Introduction to Software Design

P02. Guess the Number

Yoonsang Lee
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?

- More about print()

- What is Programming?
- 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 **higher-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.
  - take care of much of the detail of machine code.

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

“Guess the Number”

The “Guess the Number” Game

- “Guess the Number” Game
  - **Computer** will think of a **random number from 1 to 20.**
  - Ask you to **guess the number.**
    - You only get **six guesses.**
    - but the computer will tell you if your guess is **too high or too 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**

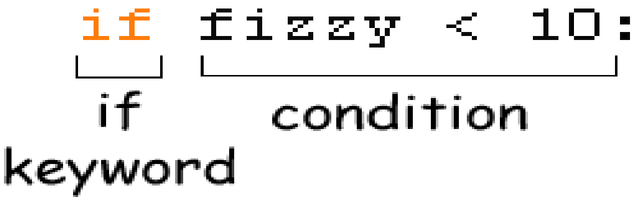
```
number = random.randint(1, 20)
```

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

- **`while` statement**

```
while guessesTaken < 6:
```

- `if` statements



```
if fizzy < 10:
```

The “Guess the Number” Game

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

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

guessesTaken = 0

print('Hello! What is your name?')
myName = 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.')  # There are four spaces in front of print.
    guess = input()
    guess = int(guess)

    guessesTaken = guessesTaken + 1
```

The “Guess the Number” Game

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

```

    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)
```

Code Explanation

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

```
# 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.**

```
import random
```

Code Explanation

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

```
guessesTaken = 0
```

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

```
print('Hello! What is your name?‘)
myName = input()
```

Code Explanation

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

```
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 100.')
```

random.randint()

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

```
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)
>>>
```

Quiz #1

- Go to https://www.slido.com/
- Join #**isd-hyu**
- Click "Polls"

- Submit your answer in the following format:
  - **Student ID: Your answer**
  - **e.g. 2017123456: 4)**

- Note that you must submit all quiz answers in the above format to be checked as "attendance".

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:
```

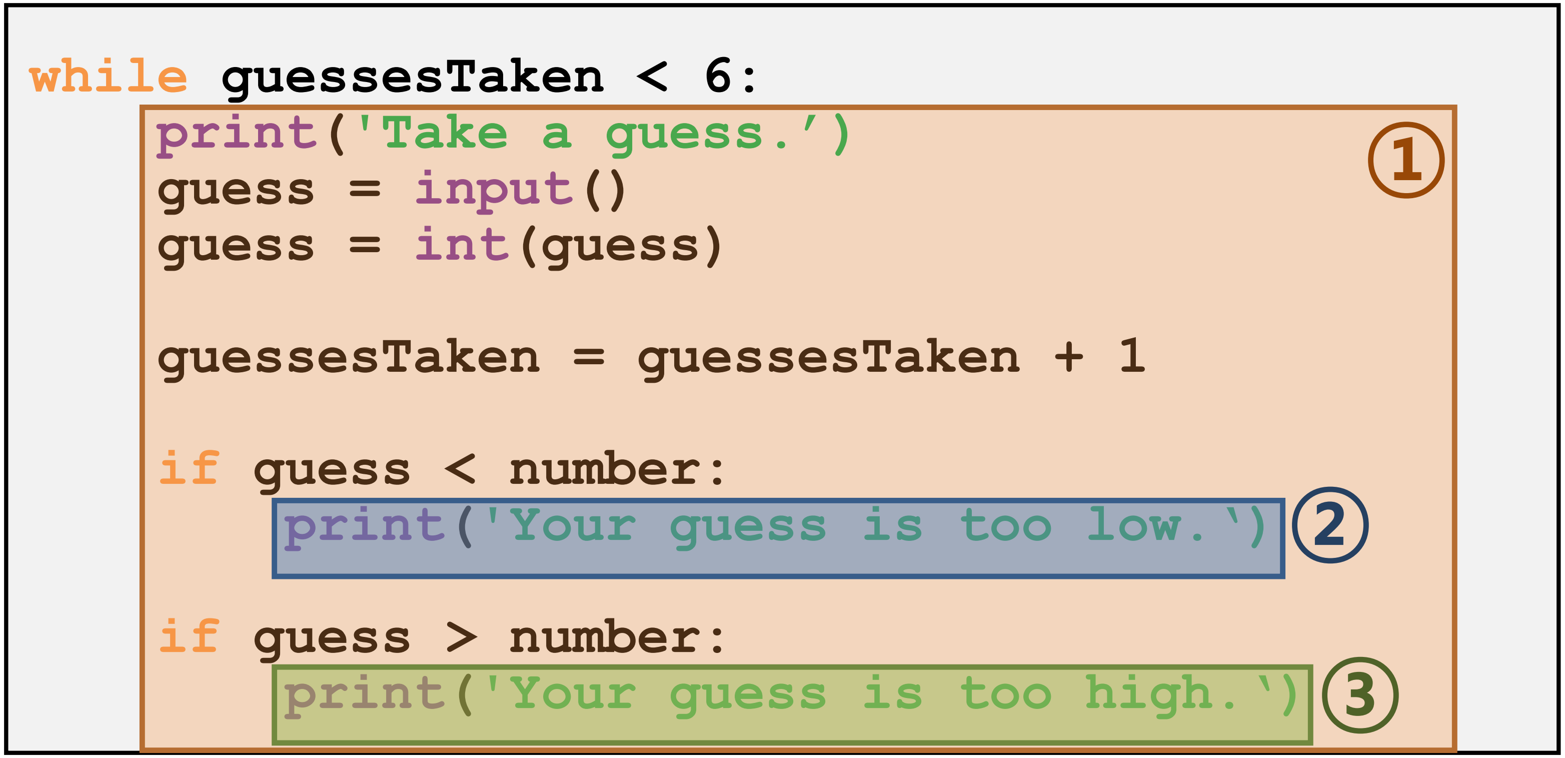
  - Is made up of the **`while` keyword**, followed by **an expression**, followed by **a colon(the : sign).**

- **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.’)
    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

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 Boleans**

  - **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` or `False` (not `true` or `fALSe`).

- **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
```

Quiz #2

- Go to https://www.slido.com/
- Join #**isd-hyu**
- Click “Polls”

- Submit your answer in the following format:
  - **Student ID: Your answer**
  - **e.g. 2017123456: 4)**

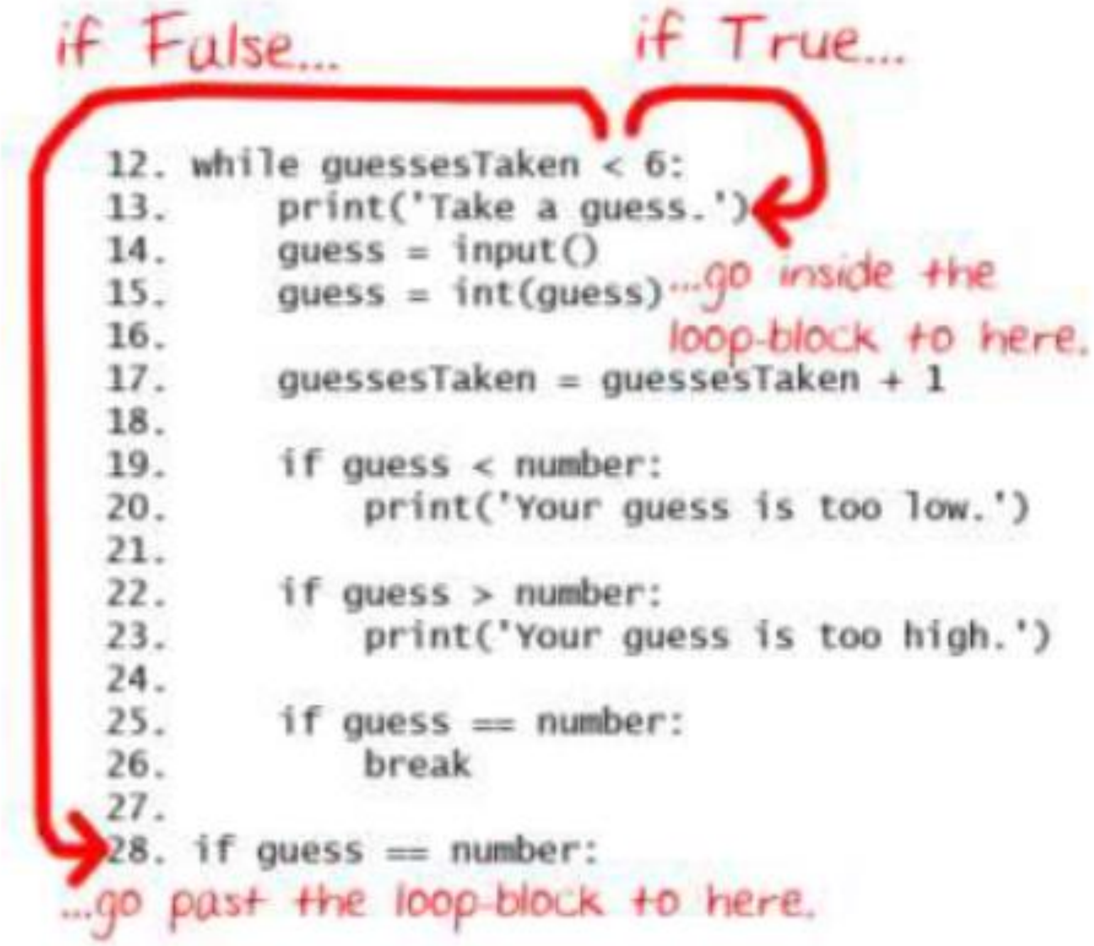
- Note that you must submit all quiz answers in the above format to be checked as “attendance”.

Code Explanation

- **Looping with `while` statements**
  - The **`while`** statement marks 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**



if False...
if True...
...go inside the loop-block to here.
...go past the loop-block to here.

```
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.')

    if guess == number:
        break

if guess == number:
```

for statement

- `for` loop
  - 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.
  - Syntax

```
for  index_variable  in list_variable :
    loop_body

for  index_variable  in string_variable :
    loop_body
```

- `range()` function
  - returns a sequence of integers (as a "range" object)
  - `range(stop)`
  - `range(start, stop[, step])`

for statement

- for loop
  - 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`**.

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

  - 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

- **`int()` Function**

```
    guess = int(guess)
```

  - Converting **Strings to Integers.**

Quiz #3

- Go to https://www.slido.com/
- Join #isd-hyu
- Click “Poll”

- Submit your answer in the following format:
  - **Student ID: Your answer**
  - **e.g. 2017123456: 4)**

- Note that you must submit all quiz answers in the above format to be checked as “attendance”.

Code Explanation

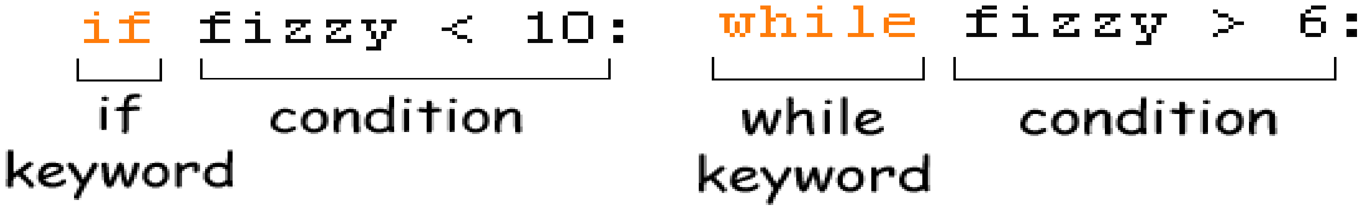
- **Incrementing Variables**

```
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!



```
if fizzy < 10:
while fizzy > 6:
```

Code Explanation

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

```
    if guess < number:
        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?**

```
    if guess > number:
        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**

```
if guess == number:
    guessesTaken = str(guessesTaken)
    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**

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

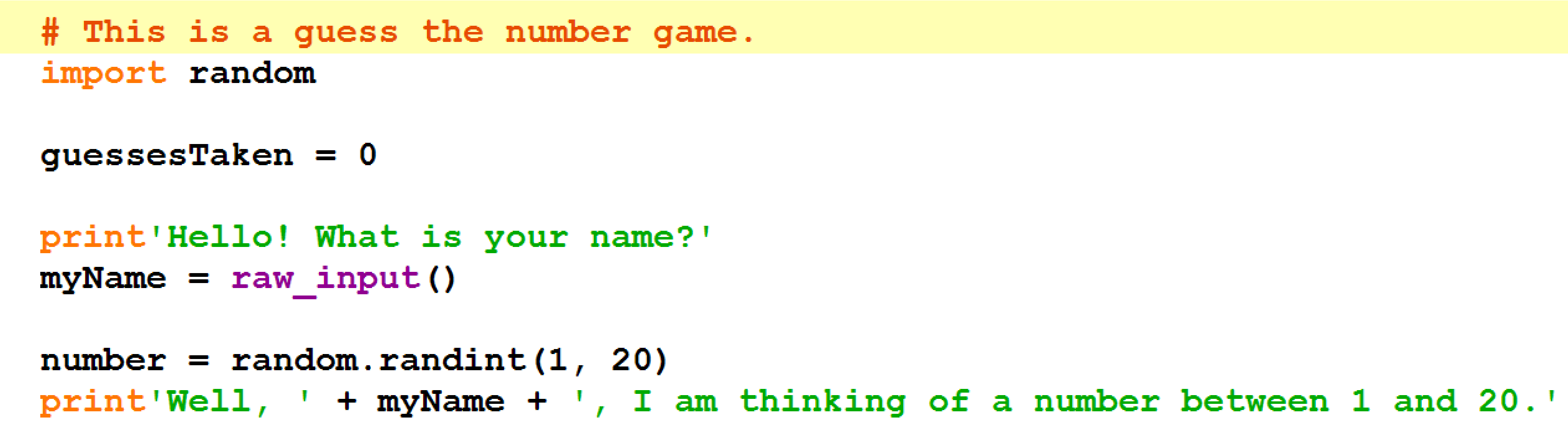
  - The player **failed to guess** the number within the `guessTaken` trials.
  - **Functiuon `str(number)`**
    - » Inside the if-block, it gets executed only if the condition was **`True`**.
  - Now, the program has reached the **end of the code**, and it **terminates.**

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.

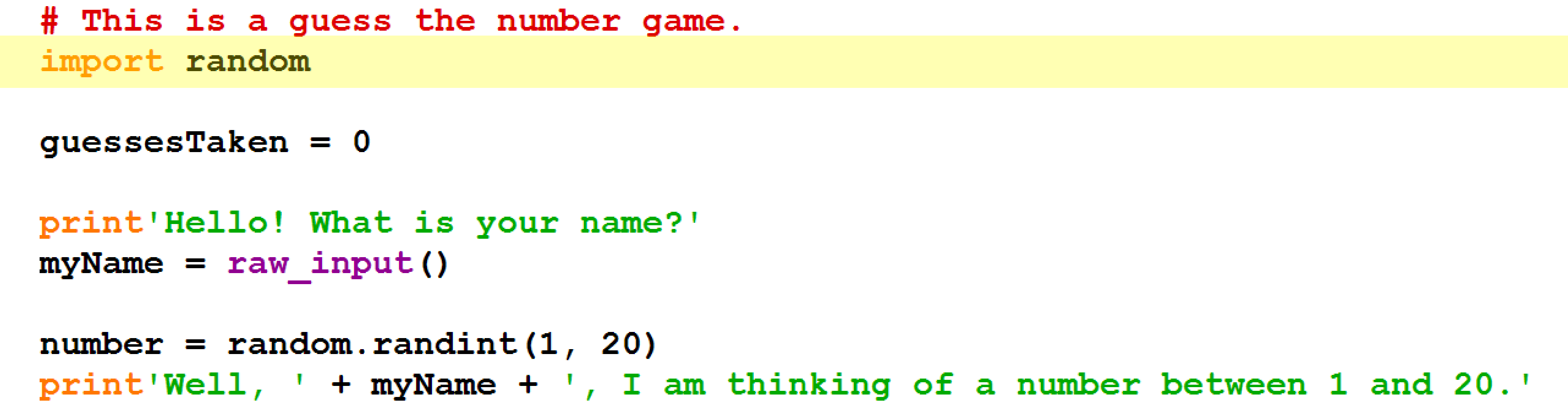
- Note that the following code is written in Python 2, so you have to use
  - print() function instead of print statement
  - input() instead of raw_input()

Code Explanation – step by step



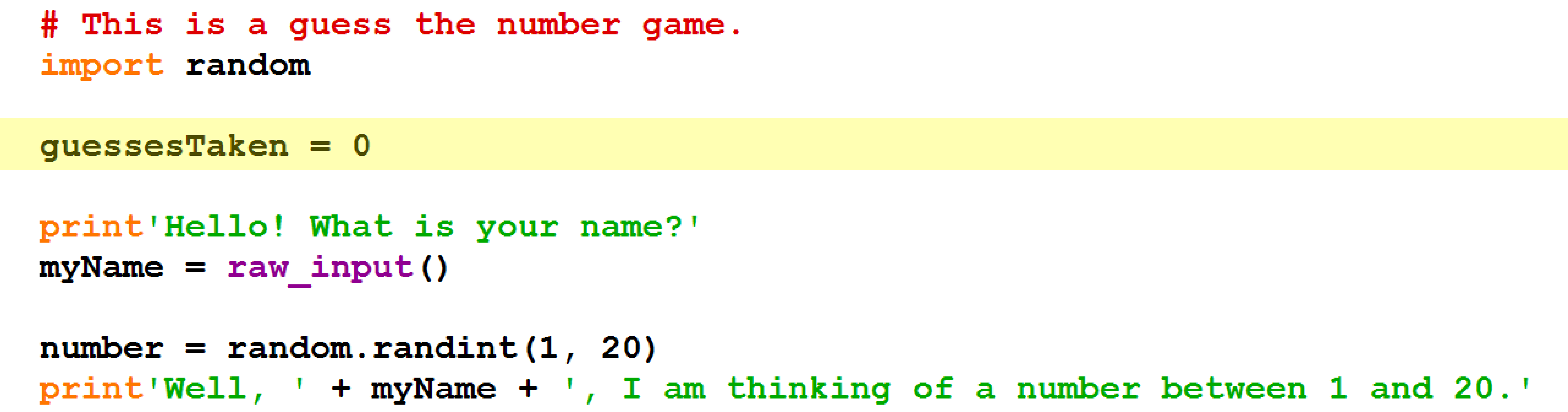
```
# This is a guess the number game.
import random

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

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.'
```

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
```

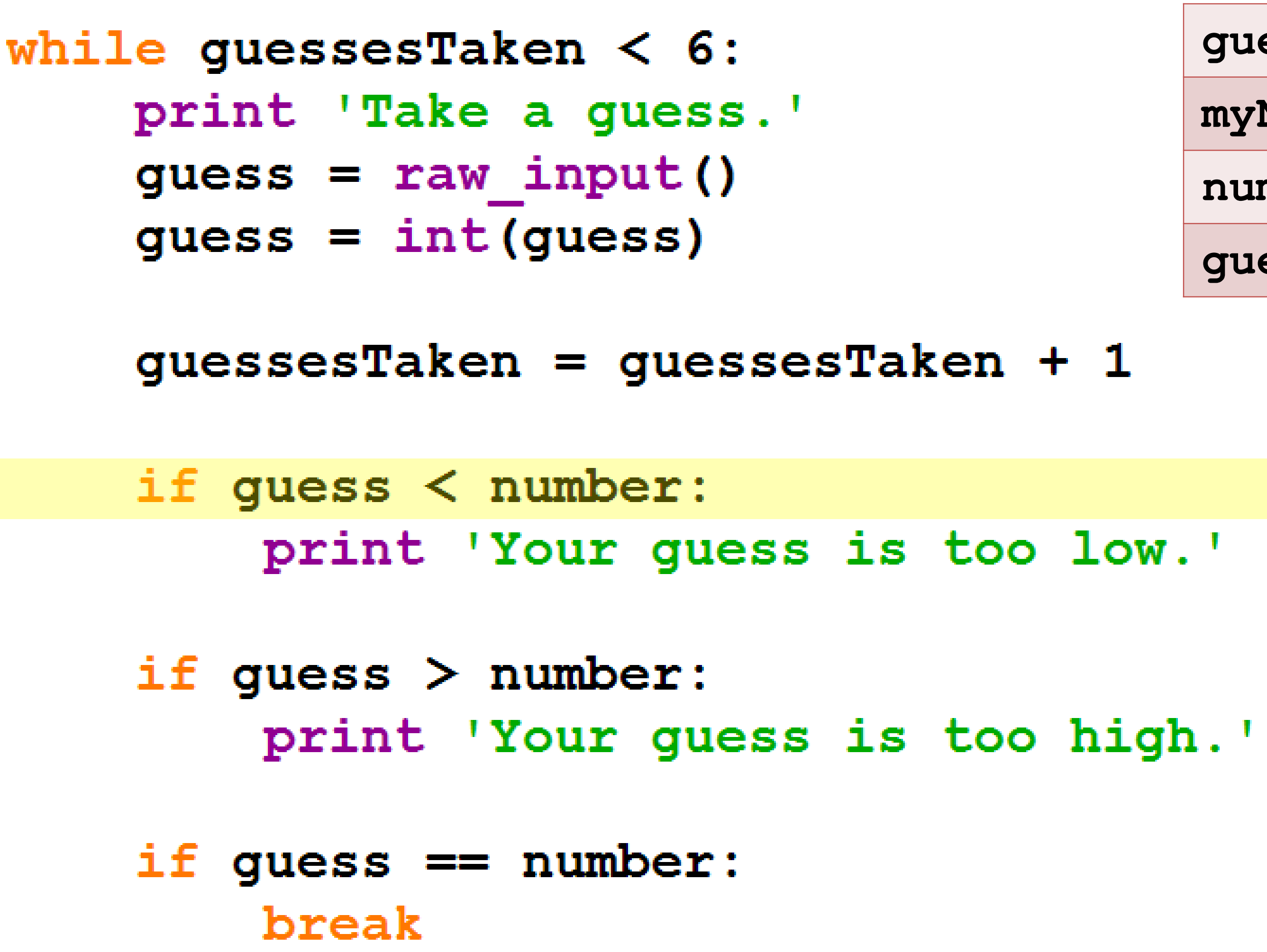
guessesTaken	0
myName	Bob
number	8
guess	12

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
```

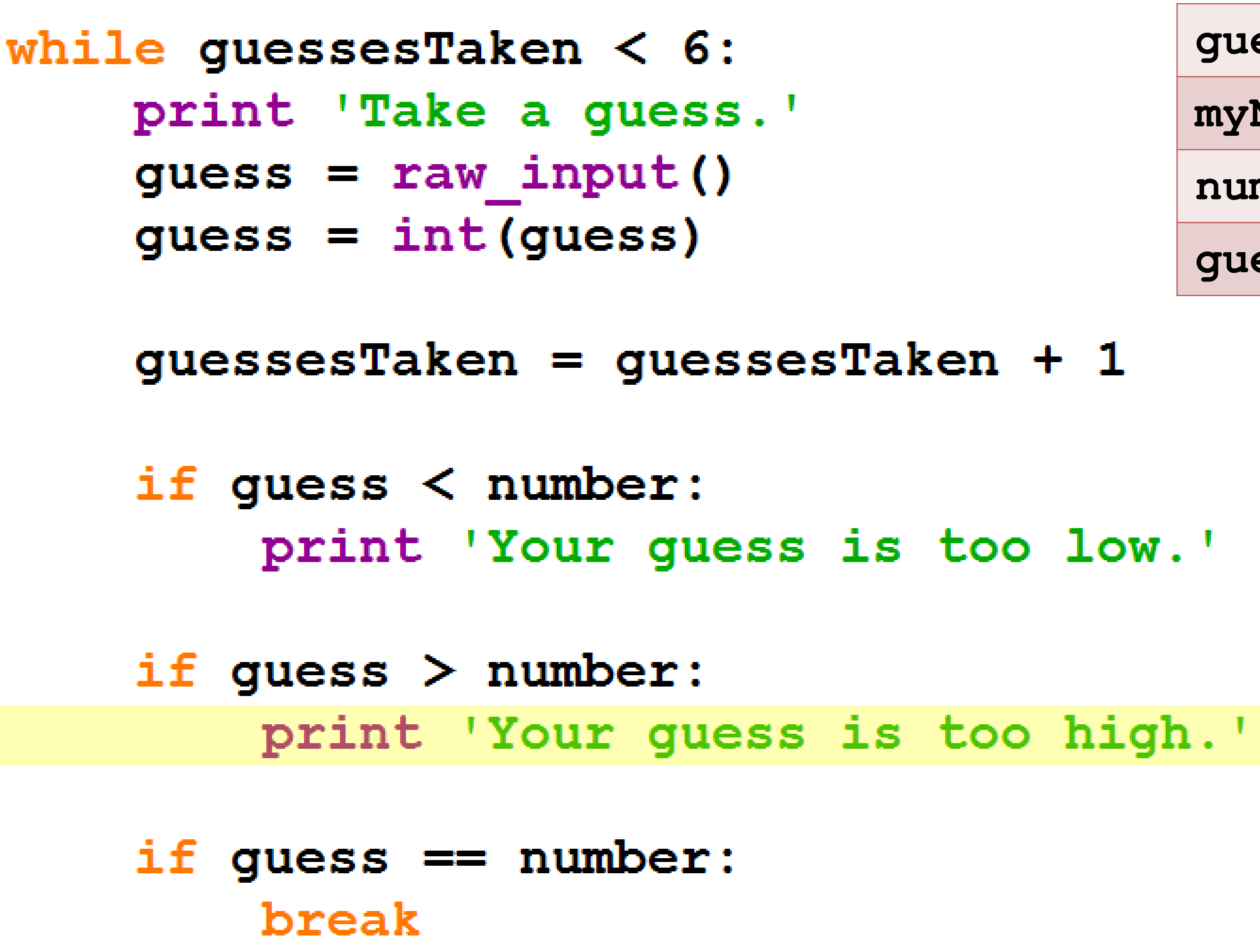
guessesTaken	1
myName	Bob
number	8
guess	12

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

guessesTaken	1
myName	Bob
number	8
guess	12

```
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
```

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

guessesTaken	1
myName	Bob
number	8
guess	6

```
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
```

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
```

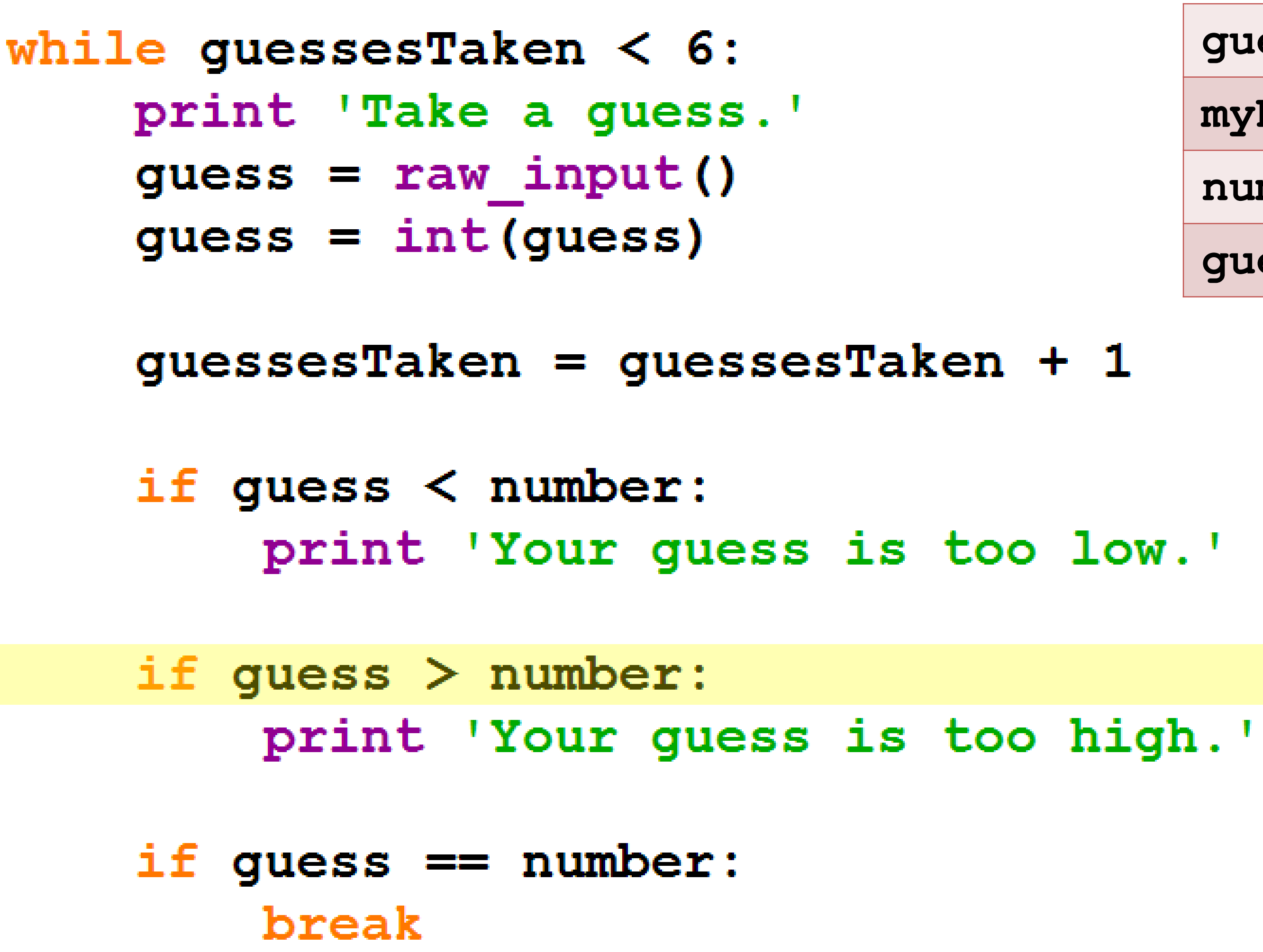
guessesTaken	2
myName	Bob
number	8
guess	6

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

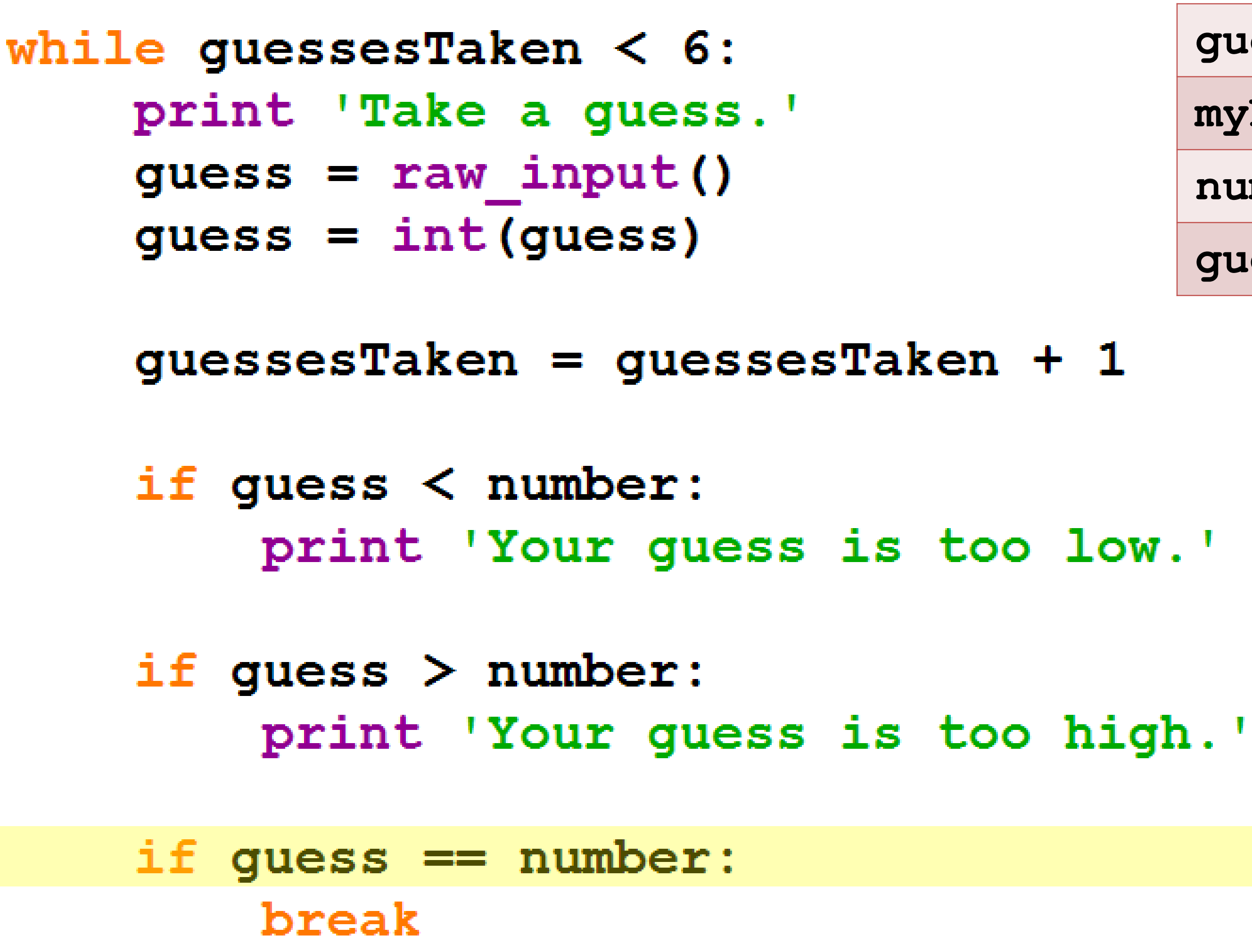
Code Explanation – step by step



```
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
```

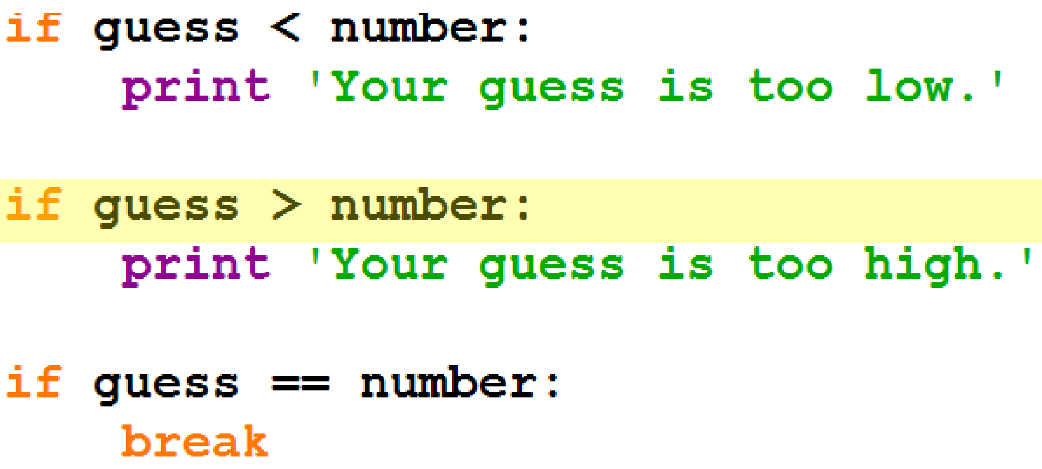
guessesTaken	3
myName	Bob
number	8
guess	8

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
```

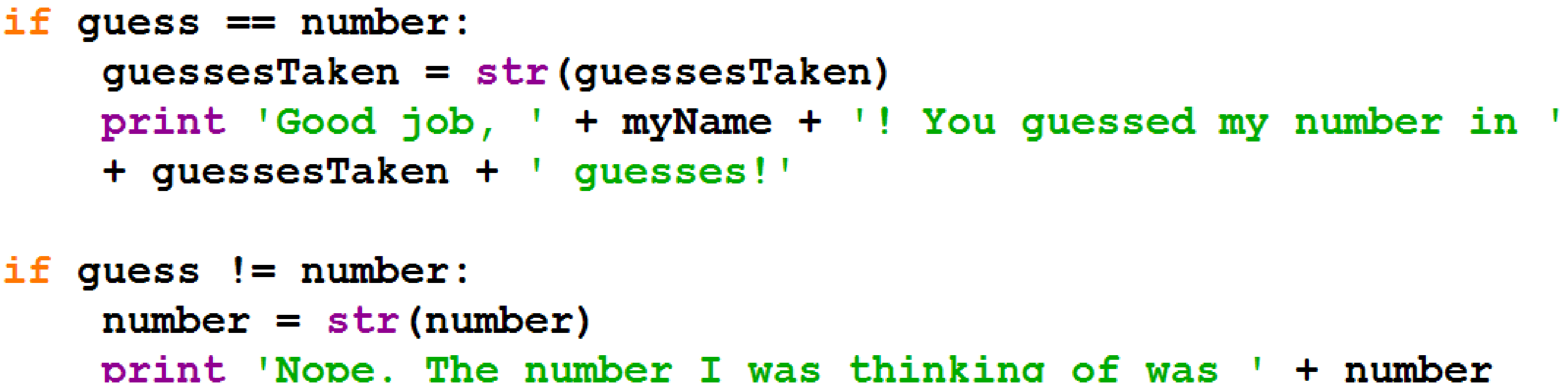
guessesTaken	3
myName	Bob
number	8
guess	8

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
```

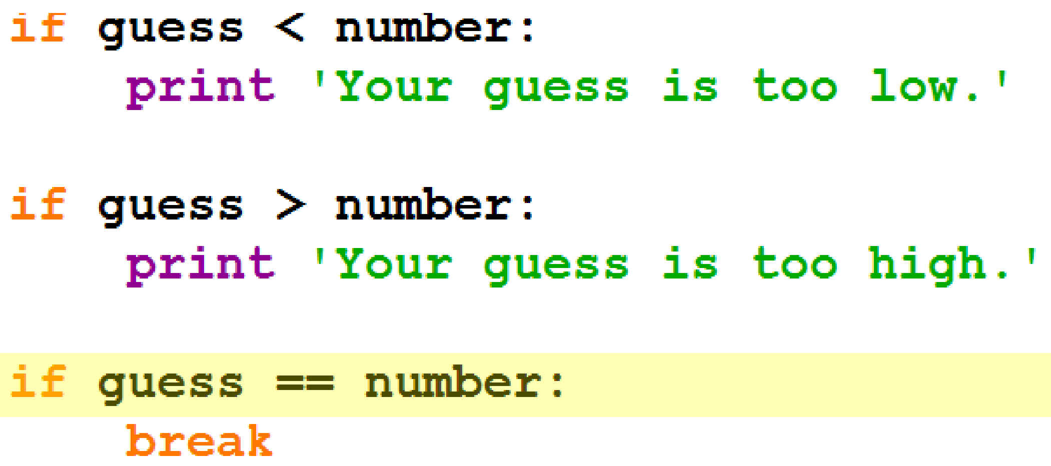
guessesTaken	3
myName	Bob
number	8
guess	8



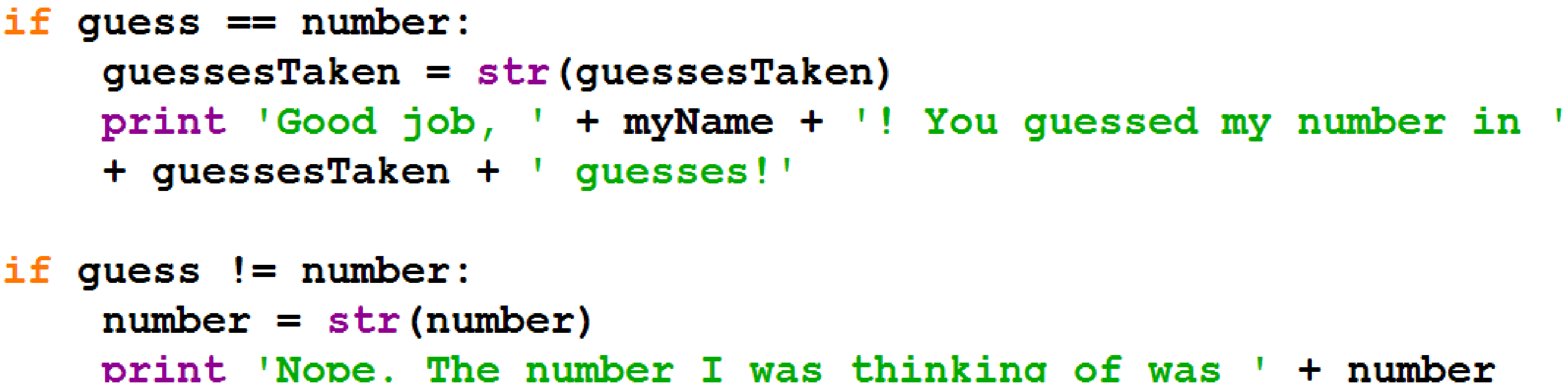
```
if guess == number:
    guessesTaken = str(guessesTaken)
    print 'Good job, ' + myName + '! You guessed my number in '
    + guessesTaken + ' guesses!'

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

Code Explanation – step by step

guessesTaken	3
myName	Bob
number	8
guess	8

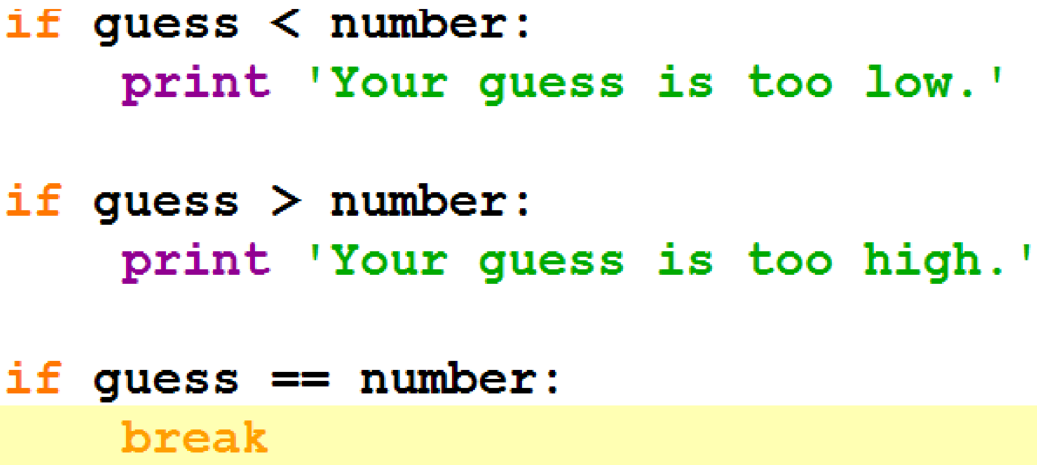


```
if guess < number:
    print 'Your guess is too low.'

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

if guess == number:
    break
```



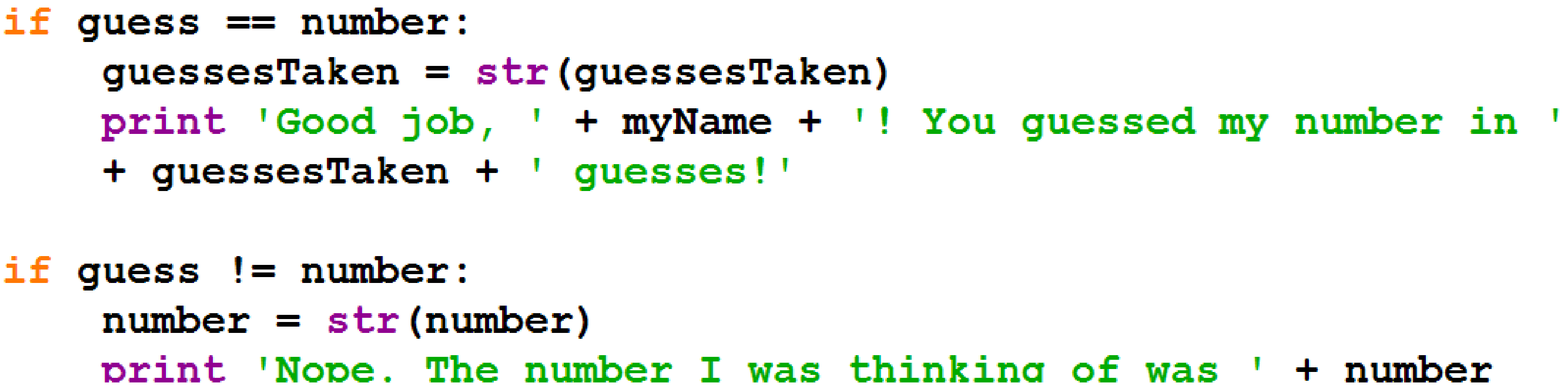
```
if guess == number:
    guessesTaken = str(guessesTaken)
    print 'Good job, ' + myName + '! You guessed my number in '
    + guessesTaken + ' guesses!'

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

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



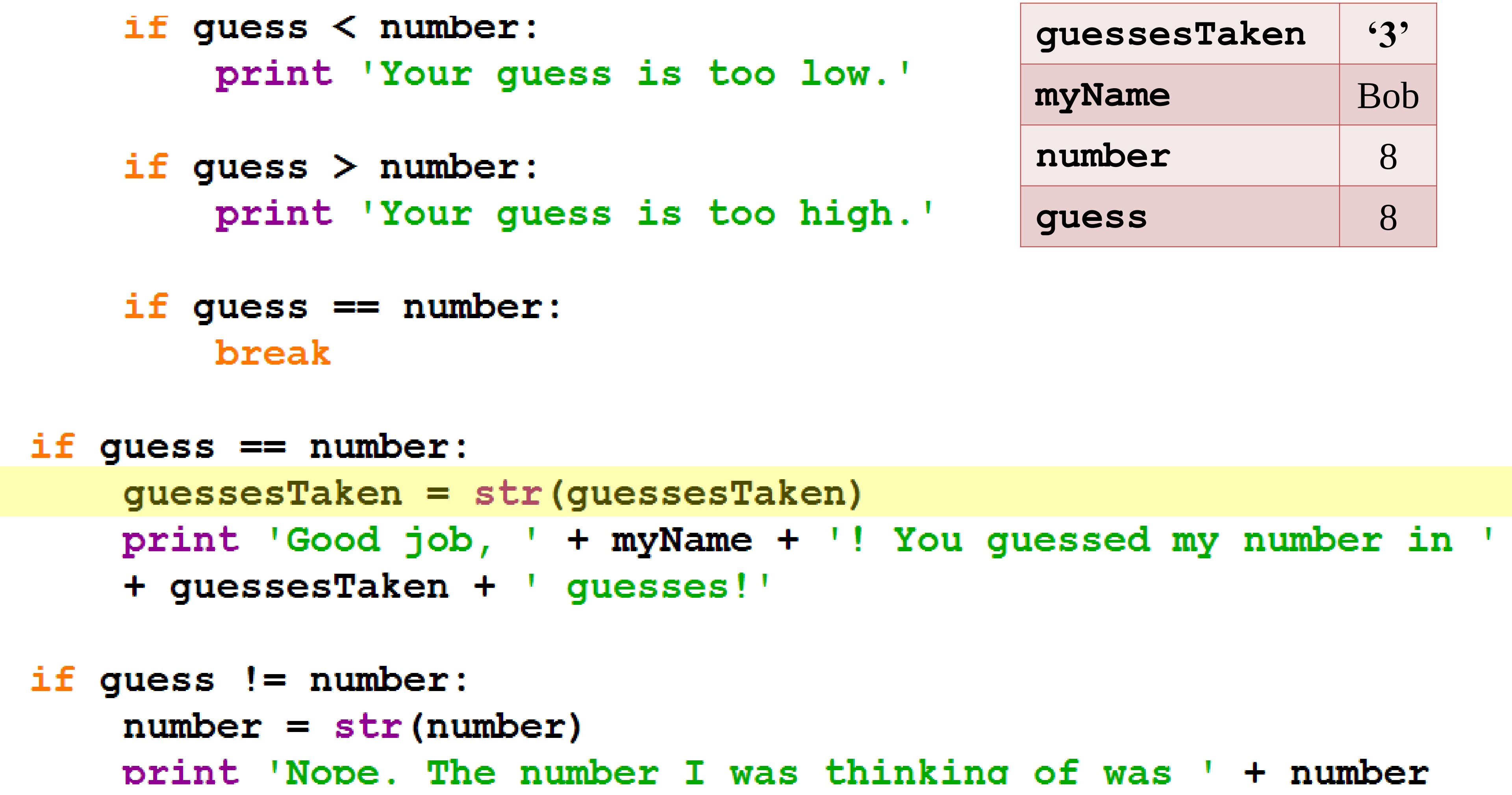
```
if guess == number:
    guessesTaken = str(guessesTaken)
    print 'Good job, ' + myName + '! You guessed my number in '
    + guessesTaken + ' guesses!'

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

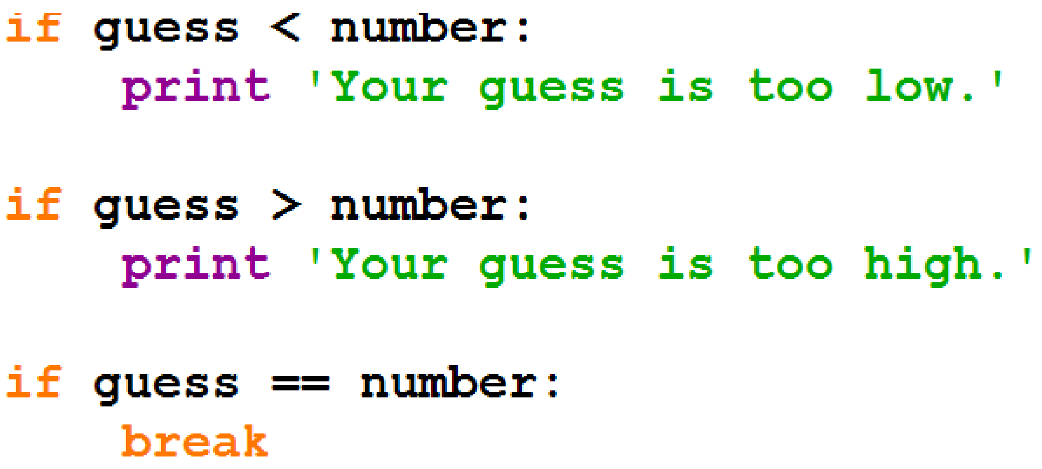
Code Explanation – step by step



```
    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
```

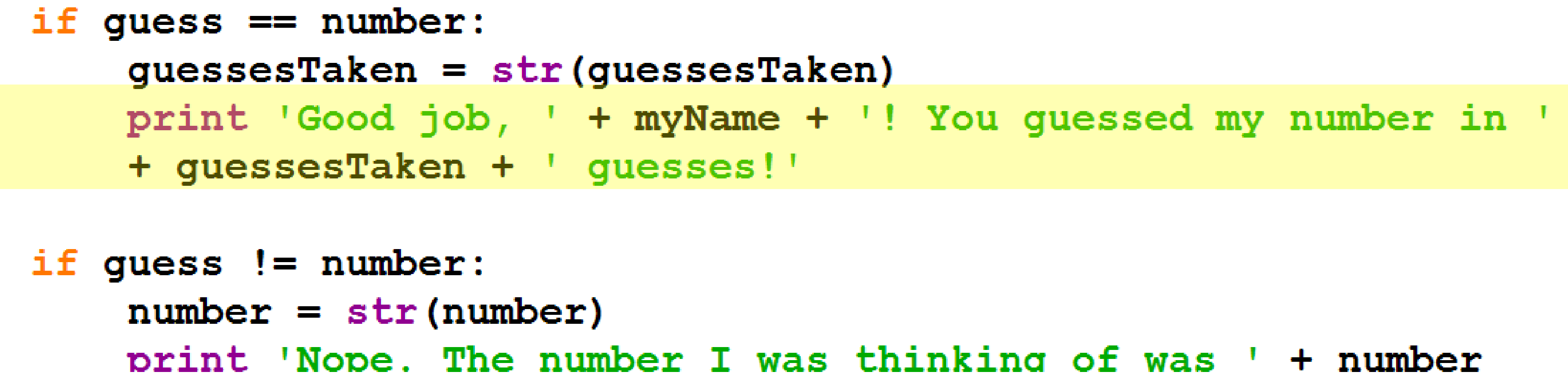
guessesTaken	‘3’
myName	Bob
number	8
guess	8

Code Explanation – step by step



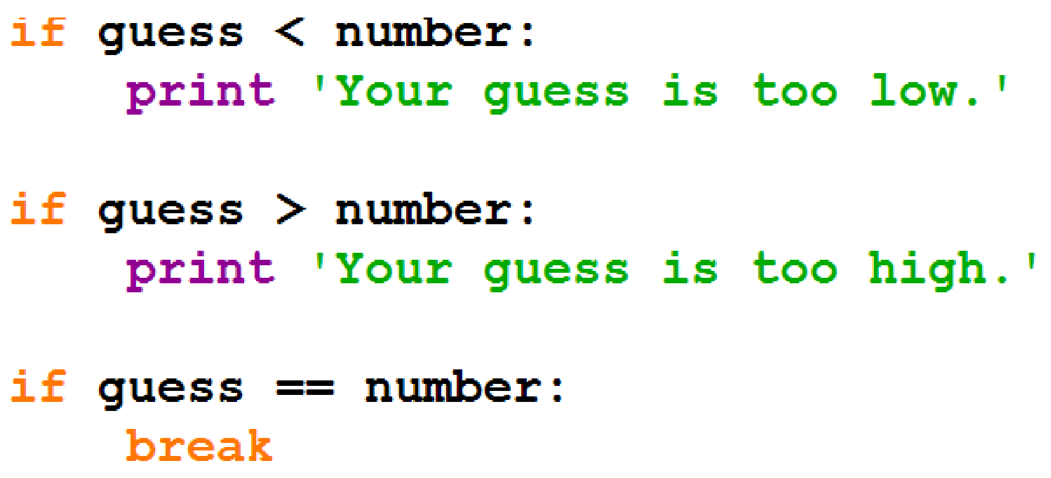
```
if guess < number:
    print 'Your guess is too low.'

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

if guess == number:
    break
```

guessesTaken	‘3’
myName	Bob
number	8
guess	8



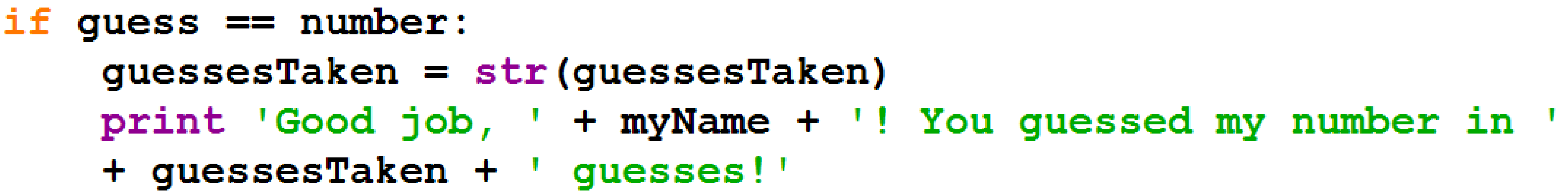
```
if guess == number:
    guessesTaken = str(guessesTaken)
    print 'Good job, ' + myName + '! You guessed my number in '
    + guessesTaken + ' guesses!'

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

Code Explanation – step by step

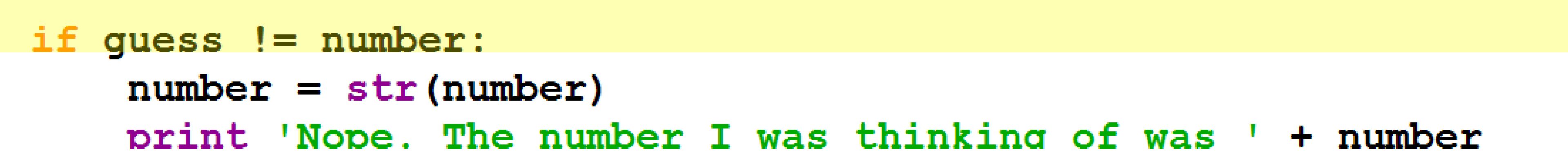


```
    if guess < number:
        print 'Your guess is too low.'

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

    if guess == number:
        break
```

guessesTaken	3
myName	Bob
number	8
guess	8



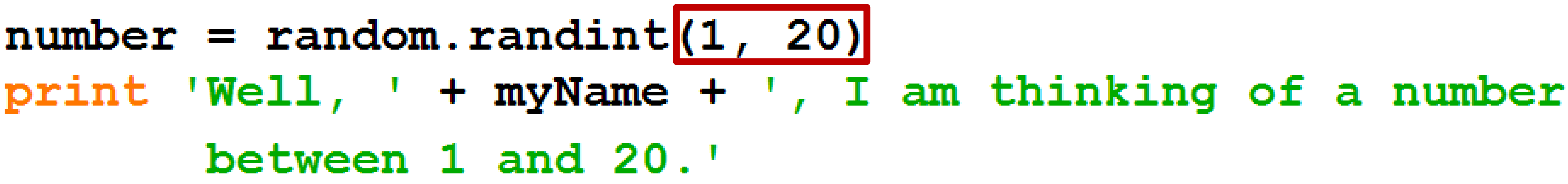
```
if guess == number:
    guessesTaken = str(guessesTaken)
    print 'Good job, ' + myName + '! You guessed my number in '
    + guessesTaken + ' guesses!'
```



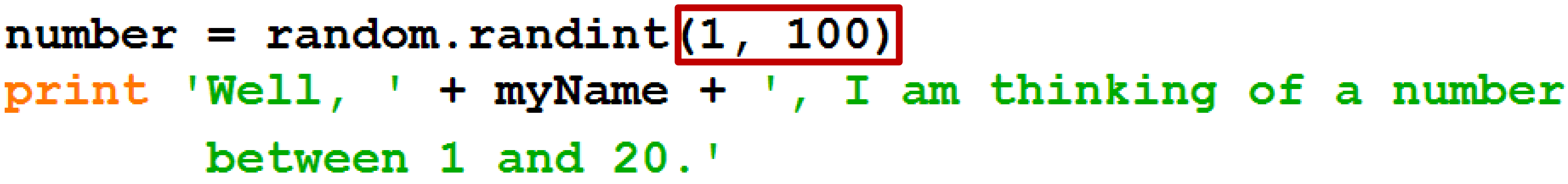
```
if guess != number:
    number = str(number)
    print 'Nope. The number I was thinking of was ' + number
```

Some Changes We Could Make

- Try changing this program



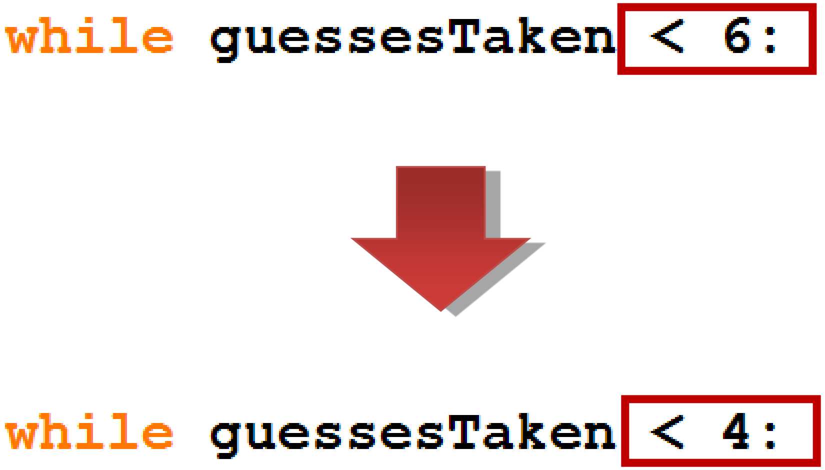
```
number = random.randint(1, 20)
print 'Well, ' + myName + ', I am thinking of a number
       between 1 and 20.'
```



```
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



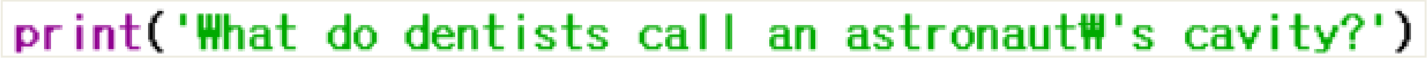
```
while guessesTaken < 6:
```

```
while guessesTaken < 4:
```

More about print()

Code Explanation

Escape Characters



```
print('What do dentists call an astronaut₩'s cavity?')
```

- **a backslash**
  - "/" is a slash, and "\" is a backslash.
  - Note that default Korean fonts in MS Windows render a backslash as a KRW symbol (similar to ₩).
  - The backslash tells us that the letter right after it is an **escape character**.
  - An escape character helps us print out letters.

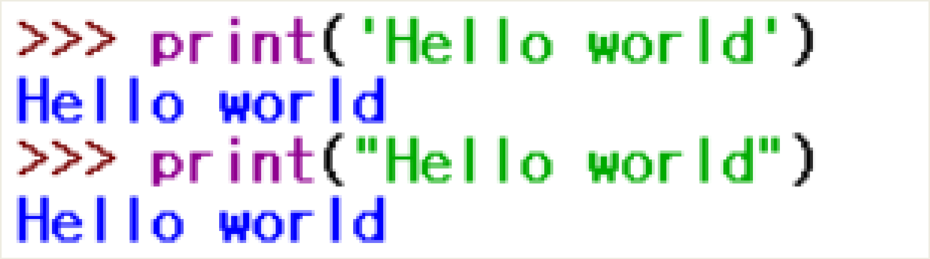
Code Explanation

- **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.**

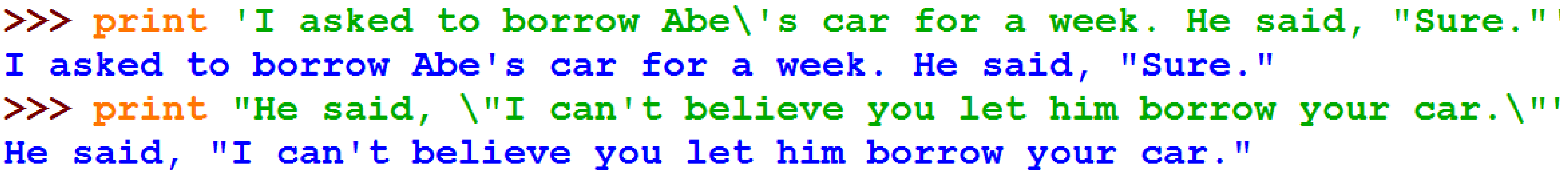


```
>>> print('Hello world')
Hello world
>>> print("Hello world")
Hello world
```

Code Explanation

Quotes and Double Quotes

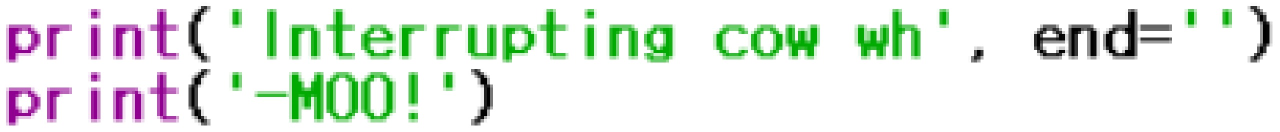
- **\' to have a single quote** in a string surrounded by **single quotes.**
- **\" to have a double quote** in a string surrounded by **double quotes.**



```
>>> 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

end parameter



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

- By default python’s print() function ends with a **newline**.
  - If you print nothing by typing just "`print()`", a newline will be displayed.
- In other words, the default value of the **end** paramemter of print() function is '**\n**'.
- You can specify any other character as the "end" character.
  - `print('test', end=' ')  # ends with a space`
  - `print('test', end='')  #  ends without any character`

What is Programming?

What Exactly is Programming?

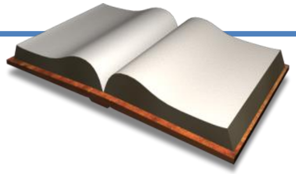
- **Programming**
  - Just the action of **writing codes** for programs (Creating programs).

- **"But what exactly is a program (in our examples)?"**
  - **Output**
    - The program decides what exact text to show on the screen.
  - **Input**
    - based on its instructions and on the text that the player typed on the keyboard.
  - A **program** is a collection of instructions.

What Exactly is Programming?

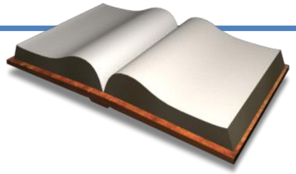
- **"What kind of instructions?"**
  - Expressions
  - Function calls
  - Conditions
  - **flow control** statements
    - if, while and break
  - The print() function, input() function
    - This is called I/O (input and output) functions
  - …

Things Covered In This Chapter(1/3)



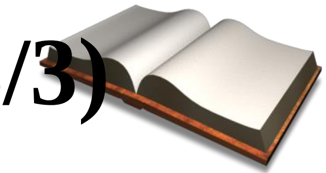
- `import` statements
- Modules
- Arguments
- `while` statements
- Conditions
- Blocks
- Comparison operators

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.

Next Time

- Labs in this week:
  - Lab1: 과제 3-1
  - Lab2: 과제 3-2
- Next lecture:
  - 4-P03. Dragon Realm