








Python 1 - Answer Key


Contents

-  Unit 1: Syntax and Sequencing Solutions
 - Welcome to Python Solutions
 - Sequencing Puzzles Part 1 Solutions
 - Python Conventions Solutions
 - Sequencing Puzzles Part 2 Solutions
 - Review and Quiz Solutions
 - Unit 1 Project: ASCII Art Solution
-  Unit 2: Loops and Patterns Solutions
 - For Loops and Indentation Solutions
 - For Loop Puzzles Solutions
 - More For Loops Solutions
 - Nested Loop Puzzles Solutions
 - Review and Quiz Solutions
 - Unit 2 Project: Pattern Maker Solution
-  Unit 3: Conditional Logic Solutions
 - Conditions Solutions
 - Conditional Puzzles Solutions
 - Else and Operators Solutions
 - More Conditional Puzzles Solutions
 - Review and Quiz Solutions
 - Unit 3 Project: Trivia Challenge Solutions
-  Unit 4: Conditional Loops Solutions
 - While Loops Solutions

- While Loops Puzzles Solutions
- More While Loops Solutions
- More While Loops Puzzles Solutions
- Review and Quiz Solutions
- Unit 4 Project: Rock Paper Scissors Solutions

-  Unit 5: Variables Solutions
 - Variables Introduction Solutions
 - Puzzle Set 1 Solutions
 - Input and Variables Solutions
 - More Variable Puzzles Solutions
 - Review and Quiz Solutions
 - Unit 5 Project: Mad Libs-Style Story Builder Solution

-  Unit 6: Data Types and Expressions Solutions
 - Introduction to Data Types Solutions
 - String Puzzles Solutions
 - Expressions Solutions
 - Expression Puzzles Solutions
 - Review and Quiz Solutions
 - Unit 6 Project: Make a Calculator Solution

-  Unit 7: Turtle Graphics Solutions
 - The Turtle API Solutions
 - More About Turtle Solutions
 - Review and Quiz Solutions
 - Unit 7 Project Solutions
 - Follow Up and Challenges Solutions

Answer Key



Unit 1: Syntax and Sequencing Solutions

- Welcome to Python

DIY: Your First Output Solution

SAMPLE Answer:

```
print("Sarah")
```

DIY: Use the print() Function Solution

SAMPLE Answer:

```
print('John Steinbeck')  
print('Leonardo da Vinci')  
print('Marie Curie')
```

Parentheses are Required Solution

```
print("hello")
```

Missing Symbols Solution

```
print("Oops.")
```

Valid Arguments Solution

```
print("hello")
```

Functions Must Exist Solution

```
print("hello")
```

Unseen Errors Solution

```
print("Hi,")  
print("this")  
print("is")  
print("a")  
print("program.")
```

DIY: Fix the Bugs, Print the Boat Solution

```
print("          /      \          ")
print("        /_____\ |          ")
print("      \      <<<      |      ")
print("    _____\_____\_____\_____ ")
print("~~~~~")
```

DIY: Fix the Haiku Solution

```
print("The computer gods")
print("Spoke upon man, hello world")
print("When it all began.")
```

● Sequencing Puzzles Part 1 Solutions

Inside the Computer

```
forward()
forward()
```

More Memory Chips

```
forward()
forward()
forward()
```

Around the Corner

```
forward()
turn_right()
forward()
```

Scale the Slope

```
stun()
forward()
forward()
forward()
stun()
forward()
forward()
forward()
```

- Python Conventions Solutions

Fix This Code! (1)

```
# modify these names to be snake_case
print("test")
shoe_size = 5
my_age = 13
def get_my_age():
    return my_age
```

Fix this Code! (2)

```
# this does not work
my_var = 4
print(my_var)
```

Do It Yourself

```
my_first_var = 5
my_second_var = 8
cupcake = 9

print(my_first_var)
print(my_second_var)
print(cupcake)
```

- Sequencing Puzzles Part 2 Solutions

Debugging Refresher

```
print("Hi")
print("There")
```

Fix the Bugs

```
forward()
forward()
forward()
turn_right()
forward()
forward()
forward()
turn_right()
```

```
forward()  
forward()  
forward()
```

Turning to Safety

```
turn_left()  
stun()  
forward()  
turn_right()  
forward()  
forward()  
turn_left()  
forward()
```

Around the Bend

```
forward()  
forward()  
forward()  
turn_left()  
forward()  
forward()  
forward()  
turn_left()  
forward()  
forward()  
forward()
```

Bunny Hop

```
# an enemy is in the way, get rid of it first  
stun()  
jump()  
jump()
```

Left-to-Right

```
stun()  
forward()  
forward()  
turn_left()  
stun()  
jump()  
turn_right()  
stun()
```

```
jump()  
turn_left()  
forward()  
forward()
```

- Review and Quiz Solutions

Question 1: What is a value?

Answer: Both of these

Question 2: An identifier cannot begin with a number. Which of the following is NOT a valid identifier?

Answer: 1num_

Question 3: An identifier can begin with an underscore ("_").

Answer: True

Question 4: An identifier can begin with a number.

Answer: False

Question 5: Which of the following is a valid function call?

Answer: forward()

Question 6: How would you call a function named my_function?

Answer: my_function()

Question 7: Identifiers in Python are case-sensitive.

Answer: True

Question 8: 78090 is a valid identifier in Python.

Answer: False

Question 9: You are not able to call the same function twice in a row.

Answer: False

Question 10: You can replace the parentheses on a function call with square brackets.

Answer: False

Question 11: The parentheses at the end of a function call are optional.

Answer: False

Question 12: Which of the following is the correct way to start a multi-line comment?

Answer: ""

Question 13: Commented lines affect the execution of your program.

Answer: False

Question 14: Print the name of the country you live in.

Answer:

```
print('United States')
```

Question 15: Use print instructions to output a 3 x 3 square of asterisk (*) characters.

Answer:

```
print("* * *")
print("* * *")
print("* * *")
```

- Unit 1 Project: ASCII Art Solution

Sample Solution:

```
# Print a 3D object
print("
print(" /_/_/_/_/_/_/_/_/_/ ")
print(" /_/_/_/_/_/_/_/_/_/ ")
print(" |_'_'_'_'_'_|/ ")

# Print a sunset
print(" ") # sky
print(" \ ' / ") # sun
print(" .-. ^ ^ ") # wave
print(" -= ( ) =- ^ ^ ") # horizon
print(" ~~~^~^~==~==~^~^~^~^~^~ ") # ocean
print(" ~~~^~^~==^~^~^~^~^~^~^~ ")
print(" :.:.:.:.:.:.:.:.:.:.:. ") # beach
print(" :.:.:.:.:.:.:.:.:.:. ")
```



Unit 2: Loops and Patterns Solutions

- For Loops and Indentation Solutions

Path to Safety

```
forward()
#add more forward() commands here
forward()
forward()
forward()
forward()
forward()
forward()
forward()
forward()
```

Do It Yourself!

```
for i in range(20, -1, -2):
    print(i)
```

Do It Yourself

```
for i in range(9):
    print(i)
```

● For Loop Puzzles Solutions

Do It Yourself

```
for i in range(10):
    print('Hello, World!')
```

Path To Safety

```
for i in range(8):
    forward()
```

Forward, Stun, Repeat

```
for i in range(8):
    stun()
    forward()
```

Make a Zig-Zag

```
for i in range(5):
    forward()
```

```
turn_right()
forward()
turn_left()
```

March On

```
for i in range(4):
    forward()
turn_right()
for i in range(6):
    forward()
```

- More For Loops Solutions

What Are Arithmetic Operators?

SAMPLE Answer:

```
print(2+2)
print(10-8)
print(5*4)
print(8/2)
print(100%5)
print(9**2)
print(10//3)
```

Range Applications

Power of 2

```
n = 4 # try changing this when you're ready!
# write your loop here
output = 1
for i in range(n):
    output = output * 2
print(output)
```

Teen Numbers

```
for i in range(13, 20):
    print(i)
```

Factorial

```
n = 4 # try changing this when you're ready!
# write your loop here
```

```
for i in range(n - 1, 0, -1):
    n = n * i
print(n)
```

What Are Nested loops

Do It Yourself

```
for i in range(11):
    for j in range(11):
        print(i+j)
```

Nested Loops Examples

Multiplication Table

```
for i in range(13):
    for j in range(13):
        print(str(i) + " * " + str(j) + " = " + str(i * j))
```

Triangle Prints

SAMPLE Answer:

```
for i in range(10):
    for j in range(10):
        print("*" * j)
```

● Nested Loop Puzzles Solutions

Jump, Forward

```
for i in range(3):
    jump()
    stun()
    forward()
```

Forward, Jump

```
for i in range(3):
    stun()
    forward()
    jump()
```

Up and Down

```
for i in range(3):
    stun()
    forward()
    turn_left()
    forward()
    turn_right()
jump()
for i in range(3):
    turn_right()
    stun()
    forward()
    turn_left()
    forward()
```

U-Turn

```
for i in range(3):
    # you need a second loop here
    for i in range(5):
        forward()
    turn_left()
```

● Review and Quiz Solutions

Question 1: How many times will function1 be called?

```
for i in range(10):
    for j in range(10):
        function1()
```

Answer: 100

Question 2: Which for loop will move your character 5 times?

Answer:

```
for i in range(1,6):
    forward()
```

Question 3: How many blocks will the character move using the following code?

```
for i in range(1, 5):
    forward()
```

Answer: 4

Question 4: Which of the following is properly indented code?

Answer:

```
for i in range(10, 5, 1):  
    forward()
```

● Unit 2 Project: Pattern Maker Solution

This is a creative project, so anything that follows the guidelines should receive credit.

Requirements

Just follow these guidelines:

- The program should use for loops and all 3 movement commands you have learned, `forward()`, `turn_right()`, and `turn_left()`.
- The program should also use the new `light_on()` and `light_off()` commands to create your pattern.
- The program should be creative and your own original idea.
- The program should involve some complicated coding.
- Add comments to the top of your program. Make sure you add:
 - Both of your names
 - The purpose of the program (describe what it does)



Unit 3: Conditional Logic Solutions

● Conditions Solutions

If Statements

Negative or Not?

```
x = 5  
if x < 0:  
    print("x is negative")
```

The output did change here, the program no longer prints anything when x is positive.

Password Panic

```
password = input("Enter the password: ")
```

```
if password == "secret":
    print("Welcome!")
```

Students may modify this code to use their own (fake) secret password.

Are You 100?

```
age = 100
if (age == 100):
    print("Happy hundredth!")
```

“Fun(d)” Raising

```
name = input("Name: ")
donation = input("Donation: ")

if (name != "" and donation != 0):
    print("Thank you " + name + " for your $" + str(donation) + " donation!")
if (name == ""):
    print("invalid name")
if (donation == 0):
    print("Donation required")
```

Branching

Are you (a)Cute, just Right, or just plain Obtuse?

```
angle = 89

if (angle < 90):
    print("acute")
if (angle == 90):
    print("right")
if (angle > 90):
    print("obtuse")
```

Debugging if Statements

Exercise 1

```
x = 2
if x > 0:
    print("positive number")
```

Exercise 2

```
age = 18
if age >= 18:
    print("go vote!")
```

Exercise 3

```
x = 28
if x >= 10 and x < 100:
    print("two digit number")
```

Exercise 4

```
if 1 < 2:
    print("first")
    print("second")

if 5 > 4:
    print("a")
    print("b")
```

Exercise 5

```
if 1 == 1:
    # no instruction
    print("instruction")
```

Mutually Exclusive Conditions

"What a Man Can Do and What a Man Can't Do"

```
age = 15

if (age < 16):
    print("cannot drive")
if (age < 18):
    print("cannot vote")
if (age < 25):
    print("cannot rent a car")
if (age >= 25):
    print("can do everything")
```

Even One Out

```
x = 11

if (x % 2 == 0):
    print("even")
if (x % 2 == 1):
    print("odd")
```

- Conditional Puzzles Solutions

Path Right

Use a "for" loop and a conditional to reach the goal

```
# set up the loop
for i in range(9):
    if has_path_right():
        turn_right()
    forward()
```

Path Left

```
# use a "for" loop and a conditional to reach the goal
# set up the loop
for i in range(7):
    if has_path_left():
        turn_left()
    forward()
```

More Turns!

```
# use conditionals and a "for" loop to get to the goal
for i in range(10):
    if has_path_right():
        turn_right()
    if has_path_left():
        turn_left()
    forward()
```

Hazardous Path

```
for i in range(10):
    if has_path_right():
        turn_right()
    if has_path_left():
```



```
    turn_left()
    if enemy_in_sight():
        stun()
    forward()
```

● Else and Operators Solutions

else Statement Syntax

Lucky (and Unlucky) Numbers

Both lucky and normal are printed because the first if statement is separate from the else. Only the second if statement is connected to the else statement.

Don't Forget the Colon!

```
n = 18
if n % 7 == 0:
    print("n is a multiple of 7")
else:
    print("n is not a multiple of 7")
```

Else Indentation

```
hungry = True

if hungry:
    print("Eat something!")
else:
    print("Go exercise!")
```

Poorly Indented Instructions

```
tired = False

if not tired:
    print("Write some code!")
    print("Eat some food!")
else:
    print("Take a nap!")
```

elif Statements

Grading Letters

```
grade = int(input("What grade did you get?"))

if grade >= 98:
    print('You got an A+')
elif grade >= 94:
    print('You got an A')
elif grade >= 90:
    print('You got an A-')
elif grade >= 87:
    print('You got a B+')
elif grade >= 83:
    print('You got a B')
elif grade >= 80:
    print('You got a B-')
elif grade >= 77:
    print('You got a C+')
elif grade >= 73:
    print('You got a C')
elif grade >= 70:
    print('You got a C-')
elif grade >= 67:
    print('You got a D+')
elif grade >= 63:
    print('You got a D')
elif grade >= 60:
    print('You got a D-')
else:
    print('You got an F')
```

Nested if Statements

Jack and Jill Went Up the Rock

```
jack = "rock"
jill = "rock"

if (jack == "rock"):
    if (jill == "rock"):
        print("tie")
    elif (jill == "paper"):
        print("Jill wins")
    else:
        print("Jack wins")
elif (jack == "paper"):
```

```

    if (jill == "rock"):
        print("Jack wins")
    elif (jill == "paper"):
        print("tie")
    else:
        print("Jill wins")
else:
    if (jill == "rock"):
        print("Jill wins")
    elif (jill == "paper"):
        print("Jack wins")
    else:
        print("tie")

```

Play Against a Computer

Add Conditional Statements

```

import random

human = input("Enter rock, paper, or scissors: ")
computer = random.choice(["rock", "paper", "scissors"]) # picks randomly
from rock, paper, and scissors

# Compare the random choice with your guess
if (human == computer):
    print('tie')
elif ((human == 'rock' and computer == 'paper') or (human == 'paper' and
computer == 'scissors') or (human == 'scissors' and computer ==
'rock')):
    print('computer wins')
else:
    print('human wins')

```

Secret Numbers

```

import random

secret = 7
guess = 5

if (guess < secret):
    print("too small")
elif (guess > secret):
    print("too big")

```

```
else:
    print("correct")
```

Guessing Game

```
secret = 7

for i in range(3):
    guess = int(input("guess: "))

    if (guess < secret):
        print("too low")
    elif (guess > secret):
        print("too big")
    else:
        print("correct")
        break

if guess != secret:
    print("game over")
```

- More Conditional Puzzles Solutions

Death Valley

```
for i in range(22):
    if enemy_in_sight():
        stun()
    elif(has_path_left()) :
        turn_left()
    forward()
```

Traps

```
for i in range(24):
    if has_path_left():
        turn_left()
        forward()
    elif has_path_right():
        turn_right()
        forward()
    elif is_gap_ahead():
        jump()
    else:
        forward()
```

G for Grandeur

```
for i in range(13) :
    if is_gap_ahead():
        jump()
    elif has_path_left():
        turn_left()
        forward()
    else:
        forward()
```

Silly Path

```
for i in range(18):
    if has_path_left():
        turn_left()
        forward()
    elif has_path_right():
        turn_right()
        forward()
    elif (is_gap_ahead()):
        jump()
    else:
        forward()
```

● Review and Quiz Solutions

If / Else

Two for Two

Question: Why does the string "x is a variable" always print, regardless of the value you assign to x?

Answer: Because the print function for the "x is a variable" string is located outside of the If / Else statement and therefore will run every time.

Nested Ifs

Is X One?

Question: How would you modify this program to check if x is equal to 1 *only if* x is less than 5? For example, if x is 6, your program will not check if x is equal to 1.

Answer: You would modify the program by nesting the if statement that checks to see if x is equal to 1 inside of the if statement that checks if it is less than 5.

Are You A Teen Voter?

Question: Actually, one of this program's conditions is slightly redundant. Which one is it, and can you modify it such that it doesn't check (part of) the same condition twice?

Answer: When there is a second check to see if the age ≥ 18 inside the else, this is partially redundant. The first if statement established that they are not a teenager, so it is sufficient to check if age ≥ 20 in the if statement inside the else.

Quiz

Question 1: Write a program that checks if two variables x and y are equal. Declare both variables and try numbers that are equal and some that are not.

Answer:

```
# Write your code here.
x = 5
y = 7

if (x == y):
    print("equal")
```

Question 2: Write a program to determine a letter grade based on the percentage. For example, if the percentage is above 90 then the letter grade is an A.

Answer:

```
# Write your code here.
grade = 91

if grade >= 90:
    print("You got an A.")
elif grade >= 80:
    print("You got a B.")
elif grade >= 70:
    print("You got a C.")
elif grade >= 60:
    print("You got a D.")
elif grade < 60:
    print("You got an F.")
```

Question 3: A leap year has 366 days (the extra day is February 29th).

- If the year is divisible by 4 but not 100, it is a leap year.
- If the year is divisible by 400, it is a leap year.

Print "leap year" if the given year is a leap year, and "not leap year" otherwise.

Answer:

```
# Write your code here.
year = 2001

if (year % 400 == 0 or (year % 4 == 0 and year % 100 != 0)):
    print("leap year")
else:
    print("not leap year")
```

Question 4: Given the number of older and younger siblings, print

- "only child" if the person has no siblings
- "oldest child" if the person only has younger siblings
- "youngest child" if the person only has older siblings
- "middle child" if the person has both younger and older siblings

Answer:

```
# Write your code here.
older_siblings = 1
younger_siblings = 1

if (older_siblings == 0 and younger_siblings == 0):
    print("only child")
elif (older_siblings == 0 and younger_siblings > 0):
    print("oldest child")
elif (younger_siblings == 0 and older_siblings > 0):
    print("youngest child")
else:
    print("middle child")
```

Question 5: Which part of a conditional goes after the if keyword and before the colon?

Answer: The condition

Question 6: When would you use an "if-else" statement over an "if" statement?

Answer: When you want to run different code if a condition is false

Question 7: There is no difference between an "if" statement and an "if-else" statement.

Answer: False

Question 8: If you wanted to run different code given different conditions, how would you do that?

Answer: Use elif statements

Question 9: Given two variables describing a quadrilateral, a width and a length, write a program to determine if the quadrilateral is a square or rectangle.

Answer:

```
# Write your code here.
width = 10
length = 12

if (width == length):
    print("This quadrilateral is a square.")
else:
    print("This quadrilateral is a rectangle")
```

Question 10: Given a variable, num, write a program to determine if num is a multiple of 3, or of 5, or both.

Answer:

```
# Write your code here.
num = 15

if (num % 3 == 0 and num % 5 == 0):
    print("Divisible by both 3 and 5")
elif (num % 3 == 0):
    print("Divisible by 3")
elif (num % 5 == 0):
    print("Divisible by 5")
else:
    print("Try again")
```

- Unit 3 Project: Trivia Challenge Solutions

Because this is a creative task, student work will vary. When assessing student work, be sure that the program is a trivia game that has at least 3 questions. In addition, the program should count the number of correct answers.



Unit 4: Conditional Loops Solutions

- While Loops Solutions

Do It Yourself

```
i = 1
while (i < 10):
    if (i % 2 == 0):
        print(str(i) + " is even")
    else:
```



```
    print(str(i) + " is odd")
    i = i + 1
```

Components of a While loop

Do It Yourself

```
number = 10

while number > 0:
    print(number)
    number = number - 1

print("Lift off!")
```

While Loops in Action

Do It Yourself

```
year = 1960

while (year <= 2022):
    print(year)
    year = year + 1
```

Testing Out While Loops

Print Hello

```
n = 4

while n > 0:
    print("hello")
    n = n - 1
```

Bottles of Root Beer

```
bottles = 10

while bottles:
    print(str(bottles) + " bottles of root beer on the wall,")
    print(str(bottles) + " bottles of root beer!")
    print("Take one down, pass it around,")
    # your code here
```

```
print(str(bottles) + " bottles of root beer on the wall.")
bottles = bottles - 1
```

Multiple Choice Exercise

Question: What is the output of the following code?

```
x = 1
y = 0

while x ** 2 < 10:
    x = x + 1
    y = y + 1
    print(y)
```

Answer:

1
2
3

● While Loops Puzzles Solutions

Detect a Path

```
while has_path_ahead():
    forward()
```

Forward, Left

```
# keep going until the goal is reached
while not reached_goal():
    # check if there is a path to the left and turn left
    if (has_path_left()):
        turn_left()
        forward()
    else:
        forward()
```

Right, Forward

```
while not reached_goal():
    if has_path_ahead():
        forward()
    else:
        turn_right()
```

Debug Challenge

```
while not reached_goal():
    if has_path_left():
        turn_left()
        forward()
    else:
        forward()
```

- More While Loops Solutions

While / Else and For / Else

Do It Yourself

```
i = 10

while i > 0:
    print(i)
    i = i - 1
else:
    print("Blast off!")
```

Accumulating Results

Product of Multiples of Threes

```
number = 3
total = 1

while(number < 100):
    print(number)
    total = total * number
    number = number + 3

print(total)
```

Cash Register

```
change = 49
total = 0

quarters = 0
dimes = 0
nickels = 0
pennies = 0
```

```

while total < change:
    if (change - total > 25):
        quarters += 1
        total += 25
    elif (change - total > 10):
        dimes += 1
        total += 10
    elif (change - total > 5):
        nickels += 1
        total += 5
    else:
        pennies += 1
        total += 1

print("change: " + str(change))
print("quarters: " + str(quarters))
print("dimes: " + str(dimes))
print("nickels: " + str(nickels))
print("pennies: " + str(pennies))

```

Candy Distribution

```

total_candy = 1000
candy_given = 0
friends = 0

while (candy_given <= total_candy):
    friends = friends + 1
    candy_given = candy_given + friends**2

#The last friend didn't get enough candy
friends = friends - 1
print("You can give candy to " + str(friends) + " friend(s).")

```

Branching in a While Loop

Fizz Buzz

```

i = 1

while i <= 100:
    if (i % 3 == 0 and i % 5 == 0):
        print("fizzbuzz")
    elif (i % 3 == 0):
        print("fizz")
    elif (i % 5 == 0):

```

```
        print("buzz")
    else:
        print(i)
    i += 1
```

Collatz Conjecture

```
n = 10

while n > 1:
    print(n)
    if (n % 2 == 0):
        n = n / 2
    else:
        n = n * 3 + 1
```

LCM of 3

```
a = 3
b = 6
c = 9

a_mult = a
b_mult = b
c_mult = c

while a_mult != b_mult or b_mult != c_mult:
    if a_mult < b_mult or a_mult < c_mult:
        a_mult = a_mult + a
    elif b_mult < a_mult or b_mult < c_mult:
        b_mult = b_mult + b
    elif c_mult < a_mult or c_mult < b_mult:
        c_mult = c_mult + c

print("The LCM is " + str(a_mult))
```

More on Break

Is It Prime?

```
n = 23

factor = 2

while factor < n:
    if n % factor == 0:
        print("not prime")
```

```
        break

    factor = factor + 1
else:
    print("prime")
```

Factorial

```
n = 9
factorial = n

while (n > 1):
    n -= 1
    factorial = factorial * n
    if factorial > 1000000:
        print("too big")
        break
else:
    print(factorial)
```

Prime Every Factor

```
n = 120
factor = 2

while n > 1:
    if n % factor == 0:
        print(factor)
        n = n / factor
        continue
    factor = factor + 1
```

- More While Loops Puzzles Solutions

While Loop Refresher

Test it Out

SAMPLE Fix:

```
number = 1
while number < 10:
    print(number)
    number += 1
```

Path Ahead?

```
for i in range(3):
    jump()
    while has_path_ahead():
        forward()
```

Left Spiral

```
while not reached_goal():
    if has_path_left():
        turn_left()
        forward()
    elif (is_gap_ahead()):
        jump()
    else:
        forward()
```

Right Spiral

```
while not reached_goal():
    if has_path_right():
        turn_right()
        forward()
    elif (is_gap_ahead()):
        jump()
    else:
        forward()
```

Choose a Path

```
forward()
forward()
forward()

while not reached_goal():
    if has_path_left():
        turn_left()
        forward()
    elif has_path_right():
        turn_right()
        forward()
    elif is_gap_ahead():
        jump()
    else:
        forward()
```

- Review and Quiz Solutions

Question 1: While loops always terminate eventually.

Answer: False

Question 2: What is the value of x after the following code is executed?

```
x = 0
while x < 20:
    x = x + 1
else:
    x = 10
```

Answer: 10

Question 3: What is the output of this code?

```
i = 1
while (i < 1):
    print("hello")
```

Answer: It won't output anything

Question 4: What does the following code do?

```
while True:
    print("hello")
```

Answer: It prints 'hello' infinitely

- Unit 4 Project: Rock Paper Scissors Solutions

More Random Commands

SAMPLE Answer:

```
import random
# iterate 10 times
for i in range(10):
    color = random.choice(['red', 'green', 'blue', 'orange', 'purple',
'yellow', 'pink']) # Add more choices to the list
    print(color)
```

Challenge: Rock Paper Scissors

```
import random

computer_choice = random.choice(['rock', 'paper', 'scissors']) # change
this line
print('the computer chooses ' + computer_choice)
```


Challenge: Rock Paper Scissors 2

```
import random

computer_choice = random.choice(['rock', 'paper', 'scissors']) # change
this line
print('the computer chooses ' + computer_choice)
human_choice = input('enter human choice: ') # change this line
print('human chooses ' + human_choice)
```

Challenge 3: Rock Paper Scissors 3

```
import random

computer_choice = random.choice(['rock', 'paper', 'scissors']) # change
this line
print('the computer chooses ' + computer_choice)
human_choice = input('enter human choice: ') # change this line
print('human chooses ' + human_choice)

if (human_choice == computer_choice):
    print('tie')
elif ((human_choice == 'rock' and computer_choice == 'paper') or
      (human_choice == 'paper' and computer_choice == 'scissors') or
      (human_choice == 'scissors' and computer_choice == 'rock')):
    print('computer wins')
else:
    print('human wins')
```

Challenge 4: Rock Paper Scissors

```
import random

keep_playing = True
while keep_playing:
    # code that plays one round**
    computer_choice = random.choice(['rock', 'paper', 'scissors']) #
change this line
    print('the computer chooses ' + computer_choice)
    human_choice = input('enter human choice: ') # change this line
    print('human chooses ' + human_choice)

    if (human_choice == computer_choice):
        print('tie')
    elif ((human_choice == 'rock' and computer_choice == 'paper') or
          (human_choice == 'paper' and computer_choice == 'scissors') or
          (human_choice == 'scissors' and computer_choice == 'rock')):
        print('computer wins')
```

```

else:
    print('human wins')

print("Do you want to play again?")
answer = input()
if answer == "no":
    keep_playing = False;
    print("Thanks for playing!")

```

Challenge 5: Rock Paper Scissors

```

import random

computer_score = 0
human_score = 0
number_of_ties = 0

keep_playing = True
while keep_playing:
    # code that plays one round**
    computer_choice = random.choice(['rock', 'paper', 'scissors']) #
change this line
    print('the computer chooses ' + computer_choice)
    human_choice = input('enter human choice: ') # change this line
    print('human chooses ' + human_choice)

    if (human_choice == computer_choice):
        number_of_ties += 1
        print('tie')
    elif ((human_choice == 'rock' and computer_choice == 'paper') or
(human_choice == 'paper' and computer_choice == 'scissors') or
(human_choice == 'scissors' and computer_choice == 'rock')):
        computer_score += 1
        print('computer wins')
    else:
        human_score += 1
        print('human wins')

print("Do you want to play again?")
answer = input()
if answer == "no":
    keep_playing = False;
    print("Thanks for playing!")

print('computer score: ' + str(computer_score))
print('human score: ' + str(human_score))
print('ties: ' + str(number_of_ties))

```



Unit 5: Variables Solutions

- Variables Introduction Solutions

Variables

Variables On Your Own

SAMPLE Answer:

```
age = 12
```

Naming Variables

Exercise 1

```
awesome_sauce = "bbq"
```

Exercise 2

```
three_musketeers = "a, b, and c"
```

Exercise 3

```
no_dashes = 1  
NOPUNCTUATION = 12
```

Fix the Variable Names

```
awesome_variable = 5  
dont_use_dashes = 101  
one_hundred_and_one_dalmatians = "test"
```

Sing It, Python Style

Students can change the song lyrics any way they want

Reassign a Variable

Students can change the names to anything they want

Question: What are the final values in the variables a and b?:

```
a = 10
```

```
b = 15
```

```
b = a
```

```
a = b
```

Answer:

a = 10

b = 10

- Puzzle Set 1 Solutions

Fuel Up

```
energy = 30

forward()
turn_left()
forward()
forward()
```

Energy Loop

```
energy = 30

for i in range(3):
    forward()
```

Stockpile

```
energy = 100

while not reached_goal():
    if has_path_left():
        turn_left()
    elif has_path_right():
        turn_right()
    forward()
```

Rocket Jump

```
energy = 150

while not reached_goal():
    if has_path_left():
        turn_left()
    elif is_gap_ahead():
        jump()
    forward()
```

- Input and Variables Solutions

Combining Values

Best Friend

If you remove the space after `is`, the program will print the message like this:

```
"My best friend isSarah"
```

Many Variables

Students can change the values to their own city, state, and country

Wacky Sentence

SAMPLE Answer:

```
noun = 'dog'
place = 'store'
verb = 'eat'

print('The ' + noun + ' goes to the ' + place + ' to ' + verb + '.')
```

The input Function

Do It Yourself Story

```
character = input("Enter a character: ")
place = input("Enter a place: ")
action = input("Enter an action: ")

print("The " + character + " went to the " + place + " and " + action +
      ".")
```

Multiple Assignments

Practice Multiple Assignment Swapping, Part 1

```
# Write your solution using the string variables a and b defined below
a = "fox"
b = "trot"

a, b = a + b, b + a

print(a)
print(b)
```

Practice Multiple Assignment Swapping, Part 2

```
# Write your solution using string variables a and b defined below
a = "fox"
b = "trot"

a, b = a + b, a + b

print(a)
print(b)
```

- More Variable Puzzles Solutions

Let Loose

```
power = 1
energy = 70

for i in range(get_enemy_health()):
    stun()
for i in range(7):
    forward()
```

It's a Toughie

```
power = 3
energy = 70

for i in range(get_enemy_health()):
    stun()
for i in range(7):
    forward()
```

Welcome to the Computer

```
energy = 170
power = 11

while not reached_goal():
    while get_enemy_health():
        stun()
    if has_path_left():
        turn_left()
    elif has_path_right():
```

```
turn_right()
forward()
```

Obstacle Course

```
energy = 200
power = 6

while not reached_goal():
    while get_enemy_health():
        stun()
    if is_gap_ahead():
        jump()
    if has_path_right():
        turn_right()
    elif has_path_left():
        turn_left()
    forward()
```

● Review and Quiz Solutions

Question 1: Which of the following is the correct way to declare the “x” variable?

Answer: x = 0

Question 2: The following pieces of code are equivalent.

Answer: True

Question 3: The following is correct syntax:

Answer: False

Question 4: You can create a variable named "10thvar".

Answer: False

Question 5: Select all valid variable names.

Answer:

```
go_go
shape
```

Question 6: What is the output of the following?

Answer: 15

Question 7: Which of the following is the correct way to declare and initialize 2 variables on the same line?

Answer:

```
a, b = 1, 2
```

Question 8: You can declare and initialize a variable with a single expression.

Answer: True

Question 9: You can create a variable named var.

Answer: True

Question 10: You must declare a variable every time you use it.

Answer: False

Question 11: Why use variables?

Answer: Variables are used to store and retrieve data or information.

Question 12: Only numbers can be stored in a variable.

Answer: False

Question 13: Once you assign a number to a variable, you can no longer assign a string to the same variable.

Answer: False

Question 14: What is the final value stored in the variable z?

Answer: 35

Question 15: Store your birthday as integers in a month, day, and year variable. For example: If you were born on January 22nd 1993, you would store 1 in month, 22 in day, and 1993 in year.

Answer:

```
# Write your code here.  
month = 1  
day = 22  
year = 1993
```

Question 16: Use a multiple assignment to set a variable a to 5 and a variable b to 10 using only one expression.

Answer:

```
# Write your code here.  
a, b = 5, 10
```

Question 17: Given an `english` string and its `spanish` translation, print a new sentence that has the form "english is spanish in Spanish."

Answer:


```
# Write your code here.
english = "Hello"
spanish = "Hola"

print(english + " is " + spanish + " in Spanish.")
```

- Unit 5 Project: Mad Libs-Style Story Builder Solution

Sample project solution:

```
adjective = input('enter adjective: ')
color = input('enter a color name: ')
thing = input('enter a thing name: ')
place = input('enter a place name: ')
person= input('enter a person name: ')
adjectivel = input('enter a adjective: ')
insect= input('enter a insect name: ')
food = input('enter a food name: ')
verb = input('enter a verb: ')

print('Last night I dreamed I was a ' + adjective + ' butterfly with ' +
color + ' splotches that looked like a ' + thing + '. I flew to ' +
place + ' with my best friend and ' + person + ' who was a ' +
adjectivel + ' ' + insect + '. We ate some ' + food + ' when we got
there and then decided to ' + verb + ' and the dream ended when I
said--let\'s ' + verb + '!')
```



Unit 6: Data Types and Expressions Solutions

- Introduction and to Data Types Solutions

Integers

Add Numbers

```
one_cent = 10
two_cents = 5

print(one_cent + 2 * two_cents)
```

String Operators

DIY: Concatenating Words

```
adjective = "slippery"  
noun = "hippo"  
  
print(adjective + " " + noun)
```

String Methods

DIY: Playing with Substrings

```
str0 = "The nice dog jumped over the cat"  
  
str1 = str0[9:19].upper()  
print(str1)
```

- String Puzzles Solutions

Hack the Trap 1

```
L1 = 'green'
```

Hack the Trap 2

```
L1 = 'blue'  
L2 = 'green'
```

Hack the Trap 3

```
L1 = 'blue'  
L2 = 'green'  
L3 = 'red'  
L4 = 'yellow'
```

- Expressions Solutions

All of the exercises in this chapter are here so that students can test different expressions in the code editor.

- Expression Puzzles Solutions

Trickier Trap 1

```
L1 = 5 * x
```

Trickier Trap 2

```
L1 = 7 * x ** 5
```

Trickier Trap 3

```
L1 = x ** 2 + 2 * x + 1
```

Trickier Trap 4

```
L1 = (3 * x - 2) / (1 + 2 * x) ** 2
```

- Review and Quiz Solutions

Question 1: True or false: While loops always terminate eventually.

Answer: False

Question 2: What is the value of x after the following code is executed?

Answer: 10

Question 3: What is the output of this code?

Answer: It won't output anything

Question 4: What would the following code do?

Answer: print('hello') infinitely

- Unit 6 Project: Make a Calculator Solution

```
print("Select operation.")
print("1.Add")
print("2.Subtract")
print("3.Multiply")
print("4.Divide")

result = 0
```

```

while True:
    # Take input from the user
    choice = int(input("Operation: "))

    num1 = input("Enter first number: ")
    if (num1 == "previous"):
        num1 = result
    else:
        num1 = float(num1)
    num2 = float(input("Enter second number: "))
    if (num2 == "previous"):
        num2 = result
    else:
        num2 = float(num2)

    if choice == 1:
        result = num1 + num2
        print(num1, "+", num2, "=", result)

    elif choice == 2:
        result = num1 - num2
        print(num1, "-", num2, "=", result)

    elif choice == 3:
        result = num1 * num2
        print(num1, "*", num2, "=", result)

    elif choice == 4:
        result = num1 / num2
        print(num1, "/", num2, "=", result)
    else:
        print("Didn't get that, try again!")
    previous = result

```



Unit 7: Turtle Graphics Solutions

- The Turtle API Solutions

Setting up the Scene

```

import turtle

screen = turtle.Screen()

```

```
screen.bgcolor("red")
```

Creating Turtles

```
import turtle # needed to use turtles
screen = turtle.Screen() # get the Turtle Screen
t = turtle.Turtle() #create a new turtle
t.shape("square")
```

Moving and Drawing with the Turtle

All code for this exercise is given in the project tutorial

- ## More about Turtle Solutions

Turtle Graphing

All code for this exercise is given in the project tutorial

Introduction to Lists

Do It Yourself: Ice Cream Flavors

```
# Write your ice cream flavors list here
ice_cream_flavors = ['vanilla', 'chocolate', 'strawberry', 'mint chip']

for flavor in ice_cream_flavors:
    print(flavor)
```

Turtle Drawing Using Lists

Exercise 1

```
import turtle
screen = turtle.getscreen() # get a Turtle Screen
screen.bgcolor("white") # You can change "white" to something else
spot = turtle.Turtle() # my Turtle is named spot
spot.shape("classic")

color_list = ["red", "orange", "blue", "yellow", "cyan", "purple",
"pink", "green"]
for c in color_list:
    spot.color(c)
    spot.forward(50)
    spot.right(45)
```

Exercise 2

```
import turtle

screen = turtle.getscreen()

screen.bgcolor((0, 0, 0))

spot = turtle.Turtle()
spot.shape("classic")
spot.color(255, 255, 255)

spot.forward(50)
```

Making Random Art

Exercise 1

```
import turtle
import random
screen = turtle.Screen()
screen.bgcolor(0, 0, 0)
t = turtle.Turtle()
t.shape("circle")
t.speed(0)
t.color("white")
t.penup()

for i in range(1000):
    t.forward(10)
    t.left(random.randint(90, 270))
    r = random.randint(0, 255) # change this to be a randint between 0
and 255
    g = random.randint(0, 255) # change this to be a randint between 0
and 255
    b = random.randint(0, 255) # change this to be a randint between 0
and 255
    # set the turtle's new color
    t.color(r, g, b)
    t.dot(25) # make dot with a radius of 25 pixels
```

Exercise 2

```
import turtle
import random
screen = turtle.Screen()
screen.bgcolor(0, 0, 0)
t = turtle.Turtle()
```

```

t.shape("circle")
t.speed(0)
t.color("white")
t.penup()

for i in range(1000):
    # inside the for loop
    if t.xcor() < -200 or t.xcor() > 200 or t.ycor() < -200 or t.ycor()
    > 200:
        t.goto(0,0)
        t.forward(10)
        t.left(random.randint(90, 270))
        r = random.randint(0, 255) # change this to be a randint between 0
and 255
        g = random.randint(0, 255) # change this to be a randint between 0
and 255
        b = random.randint(0, 255) # change this to be a randint between 0
and 255
        # set the turtle's new color
        t.color(r, g, b)
        t.dot(25) # make dot with a radius of 25 pixels

```

● Review and Quiz Solutions

Question 1: True or False : You need to import the turtle module before you can use the Turtle Tool.

Answer: True

Question 2: When you first create a turtle, it starts at (0,0) on the screen. Where will the turtle be if we execute the following code?

Answer: (100, 0)

Question 3: When you first create a turtle, it starts at (0,0) on the screen. Where will the turtle be if we execute the following code?

Answer: (0, -100)

Question 4: When you first create a turtle, it starts at (0,0) on the screen. Where will the turtle be if we execute the following code?

Answer: (100, 100)

Question 5: What function do you use to get the turtle to start drawing?

Answer:

```

turtle.pendown()

```

Question 6: What function do you use to create a new Turtle object?

Answer:

```
turtle.Turtle()
```

Question 7: What shape does the following code produce?

Answer: Square

Question 8: What shape does the following code produce?

Answer: The turtle will not draw anything

Question 9: Which of the following code sets the screen's color to red?

Answer:

```
screen.bgcolor("red")
```

Question 10: True or False : You can only have one Turtle on the screen.

Answer: False

Question 11: Write a program that makes a Turtle and a Screen. Make the screen one color and the turtle another color. Give the turtle a shape and a color and put its pen down. Have the turtle draw the letter L by moving and turning. Note: The turtle initially starts facing right.

Answer:

```
import turtle

t = turtle.Turtle()
screen = turtle.Screen()

t.color("red")
screen.bgcolor("blue")

t.shape("classic")
t.pendown()

t.right(90)
t.forward(100)
t.left(90)
t.forward(50)
```

● Unit 7 Project Solutions

Make a Snake Game: Sample Solution

```
import turtle, random
```



```

screen = turtle.Screen()
screen.bgcolor("lightgreen")

sprite = turtle.Turtle()
sprite.penup()
sprite.speed(0)
sprite.shape("square")
sprite.goto(-1000,1000)

snake = []
dir = "Right"
food = None

def u():
    global dir
    if(not dir == "Down"):
        dir = "Up"
def d():
    global dir
    if(not dir == "Up"):
        dir = "Down"
def l():
    global dir
    if(not dir == "Right"):
        dir = "Left"
def r():
    global dir
    if(not dir == "Left"):
        dir = "Right"

def createBody(x,y):
    body = sprite.clone()
    body.goto(x,y)
    snake.append(body)

def move():
    last = snake[len(snake)-1]
    first = snake[0]
    x = first.xcor()
    y = first.ycor()
    size = 22
    if(dir == "Right"):
        last.goto((x + (size)),y)
    elif(dir == "Left"):
        last.goto((x - (size)),y)
    elif(dir == "Up"):
        last.goto(x, y + (size))

```

```

else:
    last.goto(x, y - (size))
snake.insert(0, last)
snake.pop()

def createFood():
    global food
    food = sprite.clone()
    food.color("red")
    randX = random.randint(-8, 8) * 22
    randY = random.randint(-8, 8) * 22
    food.goto(randX, randY)

running = False
def update():
    if running:
        move()
        first = snake[0]
        x = first.xcor()
        y = first.ycor()
        global food
        if(x == food.xcor() and y == food.ycor()):
            createBody(first.xcor(), first.ycor())
            food.hideturtle()
            createFood()
        screen.ontimer(update, 350)

def startGame():
    global running
    running = True
    createBody(0, 0)
    createFood()
    update()

screen.onkey(u, "Up")
screen.onkey(d, "Down")
screen.onkey(l, "Left")
screen.onkey(r, "Right")
screen.listen()

startGame()

```

After the Exercise

Question 1: Describe the purpose of this program. (hint: what game does your program play?)

Answer: The purpose of this program is to play the game Snake.

Question 2: Describe the data abstraction that is used in this program. What was it called, what kind of data abstraction is it, and what does it store? (hint: it's called snake)

Answer: The data abstraction that is used in this program is called snake. This is a list that stores turtles which represent the different body parts of the snake.

Question 3: What function in your program has parameters? Why does it have parameters? Give two examples of different calls to your function and the resulting effect of those calls.

Answer: The function `create_body` has two parameters. It has parameters so that in the program a body piece can be created at any given (x, y) coordinate. One example of a call to this function would be `create_body(0, 0)`. Another example of a call to this function would be `create_body(0, 20)`.

Question 4: Which function in this game has a global statement? Why is this statement used in this function? How would the game behavior change if this statement was removed from the function?

Answer: The `up`, `down`, `left`, and `right` functions all have the statement `global dir`. This statement allows changes made to the value of the variable `dir` to be maintained after the function ends. If these statements were taken out of these functions, the changes made would not persist, and in effect the snake would not be able to change direction.

Question 5: In the Snake move function, what do these two lines of code do? In other words, what behavior do they add to the game?

Answer: These two lines of code give the impression that the snake is moving. The last body part of the snake, which is the last thing in the snake list, is inserted at the front of the list and then popped out of the list.

Challenge Tasks

Sample solution:

```
import turtle, random

screen = turtle.Screen()
screen.bgcolor("lightgreen")

sprite = turtle.Turtle()
sprite.penup()
sprite.speed(0)
sprite.shape("square")
sprite.goto(-1000,1000)

snake = []
dir = "Right"
food = None
```

```

def u():
    global dir
    if(not dir == "Down"):
        dir = "Up"
def d():
    global dir
    if(not dir == "Up"):
        dir = "Down"
def l():
    global dir
    if(not dir == "Right"):
        dir = "Left"
def r():
    global dir
    if(not dir == "Left"):
        dir = "Right"

def pause():
    global running
    running = not running

def createBody(x,y):
    body = sprite.clone()
    body.goto(x,y)
    snake.append(body)

def move():
    last = snake[len(snake)-1]
    first = snake[0]
    x = first.xcor()
    y = first.ycor()
    size = 22
    if(dir == "Right"):
        last.goto((x + (size)),y)
    elif(dir == "Left"):
        last.goto((x - (size)),y)
    elif(dir == "Up"):
        last.goto(x, y + (size))
    else:
        last.goto(x, y - (size))
    snake.insert(0,last)
    snake.pop()

def createFood():
    global food
    food = sprite.clone()
    food.color(random.choice(["red", "blue", "yellow"]))
    randX = random.randint(-8,8) * 22

```

```

    randY = random.randint(-8,8) * 22
    food.goto(randX,randY)

running = False
def update():
    if running:
        move()
        first = snake[0]
        x = first.xcor()
        y = first.ycor()
        global food
        if(x == food.xcor() and y == food.ycor()):
            createBody(first.xcor(),first.ycor())
            food.hideturtle()
            createFood()
    screen.ontimer(update, 350)

def startGame():
    global running
    running = True
    createBody(0,0)
    createBody(-22, 0)
    createFood()
    update()

screen.onkey(u, "Up")
screen.onkey(d, "Down")
screen.onkey(l, "Left")
screen.onkey(r, "Right")
screen.onkey(pause, "p")
screen.listen()

startGame()

```

- Follow Up and Challenges Solutions

Sample solution:

```

import turtle, random

START_AMOUNT = 1
GOAL_AMOUNT = 5

screen = turtle.Screen()
screen.bgcolor("lightgreen")

sprite = turtle.Turtle()
sprite.penup()

```

```

sprite.speed(0)
sprite.shape("square")
sprite.goto(-1000,1000)

snake = []
dir = "Right"
foodList = []

def u():
    global dir
    if(not dir == "Down"):
        dir = "Up"
def d():
    global dir
    if(not dir == "Up"):
        dir = "Down"
def l():
    global dir
    if(not dir == "Right"):
        dir = "Left"
def r():
    global dir
    if(not dir == "Left"):
        dir = "Right"
def pause():
    global paused
    paused = not paused

def createBody(x,y):
    body = sprite.clone()
    body.goto(x,y)
    snake.append(body)

def move():
    if not paused:
        last = snake[len(snake)-1]
        first = snake[0]
        x = first.xcor()
        y = first.ycor()
        size = 22
        if(dir == "Right"):
            last.goto((x + (size)),y)
        elif(dir == "Left"):
            last.goto((x - (size)),y)
        elif(dir == "Up"):
            last.goto(x, y + (size))
        else:
            last.goto(x, y - (size))

```

```

        snake.insert(0,last)
        snake.pop()

def createFood(amount):
    global foodList, foodCount, running
    global foodCount
    global running
    if foodCount == GOAL_AMOUNT:
        running = False
        screen.clear()
        screen.bgcolor("blue")
    else:
        foodList = []
        for i in range(amount):
            food = sprite.clone()
            food.color("red")
            randX = random.randint(-8,8) * 22
            randY = random.randint(-8,8) * 22
            food.goto(randX,randY)
            foodList.append(food)

running = False
def update():
    if running:
        move()
        first = snake[0]
        x = first.xcor()
        y = first.ycor()
        global foodList
        for food in foodList:
            if(x == food.xcor() and y == food.ycor()):
                createBody(first.xcor(),first.ycor())
                food.hideturtle()
                foodList.remove(food)
                if len(foodList) == 0:
                    global foodCount
                    foodCount = foodCount + 1
                    createFood(foodCount)
                break
        screen.ontimer(update, 350)

def startGame():
    global running, foodCount
    foodCount = START_AMOUNT
    running = True
    global paused
    paused = False
    createBody(0,0)

```

```

    createFood (START_AMOUNT)
    update ()

screen.onkey (u, "Up")
screen.onkey (d, "Down")
screen.onkey (l, "Left")
screen.onkey (r, "Right")
screen.onkey (pause, "p")
screen.listen ()

startGame ()

```

After the Exercise

Question 1: In your `create_food()` function, what is the parameter?

Answer: The parameter is the amount of food that will be placed into the food list.

Question 2: In your `create_food()` function, what is the selection statement?

Answer:

```

if foodCount == GOAL_AMOUNT:

```

Question 3: In your `create_food()` function, what is the iteration statement?

Answer:

```

for i in range (amount):

```

Question 4: Copy and paste the code segment that shows how data was stored in this list.

Answer:

```

snake.append (body)

```

Question 5: Copy and paste the code segment that shows the data in this list being used, such as creating new data from the existing data or accessing multiple elements in the list, as part of fulfilling the program's purpose.

Answer:

```

def move():
    if not paused:
        last = snake[len(snake)-1]
        first = snake[0]
        x = first.xcor()
        y = first.ycor()
        size = 22
        if (dir == "Right"):
            last.goto((x + (size)), y)

```



```

elif(dir == "Left"):
    last.goto((x - (size)),y)
elif(dir == "Up"):
    last.goto(x, y + (size))
else:
    last.goto(x, y - (size))
snake.insert(0,last)
snake.pop()

```

Question 6: Describe what the data contained in this list represents in your program.

Answer: The data in the snake list represents the body of the snake.

Question 7: Explain how the selected list manages complexity in your program code by explaining why your program code could not be written, or how it would be written differently, if you did not use this list.

Answer: This list keeps track of all of the different body segments of the snake and allows us to move them easily. Without this list it would be hard to keep track of the snake.

Question 8: Copy and paste the code segment that shows how data was stored in this list.

Answer:

```

for i in range(amount):
    food = sprite.clone()
    food.color("red")
    randX = random.randint(-8,8) * 22
    randY = random.randint(-8,8) * 22
    food.goto(randX,randY)
    foodList.append(food)

```

Question 9: Copy and paste the code segment that shows the data in this list being used, such as creating new data from the existing data or accessing multiple elements in the list, as part of fulfilling the program's purpose.

Answer:

```

for food in foodList:
    if(x == food.xcor() and y == food.ycor()):
        createBody(first.xcor(),first.ycor())
        food.hideturtle()
        foodList.remove(food)
    if len(foodList) == 0:
        global foodCount
        foodCount = foodCount + 1
        createFood(foodCount)
    break

```

Question 10: Describe what the data contained in this list represents in your program.

Answer: This list stores all of the food items in the game.

Question 11: Explain how the selected list manages complexity in your program code by explaining why your program code could not be written, or how it would be written differently, if you did not use this list.

Answer: The food list helps us store all of the food currently in the game. Without a list like this it would be much more difficult to keep track of each food item individually.