Quiz

- Define at least 3 laptop components.
- What's duo-core processor?
- What's the difference between Random Access Memory(RAM) and Hard Disk Drive(HDD)?
- What's the difference between program and algorithm?
- what's the difference between binary number system and decimal number system?

Algorithmic & Python Programming

Imad Kissami[1]
[1]Mohammed VI Polytechnic University, Benguerir, Morocco

Variables

Reserved Words and Identifiers

- Reserved word
  - Word that has a specific meaning in Python
    - Ex: def, return
- Word used to name and refer to a data element or object manipulated by the program.

Variables

Valid Identifier Names

- Begins with a letter or underscore symbol
- Consists of letters, digits, or underscores only
- Cannot be a Python reserved word
- Case sensitive
  - Total $\neq$ total $\neq$ TOTAL
- Examples:

```
1  distance
2  milesPerHour
3  _voltage
4  goodChoice
5  high_level
6  MIN_RATE
```

Variables

Invalid Identifier Names

- Does not begin with a letter or underscore symbol
- Contains other than letters, digits, and underscore
- Examples:

```
1  x-ray
2  2ndGrade
3  $amount
4  two&four
5  after five
6  return
```

Variables

Identifier Name Conventions

- Standard practice, not required by Python language
  - Normally lower case
  - Constants upper case
- Multi-word
  - Underscore between words or
  - Camel case - each word after first is capitalized

```
1  distance
2  TAX_RATE %constant
3  miles_per_hour
4  milesPerHour
```

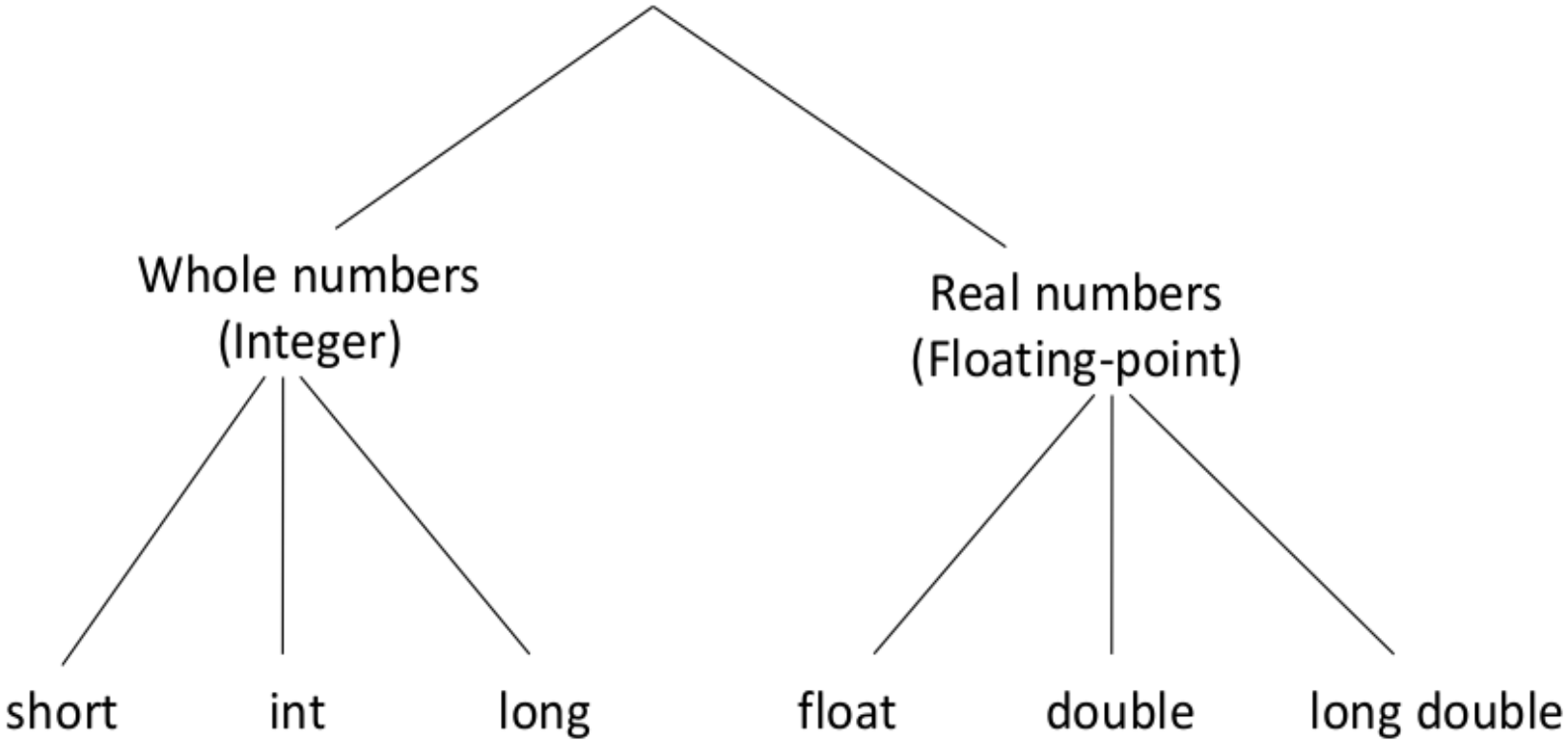
Variables

Variable

- Name is a valid identifier name
- Is a memory location where a value can be stored for use by a program
- Value can change during program execution
- Can hold only one value
  - Whenever a new value is placed into a variable, the new value replaces the previous value.

Variables

Numeric Data Types



Variables

Data Types and Typical Sizes

Type Name	Memory Used	Size Range	Precision	Guarantee
short (= short int)	2 bytes	-32,768 to 32,767	N/A	16 bits
int	4 bytes	-2,147,483,648 to 2,147,483,647	N/A	16 bits
long (= long int)	8 bytes	-9,223,372,036,854,775,808 to 9,223,372,036,854,775,807	N/A	32 bits
float	4 bytes	approximately 10^{-38} to 10^{38}	7 digits	6 digits
double	8 bytes	approximately 10^{-308} to 10^{308}	15 digits	10 digits
long double	10 bytes	approximately 10^{-4932} to 10^{4932}	19 digits	10 digits

Algorithmic & Python Programming

Imad Kissami[1]
[1]Mohammed VI Polytechnic University, Benguerir, Morocco

Repetition & Looping

Example 1

```
1  a = input("")
2  b = input("")
3  sum = int(a)+int(b)
4  print(sum);
```

- What if we want to process three different pairs of integers?

Repetition & Looping

Example 2

- One solution is to copy and paste the necessary lines of code. Consider the following modification:

```
1   a = input("")
2   b = input("")
3   sum = int(a)+int(b)
4   print(sum);
5
6   a = input("")
7   b = input("")
8   sum = int(a)+int(b)
9   print(sum);
10
11  a = input("")
12  b = input("")
13  sum = int(a)+int(b)
14  print(sum);
```

- What if you wanted to process four sets? Five? Six? $\cdots$

Repetition & Looping

Processing an arbitrary number of pairs

- We might be willing to copy and paste to process a small number of pairs of integers but
- How about 1,000,000 pairs of integers?
- The solution lies in mechanisms used to control the flow of execution
- In particular, the solution lies in the constructs that allow us to instruct the computer to perform a task repetitively

Repetition & Looping

Repetition (Looping)

- Use looping when you want to execute a block of code several times
  - Block of code = Body of loop
- Python provides two types of loops
  - while statement
    * Most flexible
    * No 'restrictions'
  - for statement
    * Natural 'counting' loop

Repetition & Looping

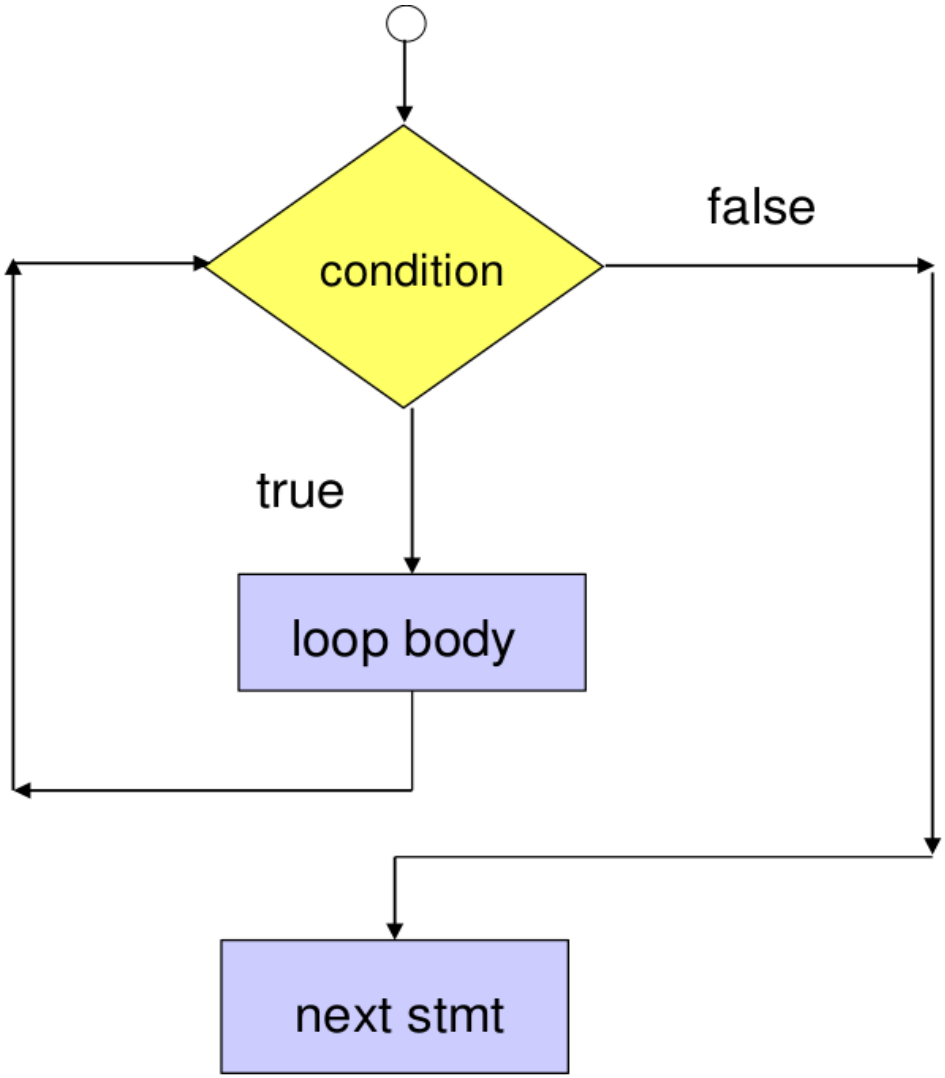
The while Repetition Structure

- Repetition structure
  - Programmer specifies
    * Condition under which actions will be executed
    * Actions to be repeated
  - Pseudocode
    * While there are more items on my shopping list Purchase next item and cross it off my list
- while loop repeated
  * As long as condition is true
  * Until condition becomes false

Repetition & Looping

The while Repetition Structure

- The condition is tested
- If the condition is true, the loop body is executed and the condition is retested.
- When the condition is false, the loop is exited.



Repetition & Looping

The while Repetition Structure

- Syntax:

```
1  while (expression):
2      basic block
```

- Expression = Condition to be tested
  - Resolves to true or false
- Basic Block = Loop Body
  - Reminder - Basic Block:
    * Single statement or
    * Multiple statements enclosed in braces

Repetition & Looping

Loop Control Variable (LCV)

- The loop control variable is the variable whose value controls loop repetition.
- For a while loop to execute properly, the loop control variable must be
  - initialized
  - tested
  - updated in the body of the loop in such a way that the expression/condition will become false
    * If not we will have an endless or infinite loop

Repetition & Looping

Example

```
1  count = 0                    # 1. Initialize LCV
2
3  while (count < 5):           # 2. Test LCV
4
5      num1 = input(""); num2 = input("")
6      sum = int(num1) + int(num2)
7      print(sum)
8
9      count=count+1            # 3. Increment LCV
```

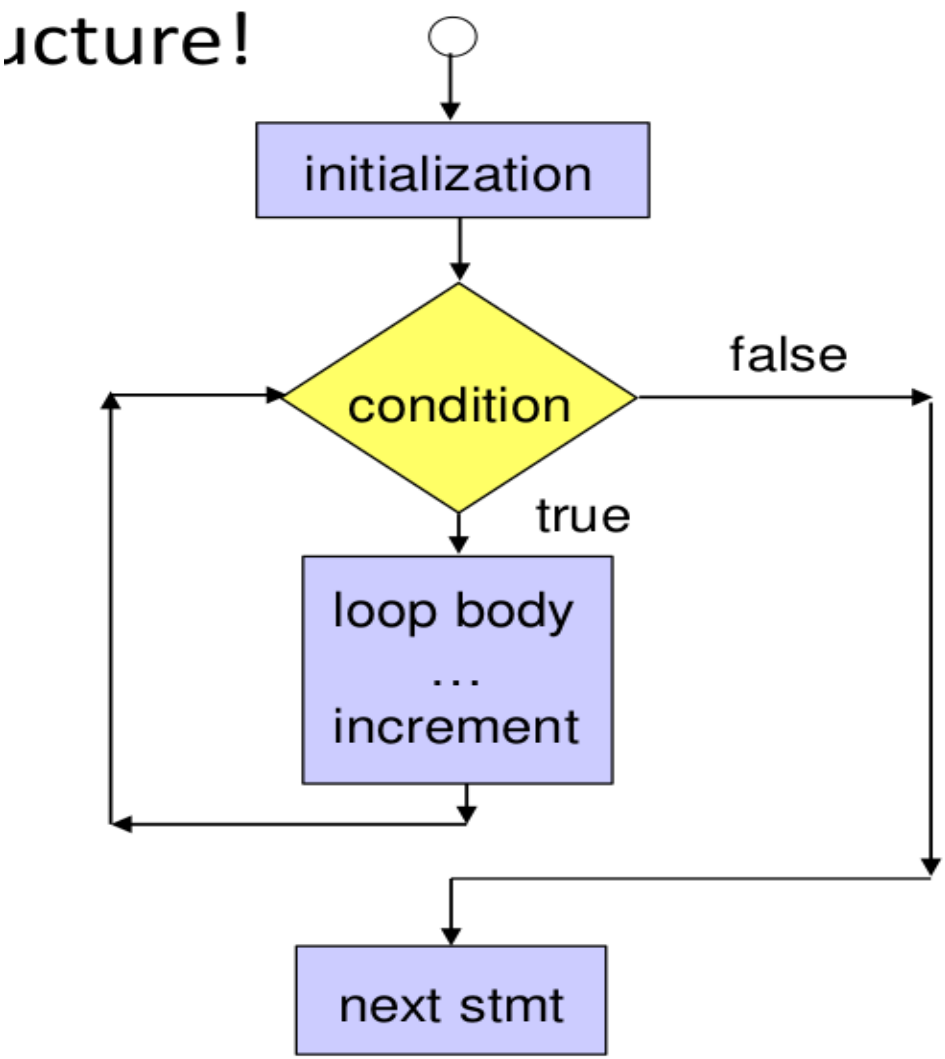
EXECUTION CHART

count	count<5	repetition
0	true	1
1	true	2
2	true	3
3	true	4
4	true	5
5	false	

Repetition & Looping

The while Repetition Structure

- A natural 'counting' loop
- Steps are built into for structure!
  1. Initialization
  2. Loop condition test
  3. Increment or decrement



Repetition & Looping

The for Repetition Structure

- Syntax:

```
1  for i in range():
2      basic block
```

- Example: Prints the integers from 0 to 9

```
1  for i in range(10):
2      print(i)
```

Repetition & Looping

The break/continue Statements

- break
  - Causes immediate exit from a while, for
- continue
  - Control passes to the next iteration