and Lab 3. You can borrow code snippets from those two labs if you like.
- Random number generation
- Two different ways to solve a repetitive task in Python
  - The **`while`** loop

Topic: Random Number

- Random numbers are useful several programming tasks:
  - Simulating a coin toss — random flipping of head or tail
  - Simulating a dice roll — random roll of one of six sides
  - Simulating a card shuffling - randomly selecting cards (out of 52)

- Python provides library to generate random numbers
  - Like `math` module or `graphics` module, you can import random module to get access to random number generating functions

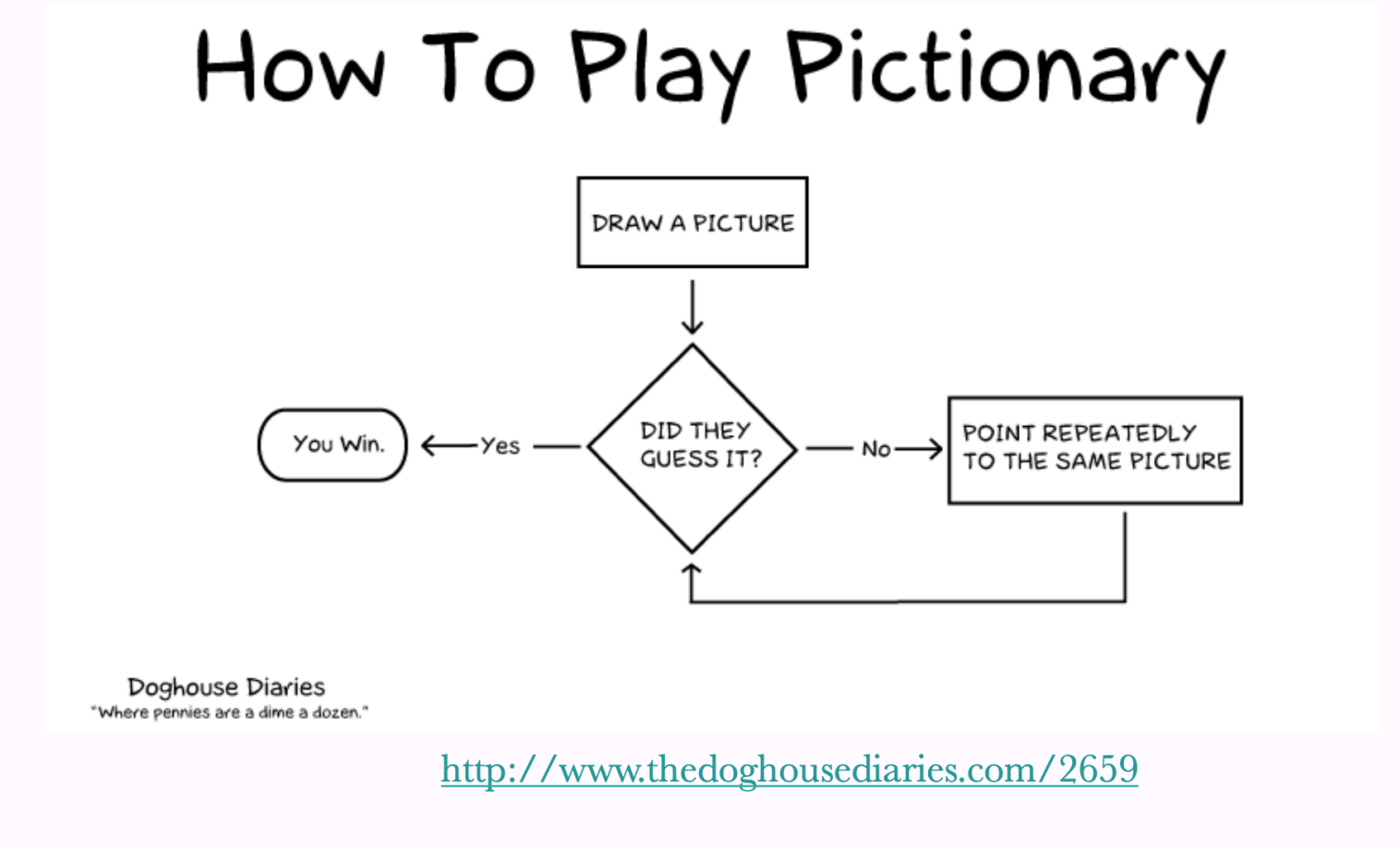
Topic: Random Number

- Steps for generating a random number are as follows:
  - Step 1: Import `random` module
  - Step 2: Generate a random number (*eg, an integer number*) between a range of values denoted by a lower_range and an upper_range
    - For example, in order to generate a random integer between lower_range of 1 and upper_range of 10, we need to do the following:

```
import random

rand_number = random.randint(1, 10)
print(rand_number)
```

Motivation: Loop



http://www.thedoghousediaries.com/2659

Topic: Solving Repetitive Task

- Designed to solve a repetitive task — runs a block of code based on a Boolean expression:
  - Summing all the numbers from 0 to 100
  - Taking user inputs until a special number is provided

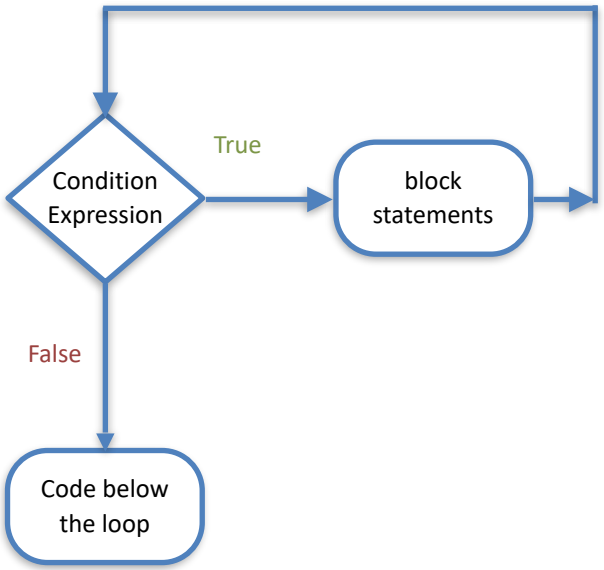
- Two different ways to solve a repetitive task in Python
  - The **while** loop
  - The **for** loop

Topic: Solving Repetitive Task

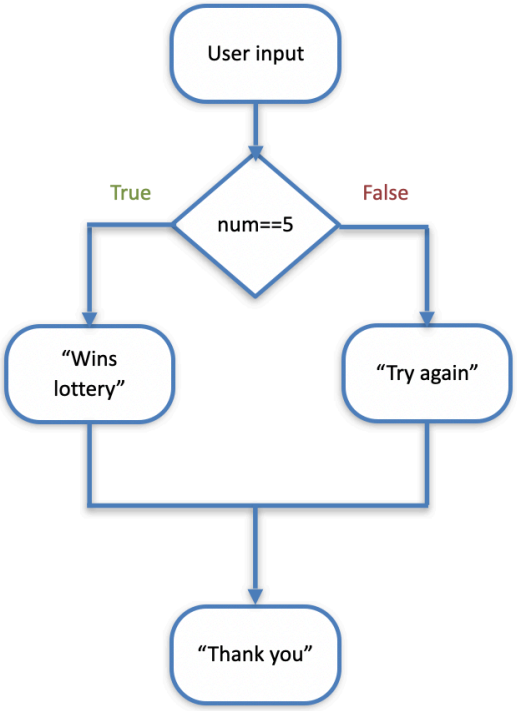
- Designed to solve a repetitive task — runs a block of code based on a Boolean expression:
  - Summing all the numbers from 0 to 100
  - Taking user inputs until a special number is provided

- Two different ways to solve a repetitive task in Python
  - The **while** loop
  - The **for** loop

`while` Loop
vs
`if`/`else` Selection Statements



While loop



If/else blocks

`while` Loop

- The index variable can be updated (**decreased**) with a shorthand:

```
num = 5

while num > 0:

    print(num)
    num = num - 1
```

```
>>> %Run lecture8_while.py

  5
  4
  3
  2
  1

>>>
```

```
num = 5

while num > 0:

    print(num)
    num -= 1
```

```
>>> %Run lecture8_while.py

  5
  4
  3
  2
  1

>>>
```

`while` Loop

- The index variable can be updated (**increased**) with a shorthand:

```
num = 5

while num > 0:

    print(num)
    num = num + 1
```

```
num = 5

while num > 0:

    print(num)
    num += 1
```

Exercise 1

- Write a code that will do the following:
  - prompt the user for an integer *(between 1 to 100)*
  - then **computes** the sum of all number from 0 to the given number

Exercise 2

- Write a code that will do the following:
  - prompt the user for an integer number *(between 1 to 100)*
  - then **prints** all the even numbers between 0 and the given number

Exercise 3

- Write a code that will do the following:
  - Prompt the user to enter one integer number.
  - Then your program should find the summation of all the odd numbers between **1** and that integer number.
  - For example, if the user enters 5. Your program should print 6 as the summation of 1+3+5 is equal to 9.

Exercise 4

- Write a code that will do the following:
  - prompt the user for a state's name from the following:
    - {"NY", "PA", "MD", "VA"},
    - {"IA", "IN", "IL", "MN"},
    - {"TX", "LA", "FL", "AK"},
    - {"CA", "OR", "WA", "NV"}
  - then prints its geographic location from one of the categories:
    - "Eastern", "Midwestern", "Southern", "Western"
  - program will terminate only when the user enters "END"