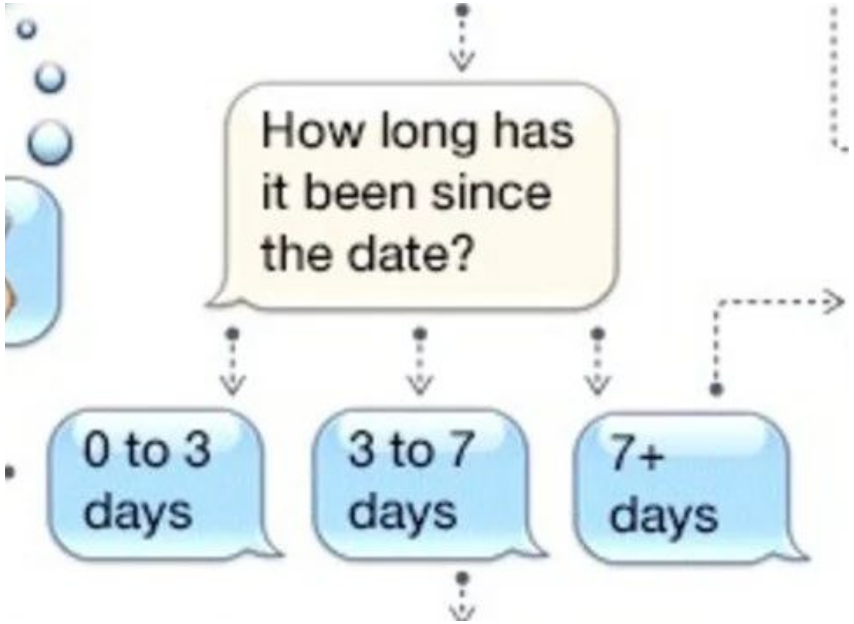
Chapter 7:
Decision Structures



Feb 11, 2020

Today’s Outline

- Review:
  - Functions with Lists
  - Quiz 5 Review
- Decision Structures
- Announcements

Function Review

Recall: A function is a “subprogram” that is outside of the main program.

When a function is “called”, the main program will be suspended, and the code inside the function will be executed.

Scope

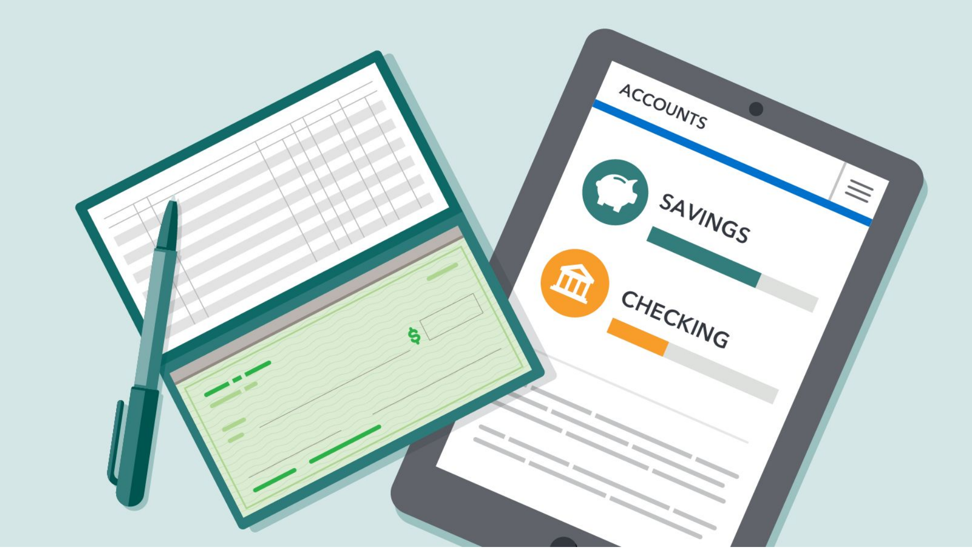
Scope refers to the places in a program where a given variable may be referenced.

The variables inside a function are **local** to that function.

What will the output be?

```
def addInterest(balance, rate):
  newBalance = balance * (1+rate)
  amount = newBalance


def main():
  amount = 1000
  rate = 0.05
  addInterest(amount,rate)
  print(amount)
main()
```



What will the output be?

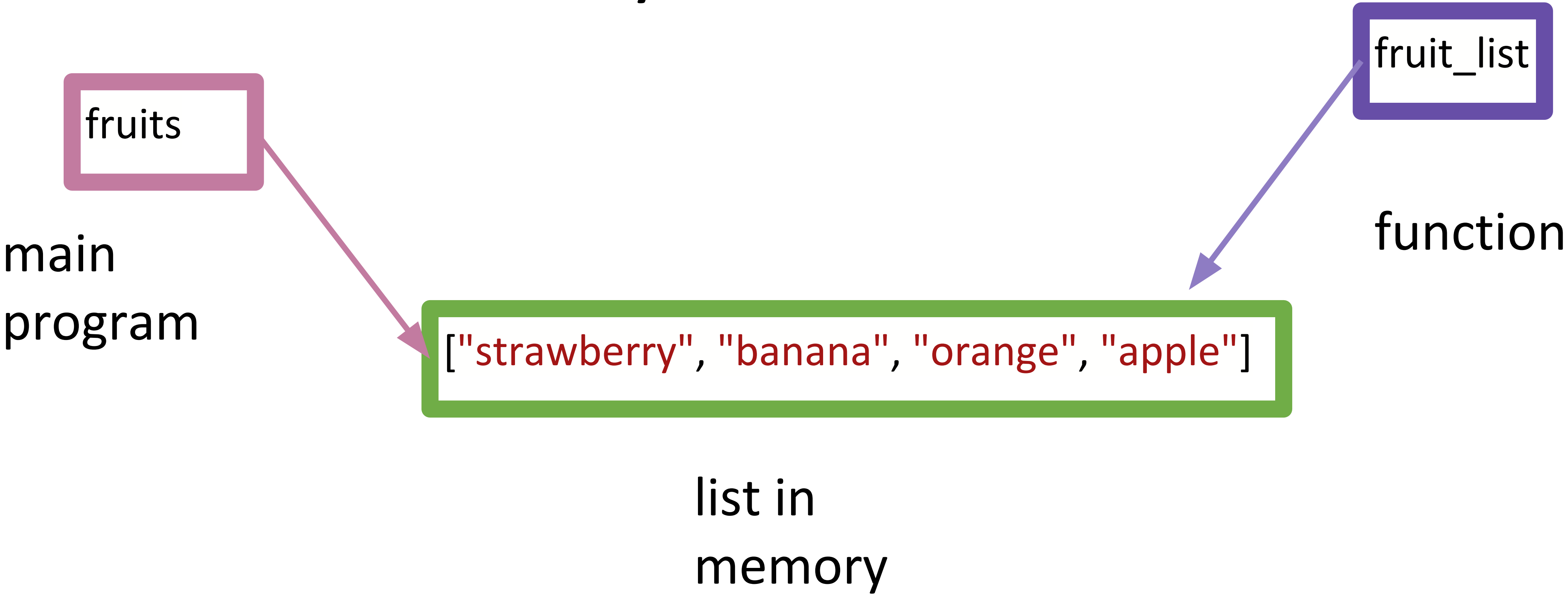
```
def addFruits(fruit_list):
  fruit_list.append("grape")
  fruit_list.append("pear")
  return fruit_list

def main():
  fruits = ["strawberry", "banana", "orange", "apple"]
  addFruits(fruits)
  print(fruits)
main()
```

**fruits and fruit_list are referring to the same list in memory

**in this specific case, updating fruit_list is the same as updating fruits

Fruit list memory



fruits

main program

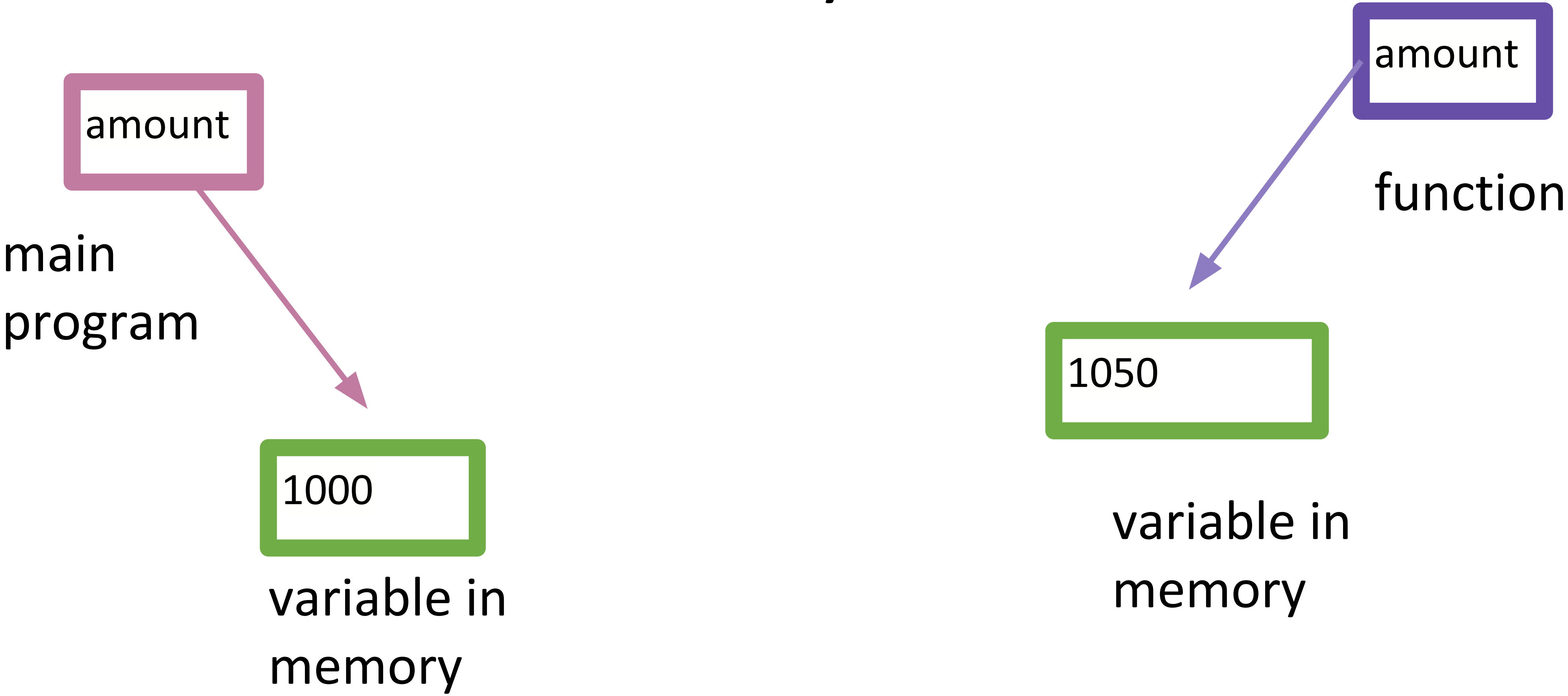
["strawberry", "banana", "orange", "apple"]

list in memory

fruit_list

function

Bank Account memory



amount

main program

1000

variable in memory

amount

function

1050

variable in memory

Summary

Variables within a function are local to the function. Exceptions occur if an object that is passed into a function is **mutable**.

Since lists and graphics objects are mutable, the function will be able to change the state of these objects in the main program.

Bank Accounts: Lists Example

Make a function that adds interest to a list of bank accounts.

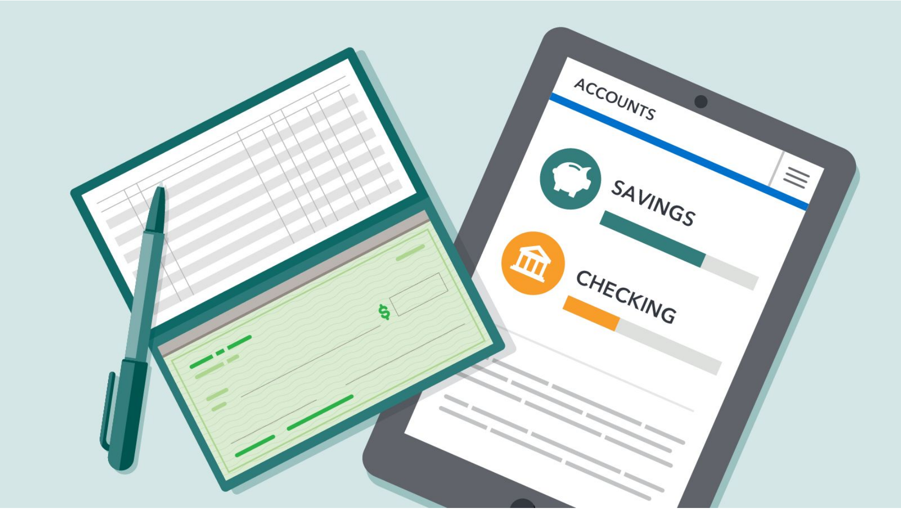
Input Example:

accounts = [1000, 0, 50000, 1000000]

rate = 0.05

Output Example:

accounts = [1050, 0, 52500, 1050000]



```python
def addInterests (accounts, rate):
  for i in range(0,len(accounts)):
    accounts[i] = accounts[i]*(1+rate)

def main ():
  bankAccounts = [1000, 0, 50000, 1000000]
  rate = 0.05

  addInterests(bankAccounts,rate)
  print(bankAccounts)
main()
```

Sequences with Functions

Write an efficient program that sings the song “Happy Birthday” to each of the Dionne quintuplets: Yvonne, Annette, Cécile, Émilie, and Marie.

Happy birthday to you!

Happy birthday to you!

Happy birthday dear [name]

Happy birthday to you!

```
def happy(text):
  return ("Happy birthday {0}\n".format(text))

def birthday(name):
  return(2*happy("to you!")+ happy("dear " + name) + happy("to you!"))

def main():
  quintuplets = ["Yvonne", "Annette", "Cécile", "Émilie", "Marie"]
  for name in quintuplets:
    print(birthday(name))

main()
```

Characters to Numbers

Write and test a function to meet this specification:

toNumbers (strList)

strList is a list of strings, each of which represents a number. Modify each entry in the list by converting it to a number.

Input: ['1', '35','25']

Output: [1, 35, 25]

Pig Latin Bonus

Napoleon the pig is very clever and wants to read War and Peace.

Make a program that translates War and Peace into Pig Latin and saves it into a .txt file so that he can read the book.

Submit it to him by Feb 14.

Quiz 5 Review

Quiz 5 will test concepts on files and functions.

The deadline will be **Sunday Feb 16** at 11:55pm.

Pollev.com/itec5920w

Which of the following is not a file-reading method in Python?

a) read
b) readline
c) readall
d) readlines

Before reading or writing to a file, a file object must be created via

a) open
b) create
c) file
d) none of the above

The string "slots" that are filled in by the format method are marked by:

a) %
b) $
c) []
d) {}

What does the following Python expression output?

```
print("{0}={1:5.2f}".format("pi", 3.14159))
```

a) {0}={1:5.2f}
b) pi= 3.14
c) pi=3.14159
d) pi=3.14

Which of the following is a reason to use functions?

a) to make the code more organized
b) to avoid duplicating code
c) To make it easier to read by keeping sections of code shorter
d) all of the above

When the execution of a function ends, what Python code is executed next?

a) the function is automatically executed again
b) the program ends
c) the code immediately after where the function was called from
d) the line of code immediately after the end of the function

What variable names does a function have access to?

a) only the parameters passed to the function
b) only variables created in the function
c) parameters that are passed to the function and any variables created in the function
d) all variables used in the program

When a function returns a value, the options for the caller are to

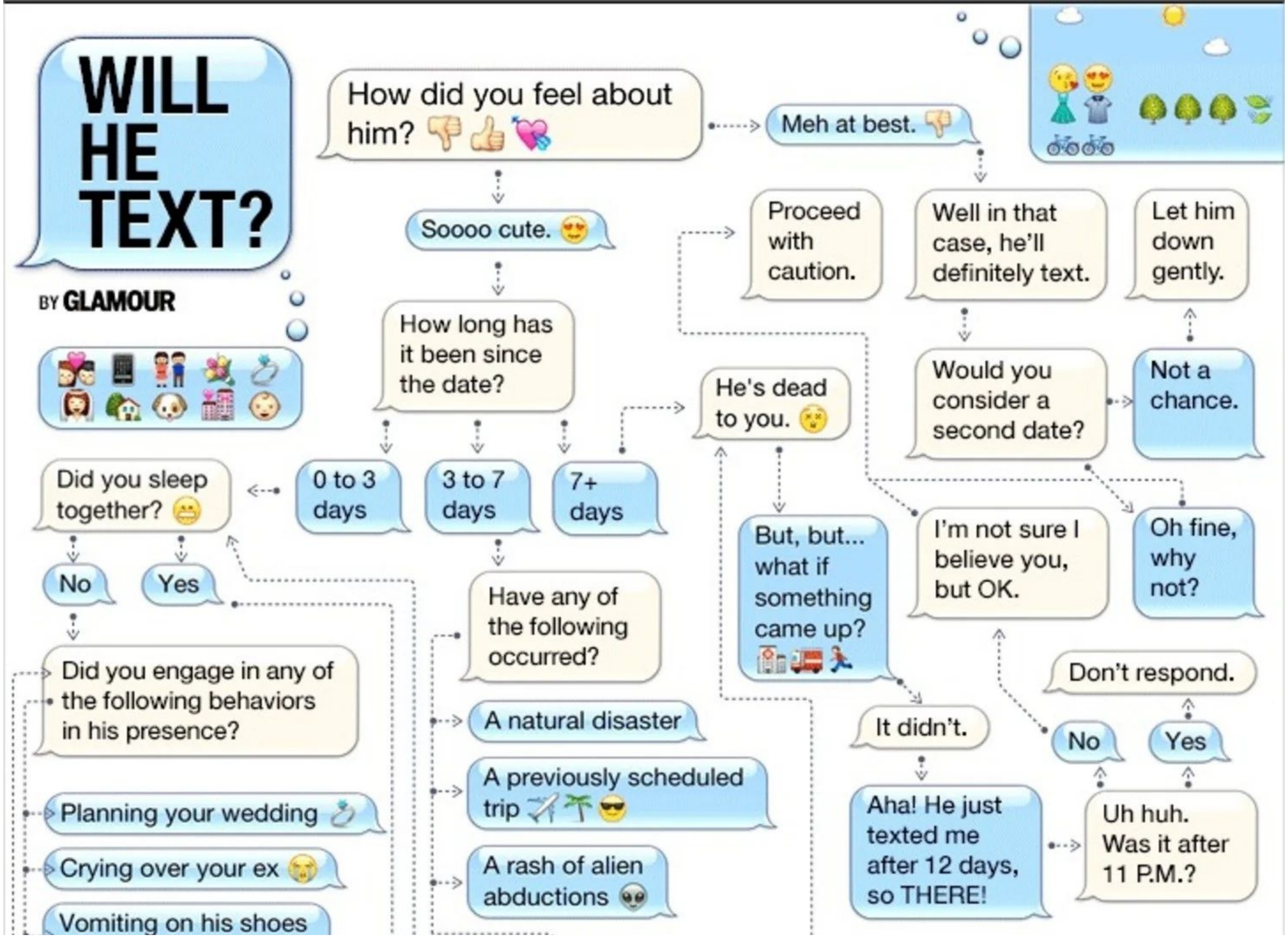
a) ignore the return value
b) use the result as part of an expression (e.g. x = y + math.sqrt(5))
c) assign it to a variable (e.g. x = sqrt(5))
d) all of the above

A function can modify the value of an actual parameter only if it's

a) mutable
b) a list
c) passed by value
d) a variable

A function with no return statement returns

a) nothing
b) its parameters
c) its variables
d) a special object called None



WILL HE TEXT?

BY GLAMOUR

How did you feel about him?

Meh at best.

Well in that case, he'll definitely text.

Would you consider a second date?

Not a chance.

Let him down gently.

Oh fine, why not?

I'm not sure I believe you, but OK.

Proceed with caution.

Soooo cute.

How long has it been since the date?

0 to 3 days

3 to 7 days

7+ days

Did you sleep together?

No

Yes

Did you engage in any of the following behaviors in his presence?

Planning your wedding

Crying over your ex

Vomiting on his shoes

Have any of the following occurred?

A natural disaster

A previously scheduled trip

A rash of alien abductions

He's dead to you.

But, but... what if something came up?

It didn't.

Aha! He just texted me after 12 days, so THERE!

Uh huh. Was it after 11 P.M.?

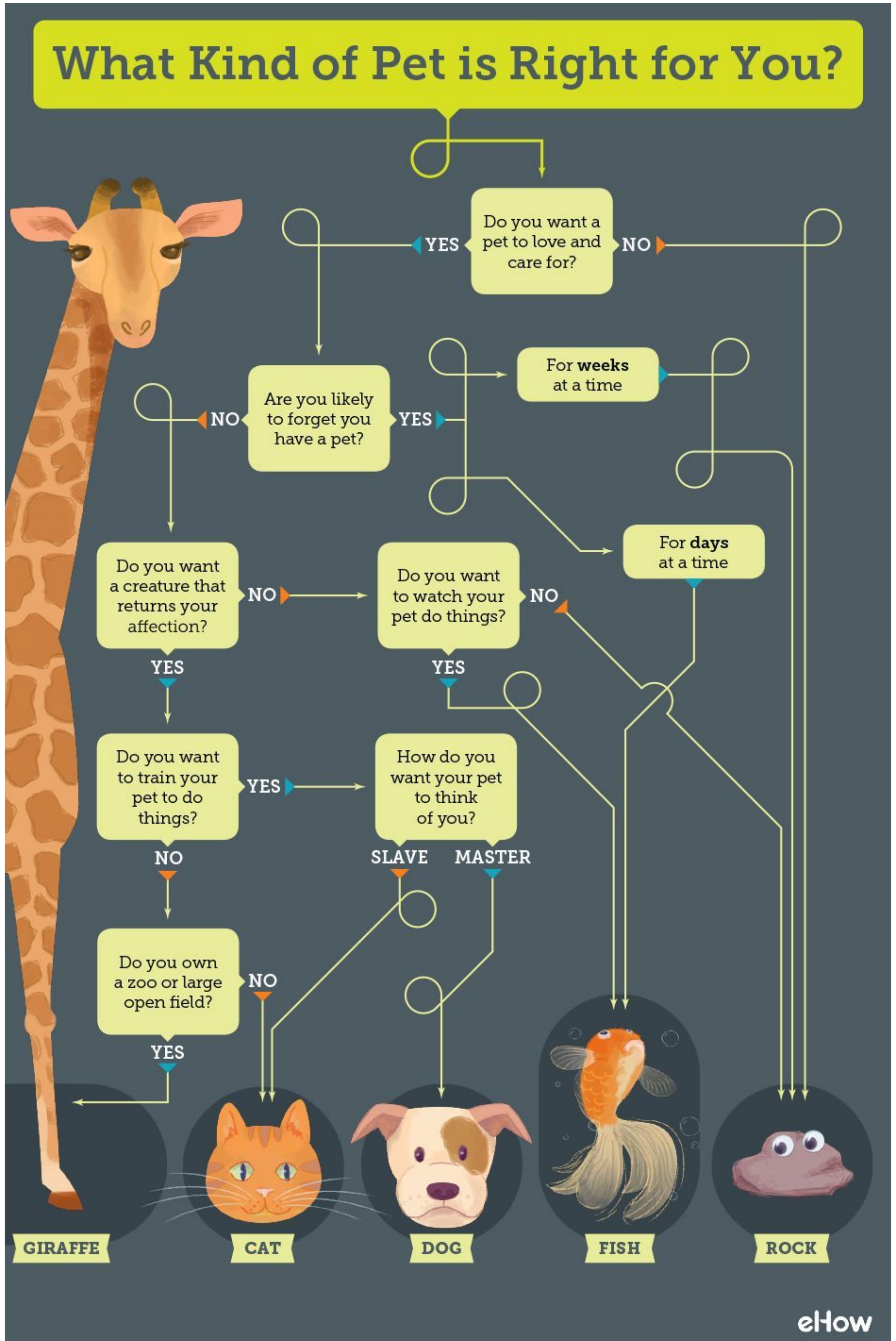
No

Yes

Don't respond.

Decision Structures

https://i.pinimg.com/originals/55/cc/a7/55cca7c380856eb2300caf24fd95097c.jpg



What Kind of Pet is Right for You?
Do you want a pet to love and care for?
YES
NO
Are you likely to forget you have a pet?
NO
YES
For weeks at a time
For days at a time
Do you want a creature that returns your affection?
NO
YES
Do you want to watch your pet do things?
NO
YES
Do you want to train your pet to do things?
YES
NO
How do you want your pet to think of you?
SLAVE
MASTER
Do you own a zoo or large open field?
NO
YES
GIRAFFE
CAT
DOG
FISH
ROCK
eHow

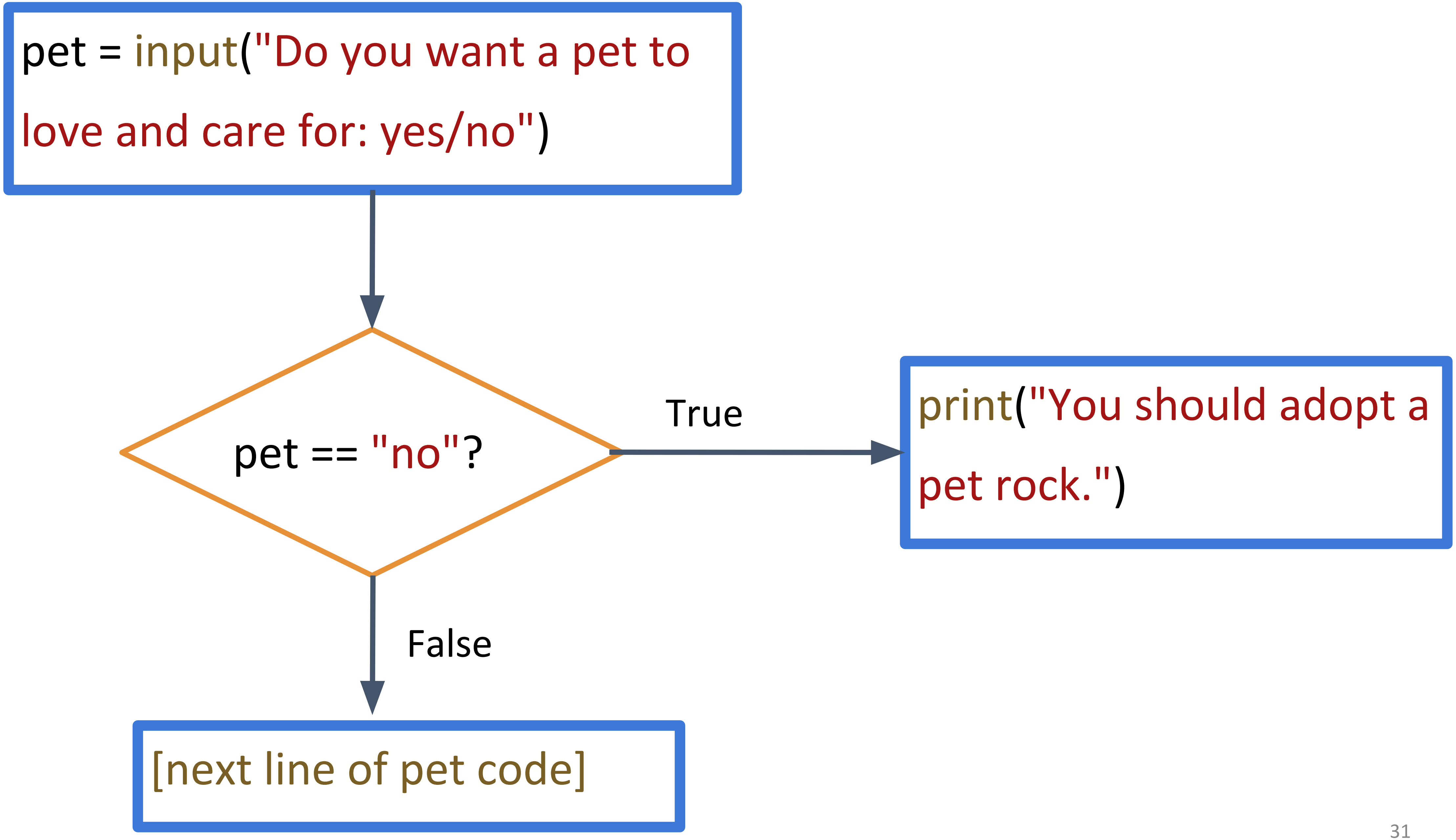
If Statement

```
if <condition>:
  <body of code>
```

Basic Pet Decision

```
#pet
pet = input("Do you want a pet to love and care for?: yes/no")

if pet == "no":
  print("You should adopt a pet rock.")
```



pet = input("Do you want a pet to love and care for: yes/no")

pet == "no"?

True

print("You should adopt a pet rock.")

False

[next line of pet code]

Relational Operators

< less than

<= less than or equal to

> greater than

>= greater than or equal to

== equal to

!= not equal to

Pet Decision with Integers

```
#pet income
income = eval(input("What is your annual income? "))

if income <= 100000:
  print("You should adopt a puppy.")

if income > 100000:
  print("You should adopt a giraffe.")
```

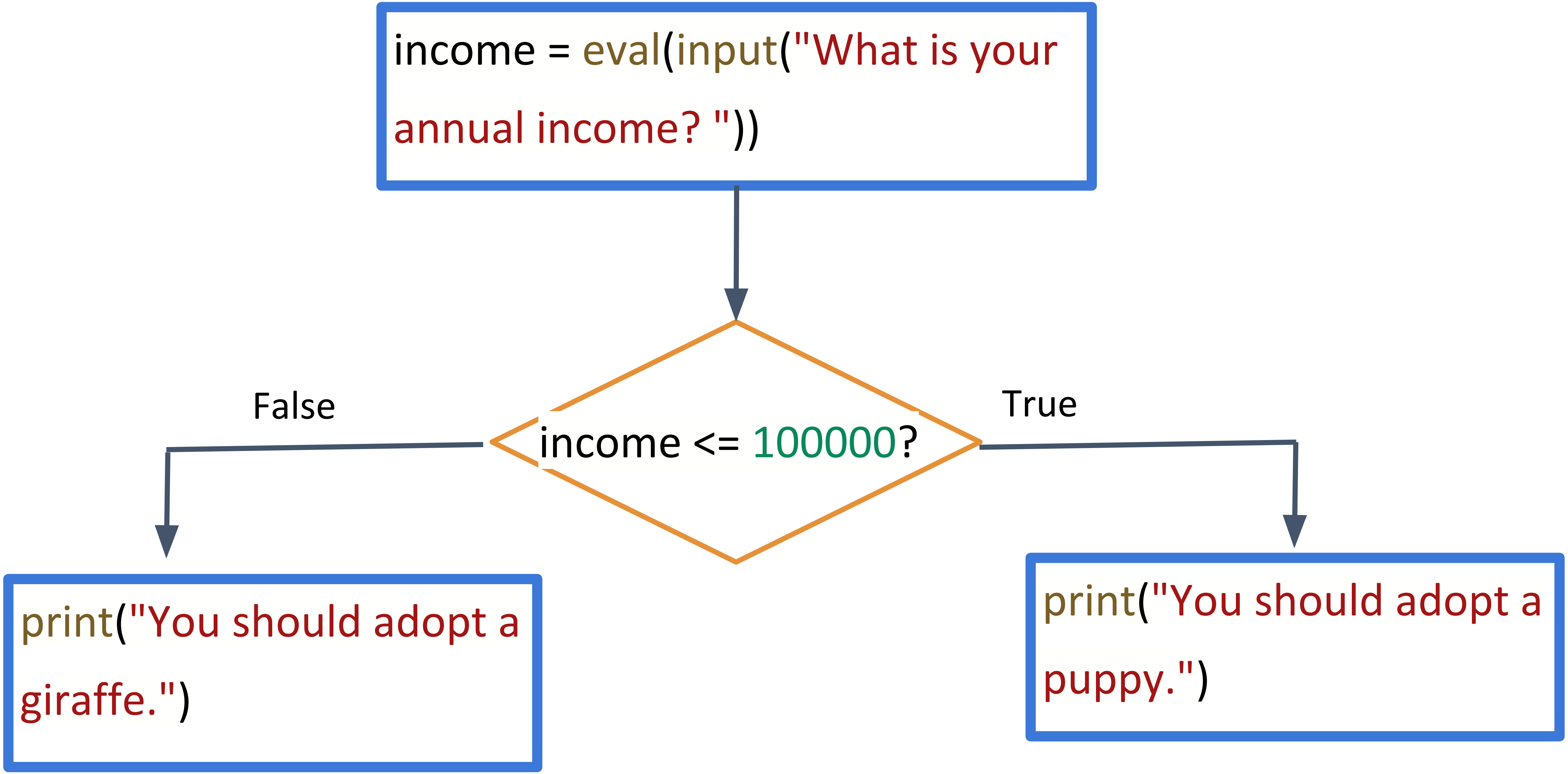
If - else statement: 2 way decision

```
if <condition 1>:
  <case 1 statement>
else:
  <default statement>
```

If - else statement: 2 way decision

```
#pet income
income = eval(input("What is your annual income? "))

if income <= 100000:
  print("You should adopt a puppy.")
else:
  print("You should adopt a giraffe.")
```



```
income = eval(input("What is your annual income? "))
```

income <= 100000?

False

True

```
print("You should adopt a giraffe.")
```

```
print("You should adopt a puppy.")
```

Decision Example

Many companies pay time-and-a-half for any hours worked above 40 in a given week.

Write a program to input the number of hours worked and the hourly rate and calculate the total wages for the week.

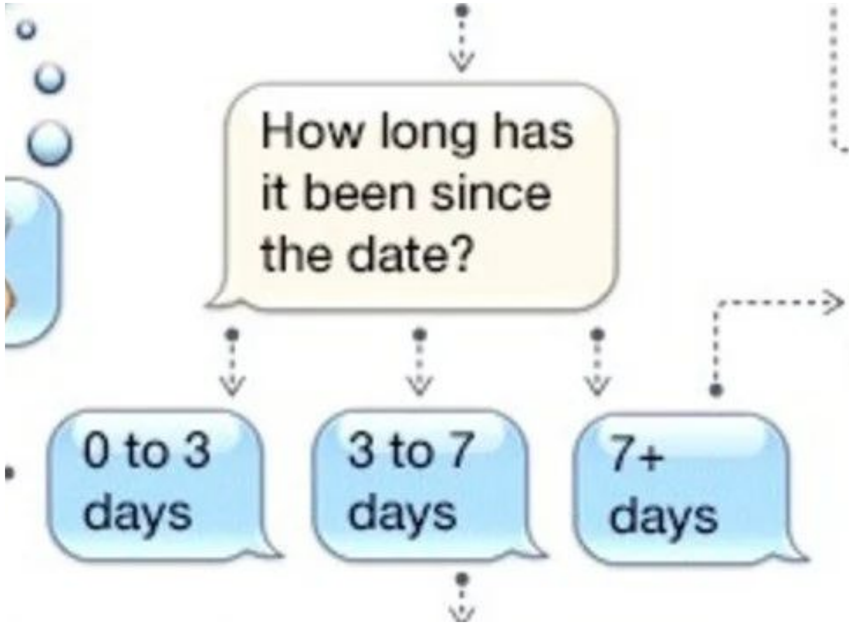
Multi-Way Decision

```
if <condition 1>:
  <case 1 statement>
elif <condition 2>:
  <case 2 statement>
elif <condition 3>:
  <case 3 statement>
else:
  <default statement>
```

Multi-Way Decision

```
#bad dating advice
days = eval(input("How many days has it been since your date? "))


if days <= 3:
  print("Wait to see if he texts.")
elif days <= 7:
  print("Call his mom to ask if he is alive.")
else:
  print("He was most likely abducted by aliens :( ")
```



Exams

A professor gives 100-point exams that are graded on the scale
90-100:A, 80-89:B, 70-79:C,
60-69:D, <60:F.

Write a program that accepts an exam score as input and uses a decision structure to calculate the corresponding grade.

And, Or

```
if numDays>1 and compatibility>90
  print("You should send them a text.")

if eggs==0 or bread==0
  print("You need to go to the grocery store.")
```

Canadian Politics

In Canada, in order to be a Member of Parliament (MP), you must be at least 18 years old, and a Canadian citizen.

In order to be a senator, you must be at least 30 years old and a Canadian citizen.

In order to be the Prime Minister, you must be appointed by the Governor General.

Write a program that accepts a user's age and citizenship, and outputs their eligibility to be an MP, Senator, and Prime Minister.

Conditional Program Execution

So far, we saw two types of Python modules.

1) Main programs that we run directly
2) Libraries that are imported and used by other programs (ex. math library, NumPy library, graphics library)

Sometimes, it may be necessary to create a hybrid module that can both run as a stand-alone program or can be imported as a library.

Conditional Program Execution

Sometimes, it may be necessary to create a hybrid module that can both run as a stand-alone program or can be imported as a library.

- Created a library, but also want to illustrate it with an example program
- Created a program, but also want it to be possible to use it as a library

Conditional Program Execution

In a program designed to be run directly OR imported, we may want to make running the main () program directly conditional.

```
if [module is being run directly]:
    main()
```

Conditional Program Execution

If a program is being run directly, Python creates a special variable called __name__ and sets its value to be "__main__"

i.e.   __name__ == "__main__"

```
if  __name__ == "__main__":
    main()
```

Conditional Program Execution

If a program is imported, the __name__ variable will be set to the name of the program file.

ex: import math
math.__name__ == "math"

```
def addInterest(balance, rate):
  newBalance = balance * (1+rate)
  return newBalance


def main():
  amount = 1000
  rate = 0.05
  amount = addInterest(amount,rate)
  print(amount)
```

```python
def addInterest(balance, rate):
  newBalance = balance * (1+rate)
  return newBalance

def main():
  amount = 1000
  rate = 0.05
  amount = addInterest(amount,rate)
  print(amount)
```

```python
if __name__ == '__main__':
  print("program is running directly")
  main()
else:
  print("program is not running directly")
```

Leap year

A year is a leap year if it is divisible by 4, unless it is a century year that is not divisible by 400. (1800 and 1900 are not leap years while 1600 and 2000 are.) Write a program that calculates whether a year is a leap year.

Midterm

Remember that the midterm is coming up on **Friday February 28**.

This week, I will post a list of programming problems that review some of the key concepts.