Introduction to Software Design

P02. Guess the Number, Jokes

Tae Hyun Kim
Spring 2020

Introduction (1/2)

- How Programs Run on Computers

- The “Guess the Number” Game
  - Code Explanation
    - Arguments
    - Blocks
    - Conditions and Booleans
    - `if` statements
  - Code Explanation - Step by step
    - Make Minor Changes
    - What Exactly is Programming?

Introduction (2/2)

- “Jokes”
  - Sample Run
  - Source Code
  - Code Explanation

- Things Covered In This Chapter

How Programs Run on Computers

How Programs Run on Computers

- **Operating System (OS)**
  - Windows, Mac OS, Linux, …
  - A software that manages computer hardware and software resources and provides common services for computer programs.

- **Hardware**
  - Parts of the computer that you **can touch**
  - CPU, GPU, RAM, mainboard, monitor, keyboard and mouse, …

- **Software**
  - Programs like OS, applications, or games that run on the computer.

How Programs Run on Computers

How Programs Run on Computers

- **Machine Code**
  - Very basic instructions
  - Simple enough for computer's main microchip to understand
    - » **CPU** or **Central Processing Unit**
  - Written in ones and zeros.
    - » 10101101 00110000 11000000
  - These instructions aren't quite easy for humans to work with.

How Programs Run on Computers

How Programs Run on Computers

- **Assembly language**
  - Ex) MOV, JMP, PUSH, or XOR
  - makes reading and writing the instructions easier
  - but still **difficult to deal with**

- This is where **high-level programming languages** come in.

How Programs Run on Computers

How Programs Run on Computers

- **High-level languages**
  - Ex) Python, Java, C++, Pascal, Perl, Basic, and many others.
  - closer to human languages and further from machine languages.

- **Interpreter**
  - translates high-level languages into machine code.

“Guess the Number”

The “Guess the Number” Game

- **“Guess the Number” Game**
  - **Computer** will generate a **random number between 1 and 20.**
  - Ask you to **guess the number.**
    - You can try only **six times.**
    - If you’re wrong, the computer will tell you if your guess is **high or low.**
  - If you guess the number **within six tries, you win.**

The “Guess the Number” Game

- **Sample Run of “Guess the Number”**

```
Hello! What is your name?
Albert
Well, Albert, I am thinking of a number between 1 and 20.
Take a guess.
10
Your guess is too high.
Take a guess.
2
Your guess is too low.
Take a guess.
4
Good job, Albert! You guessed my number in 3 guesses!
```

Building Blocks

- **The `random.randint()` Function**

```
9. number = random.randint(1, 20)
```

  - **`randint()`** function is provided by the random ***module***.

- **`while` statement**

```
while guessesTaken < 6:
```

- `if` statements

```
if fizzy < 10:
```

if keyword — condition

The “Guess the Number” Game

- **Guess the Number's Source Code**

```
1. # This is a guess the number game.
2. import random
3.
4. guessesTaken = 0
5.
6. print('Hello! What is your name?')
7. myName = input()
8.
9. number = random.randint(1, 20)
10. print('Well, ' + myName + ', I am thinking of a number between
    1 and 20.')
11.
12. while guessesTaken < 6:
13.     print('Take a guess.')  # There are four spaces in front of
                                 print.
14.     guess = input()
15.     guess = int(guess)
16.
17.     guessesTaken = guessesTaken + 1
```

The “Guess the Number” Game

- **Guess the Number's Source Code**

```
18.
19.     if guess < number:
20.         print('Your guess is too low.')
21.
22.     if guess > number:
23.         print('Your guess is too high.')
24.
25.     if guess == number:
26.         break
27.
28. if guess == number:
29.     guessesTaken = str(guessesTaken)
30.     print('Good job, ' + myName + '! You guessed my number in
' +  guessesTaken + ' guesses!')
31.
32. if guess != number:
33.     number = str(number)
34.     print('Nope. The number I was thinking of was ' + number)
```

Code Explanation

- **Comment**
  - It just tells us what this program does.

```
1. # This is a guess the number game.
```

- **Modules**
  - Other programs that contain other functions we can use.

- **`import` statement**
  - It will **add modules** and **their functions** to our program.
  - It is made up of the **`import` keyword** followed by the **module name.**

```
2. import random
```

Code Explanation

- This creates a **new variable**
  - We will store the integer 0 here.

```
4. guessesTaken = 0
```

- These two lines are something like what we have seen in the **Hello World program**.

```
6. print('Hello! What is your name?')
7. myName = input()
```

Code Explanation

- We can **change the game's code** slightly.

```
9. number = random.randint(1, 20)
10. print('Well, ' + myName + ', I am thinking of a number
    between 1 and 20.')
```

```
9. number = random.randint(1, 100)
10. print('Well, ' + myName + ', I am thinking of a number
    between 1 and 100.')
```

random.randint()

- **The `random.randint()` Function**

```
9. number = random.randint(1, 20)
```

  - The return value is placed in a variable named `number`.
  - **`randint()`** function is provided by the random module.
    - » We precede it with **random.**
    - » It returns a random integer.
      - between the **two integers** we specify. (separated by a comma)
      - Here, it should return an integer between **1 and 20**.
  - random.randint(a, b)
      - Return a random integer N such that a <= N <= b.

random.randint()

Arguments

- The values that are passed to a function, when it is called.

```
input()
random.randint(1, 20)
```

  - The **`input()`** function has **no arguments.**
  - The **`randint()`** function call has **two arguments.**
    - » The arguments are said to be **delimited** by commas.

random.randint()

- Type **`import random`** to import the random module.

```
>>> import random
>>> random.randint(1, 20)
12
>>> random.randint(1, 20)
18
>>> random.randint(1, 20)
3
>>> random.randint(1, 20)
18
>>> random.randint(1, 20)
7
```

```
>>> random.randint(1, 4)
3
>>> random.randint(1, 4)
4
>>> random.randint(1000, 2000)
1294
>>> random.randint(1000, 2000)
1585
```

random.randint()

- Try

```
>>> randint(1, 20)
>>>
```

```
>>> random.randint(100, 100)
>>>
```

```
>>> random.randint(5.0, 10.0)
>>>
```

```
>>> random.randint(5.5, 10.0)
>>>
```

Code Explanation

- **`print` function**

```
print('Well, ' + myName + ', I am thinking of a
number between 1 and 20.')
```

  - The **plus signs** are used to concatenate the three strings.
  - The **commas *inside* the quotes** are part of the strings themselves.

Code Explanation

- **`while` statement**

```
while guessesTaken < 6:
```

  - Is made up of the **`while` keyword**, followed by **an expression**, followed by **a colon(the : sign).**

- **Condition**
  - **The expression** next to the while keyword is called a condition.

Code Explanation

- **Blocks**
  - A block is made up of several lines of code grouped together.

```
while guessesTaken < 6:
    print('Take a guess.')                    ①
    guess = input()
    guess = int(guess)

    guessesTaken = guessesTaken + 1

    if guess < number:
        print('Your guess is too low.')       ②

    if guess > number:
        print('Your guess is too high.')      ③
```

Code Explanation

Blocks

- We can tell where a **block begins and ends** by looking at the line's **indentation.**

```
while guessesTaken < 6:
    print('Take a guess.`)
    guess = input()
    guess = int(guess)

    guessesTaken = guessesTaken + 1

    if guess < number:
        print('Your guess is too low.')

    if guess > number:
        print('Your guess is too high.')
```

Code Explanation

Loop block

- The block after the `while` keyword is called a **loop block.**
  - also called a **while-block.**
  - If the condition is **true**
    - » Program enters the **loop block again.**
  - If the condition is **false**
    - » Program **jumps** down to the line **after the loop block.**

Code Explanation

- **Conditions and Booleans**

```
while guessesTaken < 6:
```

  - The expression that comes after the `while` keyword is called the **condition.**
  - It contains **two values** connected by **an operator**
    - » **Two values**
      : variable `guessesTaken`, integer value 6
    - » **Operator**
      : the < sign, which is called the **"less than"** sign.

Code Explanation

Conditions and Booleans

- **Comparison operators.**

Operator Sign	Operator Name
<	**Less** than
>	**Greater** than
<=	**Less** than or **equal** to
>=	**Greater** than or **equal** to
==	**Equal** to
!=	**Not equal** to

Code Explanation

Conditions and Booleans

- **Boolean type**

```
True
False
```

  - There are two and only **two values.**
  - Must be exactly **`True`** of **`False`**
    - not **`true`** or **`fALSe`**, **`case-sensitive`**

- **Condition**
  - An expression that uses **comparison operators.**
  - Always evaluate to a **Boolean value.**

Code Explanation

- Type in the following conditions.

```
>>> 0 < 6
True
>>> 6 < 0
False
>>> 50 < 10
False
>>> 10 < 11
True
>>> 10 < 10
False
```

Code Explanation

- **Looping with `while` statements**
  - The **`while`** statement indicates the beginning of a **loop.**
  - If the condition evaluates to **True**
    - » the execution **moves inside the while-block.**
  - If the condition evaluates to **False**
    - » the execution **moves all the way past the while-block.**

Code Explanation

- **Looping with `while` statements**

for loop

- `for` Loops
  - The `for` loop is very good at looping over a list of values.
  - begins with the **`for`** keyword, followed by a variable name, the **`in`** keyword, a sequence or a range object, and then a colon.
  - Each time the program execution goes through the loop (on each **iteration** through the loop)
  - Syntax

for index_variable **in** list_variable **:**
    loop_body

for index_variable **in** string_variable **:**
    loop_body

for loop

- `for` Loops
  - For example

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

```
0
1
2
3
4
5
6
7
8
9
```

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

```
1
2
3
4
5
6
7
8
9
```

```
for i in range(10,0,-1):
    print(i)
```

```
10
9
8
7
6
5
4
3
2
1
```

Code Explanation

- **The Player Guesses**
  - The program now asks us for a guess.
  - We store this guess in a variable named **`guess`.**

```
13.     print('Take a guess.`)
14.     guess = input()
```

Code Explanation

- **`int()` Function; type casting**

```
15.     guess = int(guess)
```

  - Converting **Strings to Integers.**
  - The **`input()`** function returns a string of text that a player typed.
    - » But we **want an integer** in the program.
    - » If the player enters 5 as her guess,
      - »It will return not an integer 5, but a **string value '5'**.

Code Explanation

- **Incrementing Variables**

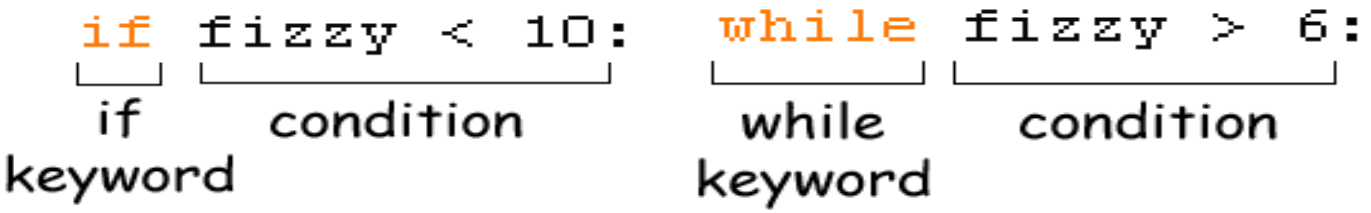
```
17.  guessesTaken = guessesTaken + 1
```

  - At the first time we enter the loop block
    - » `guessesTaken` holds value 0.
    - » Line 17 takes this value and **add 1** to it (0 + 1 is 1).
    - » **The new value 1** is placed in `guessesTaken`.

  - When we **subtract** one from a value
    - » we are **decrementing** the value.

Code Explanation

- **`if` statement**
  - It may be viewed as similar to a `while` statement.
  - But **unlike** the `while`-block,
    - It just continues on down to the next line.
    - In other words, no looping!



Code Explanation

- **`if` statements**
  - **Is the Player's Guess Too Low?**

```
19.     if guess < number:
20.         print('Your guess is too low.')
```

    - If the condition evaluates to **True**
      - » then the code in the **if-block is executed.**
    - If the condition is **False**
      - » then the code in the **if-block is skipped.**

Code Explanation

- **`if` statements**
  - **Is the Player's Guess Too High?**

```
22.     if guess > number:
23.         print('Your guess is too high.')
```

    - If the player's guess is **larger than the random integer**
      - » The program **enters the `if`-block** that follows the `if` statement.
      - » It tells the player that their guess is too big.

Code Explanation

- **`break` Statement**

```
25.     if guess == number:
26.         break
```

  - if the **guess is equal to the random integer**
    - » The program enters line 26, the `if`-block that follows it.

  - It does not bother re-checking the `while` loop's condition.
    - » It just **breaks out immediately.**
    - » Simply the `break` keyword by itself, with no condition or colon.

Code Explanation

- **Check if the Player Won**

```
28. if guess == number:
29.      guessesTaken = str(guessesTaken)
30.      print('Good job, ' + myName + '! You guessed
    my number in ' + guessesTaken + ' guesses!')
```

  - The player **correctly guessed** the computer's number.

  - **Function `str()`**

    » It converts the integer `guessesTaken` into a string value.

Code Explanation

- **Check if the Player Lost**

```
32. if guess != number:
33.      number = str(number)
34.      print('Nope. The number I was thinking of was '
    + number)
```

  - The player **failed to guess** the number within the `guessTaken` trials.
  - **Functiuon `str(number)`**
    - » Inside the if-block, it gets executed only if the condition was **`True`**.
  - Now, the program has reached the **end of the code**, and it **terminates.**

Code Explanation

- **Tracing through the program.**
  - Let's go over the code one more time.
  - To help you understand every piece of it.
  - Think about what values the variables hold and how they change, as we go.

- Note that the following code is written in Python 2, so you have to use in you own version with Python 3
  - print() function instead of print statement
  - input() instead of raw_input()

Code Explanation – step by step

```
# This is a guess the number game.
import random

guessesTaken = 0

print'Hello! What is your name?'
myName = raw_input()

number = random.randint(1, 20)
print'Well, ' + myName + ', I am thinking of a number between 1 and 20.'
```

Code Explanation – step by step

```
# This is a guess the number game.
import random

guessesTaken = 0

print'Hello! What is your name?'
myName = raw_input()

number = random.randint(1, 20)
print'Well, ' + myName + ', I am thinking of a number between 1 and 20.'
```

Code Explanation – step by step

```
# This is a guess the number game.
import random

guessesTaken = 0

print'Hello! What is your name?'
myName = raw_input()

number = random.randint(1, 20)
print'Well, ' + myName + ', I am thinking of a number between 1 and 20.'
```

Code Explanation – step by step

guessesTaken	0

```
# This is a guess the number game.
import random

guessesTaken = 0

print'Hello! What is your name?'
myName = raw_input()

number = random.randint(1, 20)
print'Well, ' + myName + ', I am thinking of a number between 1 and 20.'
```

Code Explanation – step by step

guessesTaken	0

```
# This is a guess the number game.
import random

guessesTaken = 0

print'Hello! What is your name?'
myName = raw_input()

number = random.randint(1, 20)
print'Well, ' + myName + ', I am thinking of a number between 1 and 20.'
```

Code Explanation – step by step

```
# This is a guess the number game.
import random

guessesTaken = 0

print'Hello! What is your name?'
myName = raw_input()

number = random.randint(1, 20)
print'Well, ' + myName + ', I am thinking of a number between 1 and 20.'
```

guessesTaken	0
myName	Bob

Code Explanation – step by step

guessesTaken	0
myName	Bob

```
# This is a guess the number game.
import random

guessesTaken = 0

print'Hello! What is your name?'
myName = raw_input()

number = random.randint(1, 20)
print'Well, ' + myName + ', I am thinking of a number between 1 and 20.'

while guessesTaken < 6:
    print 'Take a guess.'
    guess = raw_input()
    guess = int(guess)

    guessesTaken = guessesTaken + 1
```

Code Explanation – step by step

guessesTaken	0
myName	Bob
number	8

```
# This is a guess the number game.
import random

guessesTaken = 0

print'Hello! What is your name?'
myName = raw_input()

number = random.randint(1, 20)
print'Well, ' + myName + ', I am thinking of a number between 1 and 20.'

while guessesTaken < 6:
    print 'Take a guess.'
    guess = raw_input()
    guess = int(guess)

    guessesTaken = guessesTaken + 1
```

Code Explanation – step by step

guessesTaken	0
myName	Bob
number	8

```
# This is a guess the number game.
import random

guessesTaken = 0

print'Hello! What is your name?'
myName = raw_input()

number = random.randint(1, 20)
print'Well, ' + myName + ', I am thinking of a number between 1 and 20.'

while guessesTaken < 6:
    print 'Take a guess.'
    guess = raw_input()
    guess = int(guess)

    guessesTaken = guessesTaken + 1
```

Code Explanation – step by step

guessesTaken	0
myName	Bob
number	8

```
# This is a guess the number game.
import random

guessesTaken = 0

print'Hello! What is your name?'
myName = raw_input()

number = random.randint(1, 20)
print'Well, ' + myName + ', I am thinking of a number between 1 and 20.'

while guessesTaken < 6:
    print 'Take a guess.'
    guess = raw_input()
    guess = int(guess)

    guessesTaken = guessesTaken + 1
```

Code Explanation – step by step

```
while guessesTaken < 6:
    print 'Take a guess.'
    guess = raw_input()
    guess = int(guess)

    guessesTaken = guessesTaken + 1

    if guess < number:
        print 'Your guess is too low.'

    if guess > number:
        print 'Your guess is too high.'

    if guess == number:
        break
```

guessesTaken	0
myName	Bob
number	8

Code Explanation – step by step

```
while guessesTaken < 6:
    print 'Take a guess.'
    guess = raw_input()
    guess = int(guess)

    guessesTaken = guessesTaken + 1

    if guess < number:
        print 'Your guess is too low.'

    if guess > number:
        print 'Your guess is too high.'

    if guess == number:
        break
```

guessesTaken	0
myName	Bob
number	8
guess	‘12’

Code Explanation – step by step

```
while guessesTaken < 6:
    print 'Take a guess.'
    guess = raw_input()
    guess = int(guess)

    guessesTaken = guessesTaken + 1

    if guess < number:
        print 'Your guess is too low.'

    if guess > number:
        print 'Your guess is too high.'

    if guess == number:
        break
```

guessesTaken	0
myName	Bob
number	8
guess	12

Code Explanation – step by step

```
while guessesTaken < 6:
    print 'Take a guess.'
    guess = raw_input()
    guess = int(guess)

    guessesTaken = guessesTaken + 1

    if guess < number:
        print 'Your guess is too low.'

    if guess > number:
        print 'Your guess is too high.'

    if guess == number:
        break
```

guessesTaken	0
myName	Bob
number	8
guess	12

Code Explanation – step by step

```
while guessesTaken < 6:
    print 'Take a guess.'
    guess = raw_input()
    guess = int(guess)

    guessesTaken = guessesTaken + 1

    if guess < number:
        print 'Your guess is too low.'

    if guess > number:
        print 'Your guess is too high.'

    if guess == number:
        break
```

guessesTaken	1
myName	Bob
number	8
guess	12

Code Explanation – step by step

```
while guessesTaken < 6:
    print 'Take a guess.'
    guess = raw_input()
    guess = int(guess)

    guessesTaken = guessesTaken + 1

    if guess < number:
        print 'Your guess is too low.'

    if guess > number:
        print 'Your guess is too high.'

    if guess == number:
        break
```

guessesTaken	1
myName	Bob
number	8
guess	12

Code Explanation – step by step

```
while guessesTaken < 6:
    print 'Take a guess.'
    guess = raw_input()
    guess = int(guess)

    guessesTaken = guessesTaken + 1

    if guess < number:
        print 'Your guess is too low.'

    if guess > number:
        print 'Your guess is too high.'

    if guess == number:
        break
```

guessesTaken	1
myName	Bob
number	8
guess	12

Code Explanation – step by step

```
while guessesTaken < 6:
    print 'Take a guess.'
    guess = raw_input()
    guess = int(guess)

    guessesTaken = guessesTaken + 1

    if guess < number:
        print 'Your guess is too low.'

    if guess > number:
        print 'Your guess is too high.'

    if guess == number:
        break
```

guessesTaken	1
myName	Bob
number	8
guess	12

Code Explanation – step by step

```
while guessesTaken < 6:
    print 'Take a guess.'
    guess = raw_input()
    guess = int(guess)

    guessesTaken = guessesTaken + 1

    if guess < number:
        print 'Your guess is too low.'

    if guess > number:
        print 'Your guess is too high.'

    if guess == number:
        break
```

guessesTaken	1
myName	Bob
number	8
guess	12

Code Explanation – step by step

```
while guessesTaken < 6:
    print 'Take a guess.'
    guess = raw_input()
    guess = int(guess)

    guessesTaken = guessesTaken + 1

    if guess < number:
        print 'Your guess is too low.'

    if guess > number:
        print 'Your guess is too high.'

    if guess == number:
        break
```

guessesTaken	1
myName	Bob
number	8
guess	12

Code Explanation – step by step

```
while guessesTaken < 6:
    print 'Take a guess.'
    guess = raw_input()
    guess = int(guess)

    guessesTaken = guessesTaken + 1

    if guess < number:
        print 'Your guess is too low.'

    if guess > number:
        print 'Your guess is too high.'

    if guess == number:
        break
```

guessesTaken	1
myName	Bob
number	8
guess	‘6’

Code Explanation – step by step

```
while guessesTaken < 6:
    print 'Take a guess.'
    guess = raw_input()
    guess = int(guess)

    guessesTaken = guessesTaken + 1

    if guess < number:
        print 'Your guess is too low.'

    if guess > number:
        print 'Your guess is too high.'

    if guess == number:
        break
```

guessesTaken	1
myName	Bob
number	8
guess	6

Code Explanation – step by step

```
while guessesTaken < 6:
    print 'Take a guess.'
    guess = raw_input()
    guess = int(guess)

    guessesTaken = guessesTaken + 1

    if guess < number:
        print 'Your guess is too low.'

    if guess > number:
        print 'Your guess is too high.'

    if guess == number:
        break
```

guessesTaken	1
myName	Bob
number	8
guess	6

Code Explanation – step by step

```
while guessesTaken < 6:
    print 'Take a guess.'
    guess = raw_input()
    guess = int(guess)

    guessesTaken = guessesTaken + 1

    if guess < number:
        print 'Your guess is too low.'

    if guess > number:
        print 'Your guess is too high.'

    if guess == number:
        break
```

guessesTaken	2
myName	Bob
number	8
guess	6

Code Explanation – step by step

```
while guessesTaken < 6:
    print 'Take a guess.'
    guess = raw_input()
    guess = int(guess)

    guessesTaken = guessesTaken + 1

    if guess < number:
        print 'Your guess is too low.'

    if guess > number:
        print 'Your guess is too high.'

    if guess == number:
        break
```

guessesTaken	2
myName	Bob
number	8
guess	6

Code Explanation – step by step

```
while guessesTaken < 6:
    print 'Take a guess.'
    guess = raw_input()
    guess = int(guess)

    guessesTaken = guessesTaken + 1

    if guess < number:
        print 'Your guess is too low.'

    if guess > number:
        print 'Your guess is too high.'

    if guess == number:
        break
```

guessesTaken	2
myName	Bob
number	8
guess	6

Code Explanation – step by step

```
while guessesTaken < 6:
    print 'Take a guess.'
    guess = raw_input()
    guess = int(guess)

    guessesTaken = guessesTaken + 1

    if guess < number:
        print 'Your guess is too low.'

    if guess > number:
        print 'Your guess is too high.'

    if guess == number:
        break
```

guessesTaken	2
myName	Bob
number	8
guess	6

Code Explanation – step by step

```
while guessesTaken < 6:
    print 'Take a guess.'
    guess = raw_input()
    guess = int(guess)

    guessesTaken = guessesTaken + 1

    if guess < number:
        print 'Your guess is too low.'

    if guess > number:
        print 'Your guess is too high.'

    if guess == number:
        break
```

guessesTaken	2
myName	Bob
number	8
guess	6

Code Explanation – step by step

```
while guessesTaken < 6:
    print 'Take a guess.'
    guess = raw_input()
    guess = int(guess)

    guessesTaken = guessesTaken + 1

    if guess < number:
        print 'Your guess is too low.'

    if guess > number:
        print 'Your guess is too high.'

    if guess == number:
        break
```

guessesTaken	2
myName	Bob
number	8
guess	6

Code Explanation – step by step

```
while guessesTaken < 6:
    print 'Take a guess.'
    guess = raw_input()
    guess = int(guess)

    guessesTaken = guessesTaken + 1

    if guess < number:
        print 'Your guess is too low.'

    if guess > number:
        print 'Your guess is too high.'

    if guess == number:
        break
```

guessesTaken	2
myName	Bob
number	8
guess	‘8’

Code Explanation – step by step

```
while guessesTaken < 6:
    print 'Take a guess.'
    guess = raw_input()
    guess = int(guess)

    guessesTaken = guessesTaken + 1

    if guess < number:
        print 'Your guess is too low.'

    if guess > number:
        print 'Your guess is too high.'

    if guess == number:
        break
```

guessesTaken	2
myName	Bob
number	8
guess	8

Code Explanation – step by step

```
while guessesTaken < 6:
    print 'Take a guess.'
    guess = raw_input()
    guess = int(guess)

    guessesTaken = guessesTaken + 1

    if guess < number:
        print 'Your guess is too low.'

    if guess > number:
        print 'Your guess is too high.'

    if guess == number:
        break
```

guessesTaken	2
myName	Bob
number	8
guess	8

Code Explanation – step by step

```
while guessesTaken < 6:
    print 'Take a guess.'
    guess = raw_input()
    guess = int(guess)

    guessesTaken = guessesTaken + 1

    if guess < number:
        print 'Your guess is too low.'

    if guess > number:
        print 'Your guess is too high.'

    if guess == number:
        break
```

guessesTaken	3
myName	Bob
number	8
guess	8

Code Explanation – step by step

```
    if guess < number:
        print 'Your guess is too low.'

    if guess > number:
        print 'Your guess is too high.'

    if guess == number:
        break

if guess == number:
    guessesTaken = str(guessesTaken)
    print 'Good job, ' + myName + '! You guessed my number in '
    + guessesTaken + ' guesses!'

if guess != number:
    number = str(number)
    print 'Nope. The number I was thinking of was ' + number
```

guessesTaken	3
myName	Bob
number	8
guess	8

Code Explanation – step by step

```
    if guess < number:
        print 'Your guess is too low.'

    if guess > number:
        print 'Your guess is too high.'

    if guess == number:
        break

if guess == number:
    guessesTaken = str(guessesTaken)
    print 'Good job, ' + myName + '! You guessed my number in '
    + guessesTaken + ' guesses!'

if guess != number:
    number = str(number)
    print 'Nope. The number I was thinking of was ' + number
```

guessesTaken	3
myName	Bob
number	8
guess	8

Code Explanation – step by step

```
    if guess < number:
        print 'Your guess is too low.'

    if guess > number:
        print 'Your guess is too high.'

    if guess == number:
        break

if guess == number:
    guessesTaken = str(guessesTaken)
    print 'Good job, ' + myName + '! You guessed my number in '
    + guessesTaken + ' guesses!'

if guess != number:
    number = str(number)
    print 'Nope. The number I was thinking of was ' + number
```

guessesTaken	3
myName	Bob
number	8
guess	8

Code Explanation – step by step

```
    if guess < number:
        print 'Your guess is too low.'

    if guess > number:
        print 'Your guess is too high.'

    if guess == number:
        break
```

guessesTaken	3
myName	Bob
number	8
guess	8

```
if guess == number:
    guessesTaken = str(guessesTaken)
    print 'Good job, ' + myName + '! You guessed my number in '
    + guessesTaken + ' guesses!'

if guess != number:
    number = str(number)
    print 'Nope. The number I was thinking of was ' + number
```

Code Explanation – step by step

```
    if guess < number:
        print 'Your guess is too low.'

    if guess > number:
        print 'Your guess is too high.'

    if guess == number:
        break

if guess == number:
    guessesTaken = str(guessesTaken)
    print 'Good job, ' + myName + '! You guessed my number in '
    + guessesTaken + ' guesses!'

if guess != number:
    number = str(number)
    print 'Nope. The number I was thinking of was ' + number
```

guessesTaken	3
myName	Bob
number	8
guess	8

Code Explanation – step by step

```
    if guess < number:
        print 'Your guess is too low.'

    if guess > number:
        print 'Your guess is too high.'

    if guess == number:
        break

if guess == number:
    guessesTaken = str(guessesTaken)
    print 'Good job, ' + myName + '! You guessed my number in '
    + guessesTaken + ' guesses!'

if guess != number:
    number = str(number)
    print 'Nope. The number I was thinking of was ' + number
```

guessesTaken	'3'
myName	Bob
number	8
guess	8

Code Explanation – step by step

```
    if guess < number:
        print 'Your guess is too low.'

    if guess > number:
        print 'Your guess is too high.'

    if guess == number:
        break

if guess == number:
    guessesTaken = str(guessesTaken)
    print 'Good job, ' + myName + '! You guessed my number in '
    + guessesTaken + ' guesses!'

if guess != number:
    number = str(number)
    print 'Nope. The number I was thinking of was ' + number
```

guessesTaken	'3'
myName	Bob
number	8
guess	8

Code Explanation – step by step

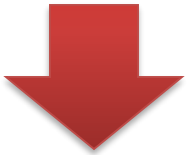
```
    if guess < number:
        print 'Your guess is too low.'

    if guess > number:
        print 'Your guess is too high.'

    if guess == number:
        break

if guess == number:
    guessesTaken = str(guessesTaken)
    print 'Good job, ' + myName + '! You guessed my number in '
    + guessesTaken + ' guesses!'

if guess != number:
    number = str(number)
    print 'Nope. The number I was thinking of was ' + number
```

guessesTaken	3
myName	Bob
number	8
guess	8

Some Changes We Could Make

- Try changing this program

```
number = random.randint(1, 20)
print 'Well, ' + myName + ', I am thinking of a number
       between 1 and 20.'
```



```
number = random.randint(1, 100)
print 'Well, ' + myName + ', I am thinking of a number
       between 1 and 20.'
```

Some Changes We Could Make

- Try changing this program

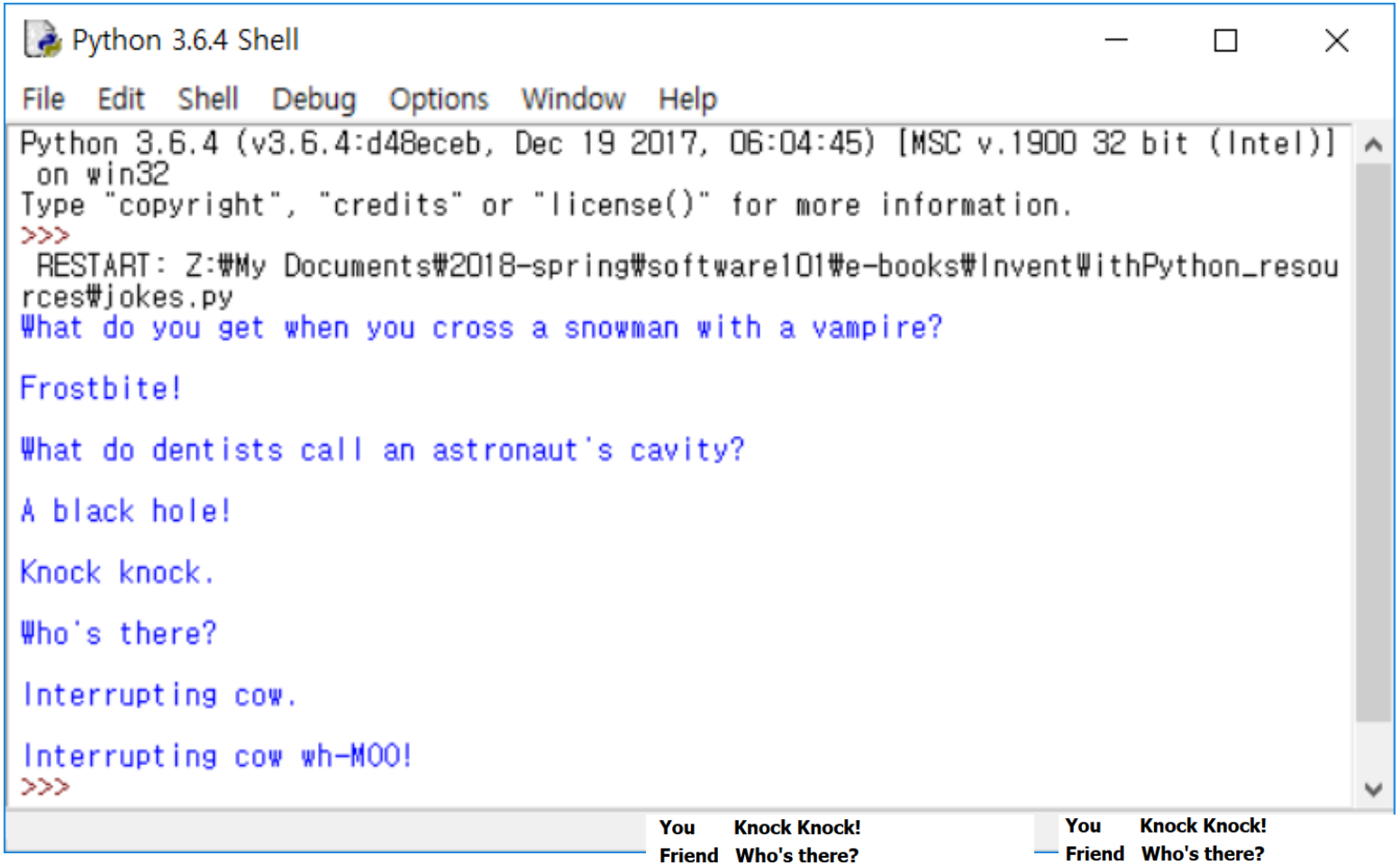
```
while guessesTaken < 6:
```

```
while guessesTaken < 4:
```

“Jokes”

“Jokes”

- **Sample Run**



참고 (강의내용과 관련 없음):

You	Knock Knock!
Friend	Who's there?
You	Harry.
Friend	Harry who?
You	Harry up and it's cold out here!

You	Knock Knock!
Friend	Who's there?
You	Irish.
Friend	Irish who?
You	Irish you a Merry Christmas!

“Jokes”

- **Source Code**

jokes.py - Z:₩My Documents₩2018-spring₩software101₩e-books₩InventW...

File Edit Format Run Options Window Help

```
print('What do you get when you cross a snowman with a vampire?')
input()
print('Frostbite!')
print()
print('What do dentists call an astronaut's cavity?')
input()
print('A black hole!')
print()
print('Knock knock.')
input()
print("Who's there?")
input()
print('Interrupting cow.')
input()
print('Interrupting cow wh', end='')
print('-MOO!')
```

Ln: 17 Col: 0

Code Explanation

- **Three `print` function**

```
print('What do you get when you cross a snowman with a vampire?')
input()
print('Frostbite!')
print()
```

- Read the first line, press Enter, and then read the punch line.
- The user can type in any string and hit Enter
  - because we aren't storing this string in any variable.
- The last call to `print` function has no string.

Code Explanation

Escape Characters

```
print('₩What do dentists call an astronaut₩'s cavity?')
input()
print('A black hole!')
print()
```

- **a slash** right before the single quote (that is, the apostrophe).
  - "\" is a backslash, "/" is a forward slash.
  - The backslash tells us that the letter right after it is an escape character (예외문자, 특수문자).
  - An escape character helps us print out letters.

Code Explanation

- **Some Other Escape Characters**
  - What if you really want to display a backslash?
  - This line of code would not work.

```
>>> print('He flew away in a green\teal helicopter.')
He flew away in a green eal helicopter.
```

Code Explanation

- **Quiz**
  - Instead, try this line

```
>>> print('He flew away in a green##teal helicopter.')
```

Code Explanation

- **Escape Characters**

Escape Character	What Is Actually Printed
\\	Backslash (\)
\'	Single quote (')
\"	Double quote (")
\n	Newline
\t	Tab

Code Explanation

Quotes and Double Quotes

- Strings don't always have to be in between single quotes.
- You can also put them in between **double quotes.**

```
>>> print('Hello world')
Hello world
>>> print("Hello world")
Hello world
```

Code Explanation

- **Quiz**

```
>>> print('Hello world")
```

Code Explanation

- **Quotes and Double Quotes**
  - \' **to have a single quote** in a string surrounded by **single quotes.**
  - \" **to have a double quote** in a string surrounded by **double quotes.**

```
>>> print 'I asked to borrow Abe\'s car for a week. He said, "Sure."'
I asked to borrow Abe's car for a week. He said, "Sure."
>>> print "He said, \"I can't believe you let him borrow your car.\""
He said, "I can't believe you let him borrow your car."
```

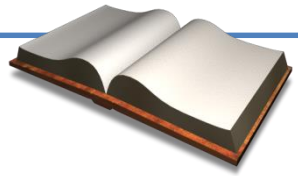
Code Explanation

- **Using Commas**

```
print('Interrupting cow wh', end='')
print('-MOO!')
```

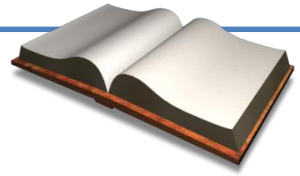
  - The **end** parameter indicates that we don't want to end the line with a newline, but with a blank string.
  - This is why '`-MOO!'` appears next to the previous line, instead of its own new line.

Things Covered In This Chapter(1/3)



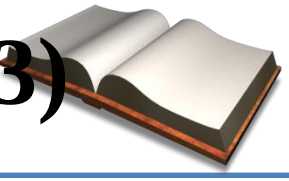
- `import` statements
- Modules
- Arguments
- `while` statements
- Conditions
- Blocks
- Comparison operators

Things Covered In This Chapter(2/3)



- The difference between = and ==.
- `if` statements
- The `break` keyword.
- The `str()` function.
- The `random.randint()` function.

Things Covered In This Chapter(3/3)



- Using `print` function with no parameters to display blank lines.
- Escape characters.
- Using single quotes and double quotes for strings.
- Using the end keyword argument with a blank string.