

Variables

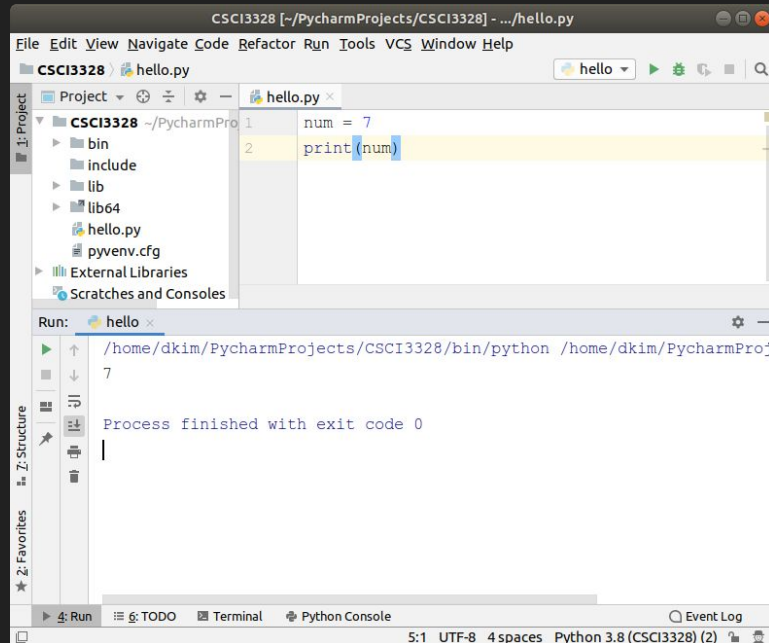
Dr. Dongchul Kim

Variables

- Variables and Memory:
 - Variables are reserved memory locations used to store values.
 - They serve as labels to identify specific memory spaces.
- Creating Variables:
 - The first assignment of a value to a variable creates that variable.
 - In Python, there's no need for explicit variable declaration.
- Assignment Example:
 - For example: `num = 3`
 - This statement declares the variable `num` and assigns the integer value `3` to it.
- Assignment Operator (=):
 - The `=` sign is an assignment operator.
 - It assigns the right-hand-side operand (e.g., `3`) to the memory space of the left-hand-side operand (e.g., `num`).

Assignment Operator

You can confirm the assigned value by printing the variable's value.
For example,



The screenshot shows the PyCharm IDE interface. The main editor window displays a Python file named `hello.py` with the following code:

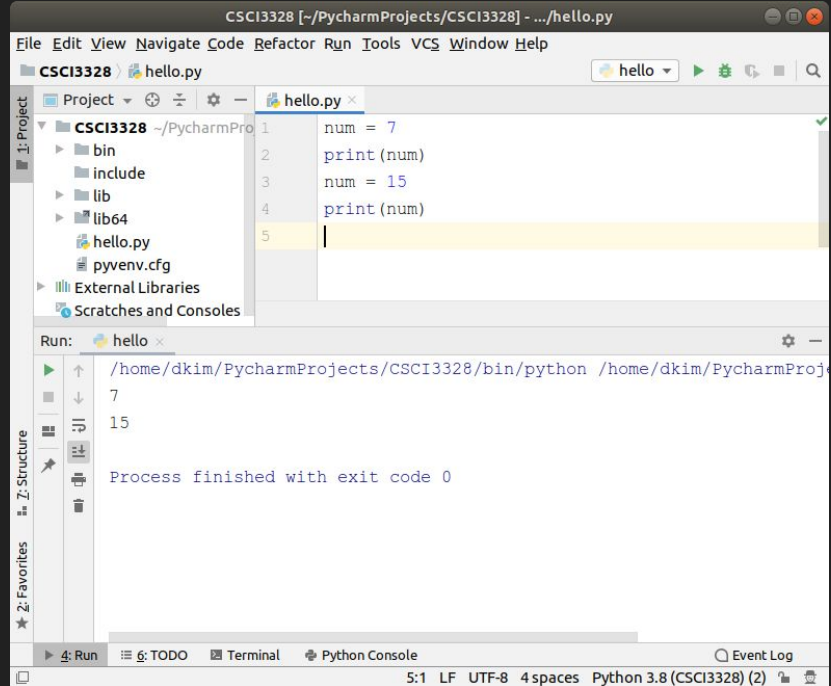
```
1 num = 7
2 print(num)
```

The `print(num)` line is highlighted in yellow. Below the editor, the Run tool window is open, showing the execution of the `hello` script. The command executed is `/home/dkim/PycharmProjects/CSCI3328/bin/python /home/dkim/PycharmProj`, and the output is `7`. The status bar at the bottom indicates the file encoding is UTF-8, the indentation is 4 spaces, and the Python version is 3.8.

Assignment Operator

Assigned values can be updated by using the assignment operator (=) again.

For example,



The screenshot shows the PyCharm IDE interface. The main editor window displays a Python file named `hello.py` with the following code:

```
1 num = 7
2 print(num)
3 num = 15
4 print(num)
5
```

The Run window at the bottom shows the execution output for the `hello` script:

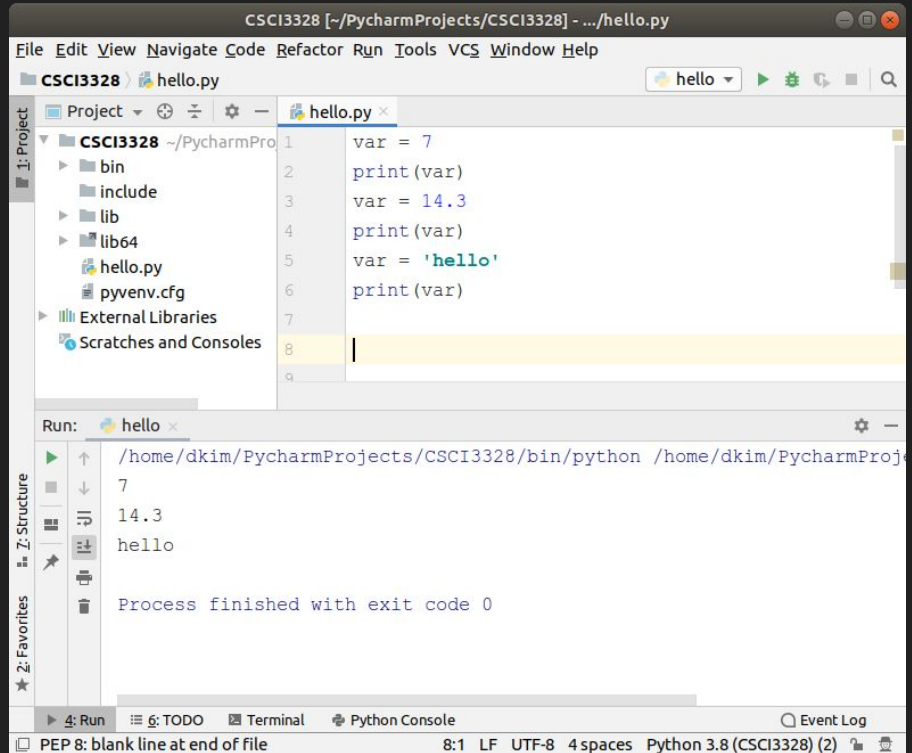
```
/home/dkim/PycharmProjects/CSCI3328/bin/python /home/dkim/PycharmProjects
7
15
Process finished with exit code 0
```

The status bar at the bottom indicates the file encoding is UTF-8, the indentation is 4 spaces, and the Python version is 3.8 (CSCI3328) (2).

Variables

The interpreter allocates memory and determines what can be stored in variables based on their data type.

Assigning different data types allows storage of integers, decimals, or characters in these variables.



The screenshot shows the PyCharm IDE interface. The main editor window displays a Python file named `hello.py` with the following code:

```
1 var = 7
2 print(var)
3 var = 14.3
4 print(var)
5 var = 'hello'
6 print(var)
7
8
9
```

The Run window at the bottom shows the execution output:

```
/home/dkim/PycharmProjects/CSCI3328/bin/python /home/dkim/PycharmProj
7
14.3
hello

Process finished with exit code 0
```

The status bar at the bottom indicates the current cursor position and file encoding: `PEP 8: blank line at end of file 8:1 LF UTF-8 4 spaces Python 3.8 (CSCI3328) (2)`.

Variable types

- Python features five standard data types:
 - Numbers
 - Strings
 - Lists
 - Tuples
 - Dictionaries
- Each of these will be explained in detail later.

Naming rule

- Variable name is an identifier. It means the variable name should be unique in its scope.
- Variable name begins with letter or underscore character
- Alpha-numeric and underscores are allowed in the rest of name
- Variable names are case-sensitive

Naming rule

- These are reserved words and cannot be used as variables.

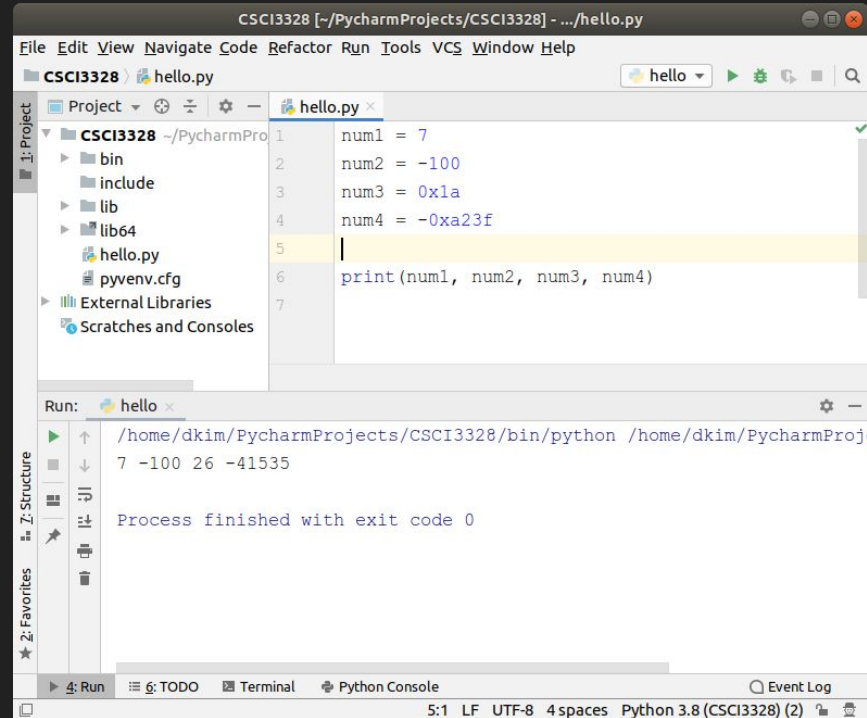
and	exec	not
assert	finally	or
break	for	pass
class	from	print
continue	global	raise
def	if	return
del	import	try
elif	in	while
else	is	with
except	lambda	yield

Variable types

- Python has five standard data types
 - Numbers
 - int (signed integers)
 - float (floating point real values)
 - String
 - List
 - Tuple
 - Dictionary

Numbers

Integer



The screenshot shows the PyCharm IDE interface. The main editor window displays a Python file named `hello.py` with the following code:

```
1 num1 = 7
2 num2 = -100
3 num3 = 0x1a
4 num4 = -0xa23f
5
6 print(num1, num2, num3, num4)
7
```

The code is executed, and the Run console shows the output:

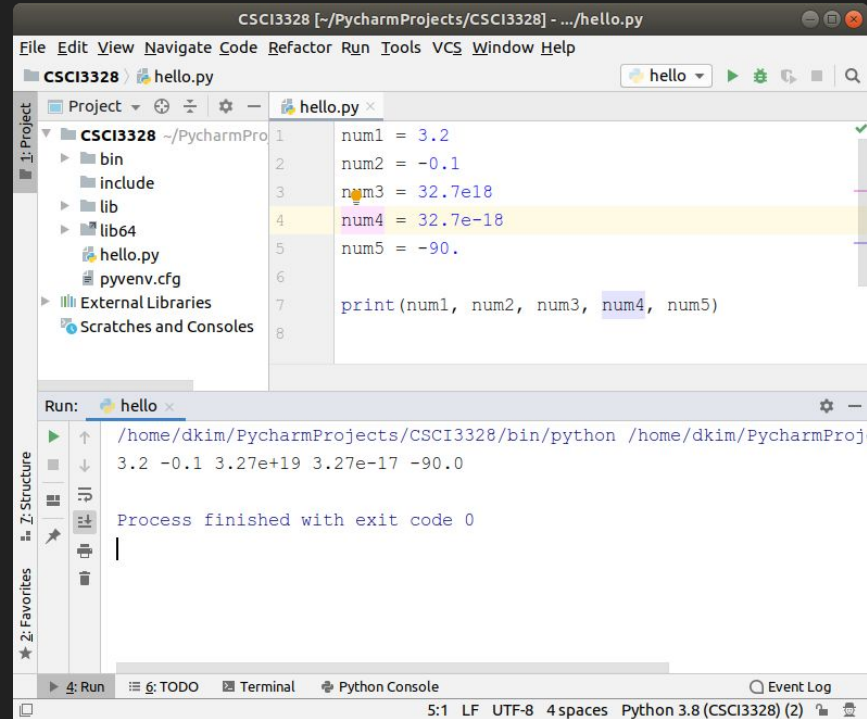
```
/home/dkim/PycharmProjects/CSCI3328/bin/python /home/dkim/PycharmProj
7 -100 26 -41535

Process finished with exit code 0
```

The status bar at the bottom indicates the file encoding is UTF-8, 4 spaces, and the Python version is 3.8 (CSCI3328) (2).

Numbers

Float



The screenshot shows the PyCharm IDE interface. The main editor window displays a Python file named `hello.py` with the following code:

```
1 num1 = 3.2
2 num2 = -0.1
3 num3 = 32.7e18
4 num4 = 32.7e-18
5 num5 = -90.
6
7 print(num1, num2, num3, num4, num5)
8
```

The `num4` variable is highlighted in yellow. Below the editor, the Run console shows the output of the script:

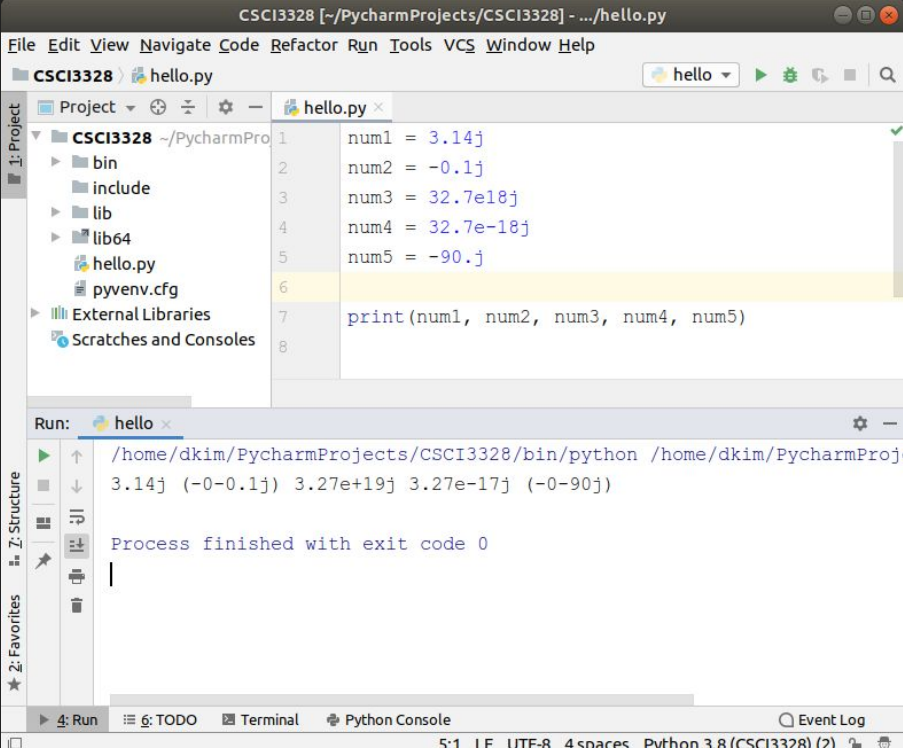
```
Run: hello x
/home/dkim/PycharmProjects/CSCI3328/bin/python /home/dkim/PycharmProj
3.2 -0.1 3.27e+19 3.27e-17 -90.0
Process finished with exit code 0
```

The status bar at the bottom indicates the file encoding is UTF-8, the editor uses 4 spaces for indentation, and the Python version is 3.8 (CSCI3328) (2).

Numbers

Complex

A complex number consists of an ordered pair of real floating-point numbers denoted by $x + yj$, where x and y are real numbers and j is the imaginary unit. [\[wiki\]](#)



```
CSCI3328 [-~/PycharmProjects/CSCI3328] - .../hello.py
File Edit View Navigate Code Refactor Run Tools VCS Window Help
CSCI3328 hello.py hello
Project CSCI3328 ~/PycharmPro
  bin
  include
  lib
  lib64
  hello.py
  pyenvv.cfg
  External Libraries
  Scratches and Consoles
1 num1 = 3.14j
2 num2 = -0.1j
3 num3 = 32.7e18j
4 num4 = 32.7e-18j
5 num5 = -90.j
6
7 print(num1, num2, num3, num4, num5)
8
Run: hello x
/home/dkim/PycharmProjects/CSCI3328/bin/python /home/dkim/PycharmProj
3.14j (-0-0.1j) 3.27e+19j 3.27e-17j (-0-90j)
Process finished with exit code 0
Run TODO Terminal Python Console Event Log
5:1 LF UTF-8 4 spaces Python 3.8 (CSCI3328) (2)
```

Python F-String Format

Introduction to F-Strings:

- F-strings (formatted string literals) provide a concise and readable way to format strings in Python.

Usage:

- F-strings are created by prefixing a string with 'f' or 'F'.
- Inside an f-string, you can embed expressions inside curly braces {}.

Example:

- Syntax: `f"Hello, {name}!"`
- In this example, `name` is a variable, and its value will be inserted into the string.

Benefits:

- F-strings offer a convenient way to create dynamic strings.
- They improve code readability and reduce string concatenation complexity.

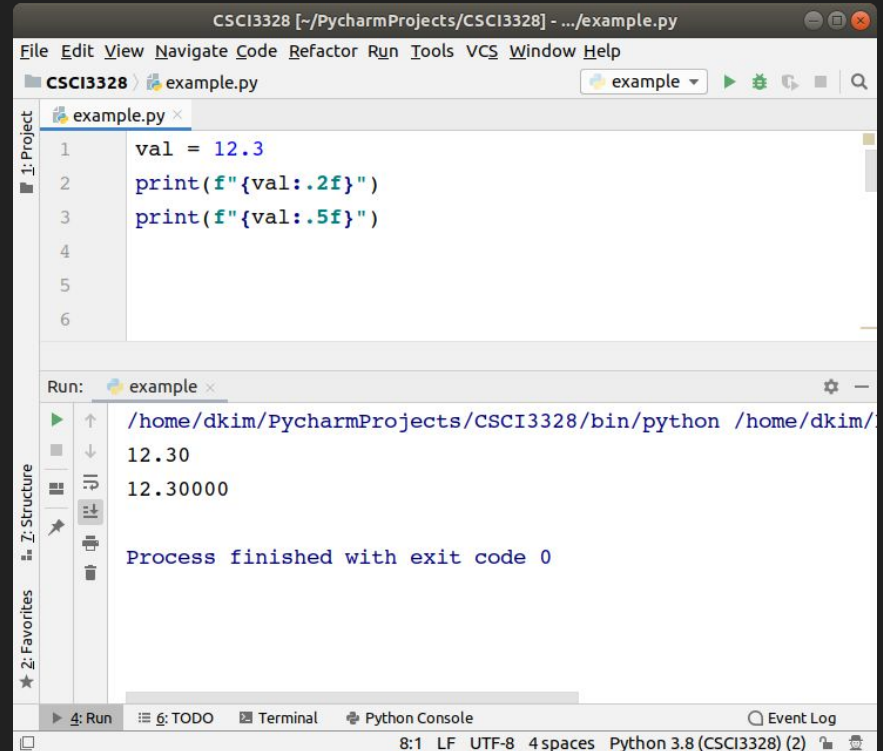
Precision and Formatting:

- F-strings can specify precision for floating-point numbers and control formatting.

Python F-String Format: floats

Floating-point values in Python can be specified with the 'f' suffix.

Precision, or the number of decimal places, is indicated immediately after the dot character.



The screenshot shows the PyCharm IDE interface. The main editor window displays a Python script named `example.py` with the following code:

```
1 val = 12.3
2 print(f"{val:.2f}")
3 print(f"{val:.5f}")
4
5
6
```

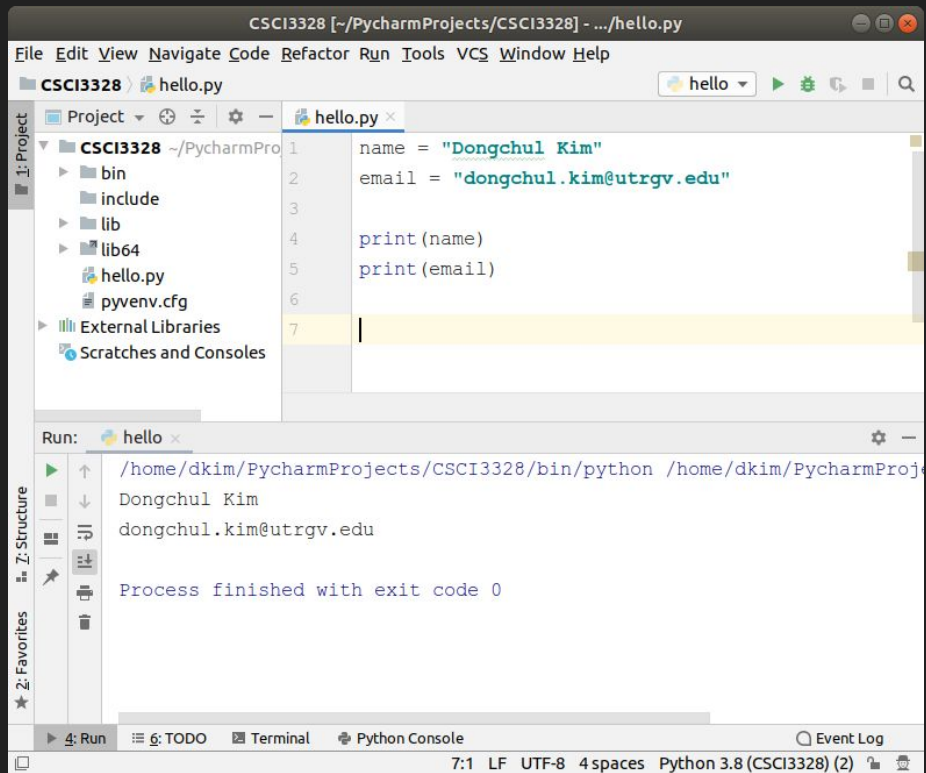
Below the editor, the Run console shows the output of the script:

```
Run: example x
/home/dkim/PycharmProjects/CSCI3328/bin/python /home/dkim/
12.30
12.30000
Process finished with exit code 0
```

The status bar at the bottom indicates the file encoding is UTF-8, the line length is 8:1, and the Python version is Python 3.8 (CSCI3328) (2).

String

strings are defined as a sequence of characters enclosed in either single or double quotation marks.



The screenshot shows the PyCharm IDE interface. The main editor window displays a Python file named `hello.py` with the following code:

```
1 name = "Dongchul Kim"
2 email = "dongchul.kim@utrgv.edu"
3
4 print(name)
5 print(email)
6
7
```

The code is executed, and the Run window shows the output:

```
/home/dkim/PycharmProjects/CSCI3328/bin/python /home/dkim/PycharmProj
Dongchul Kim
dongchul.kim@utrgv.edu

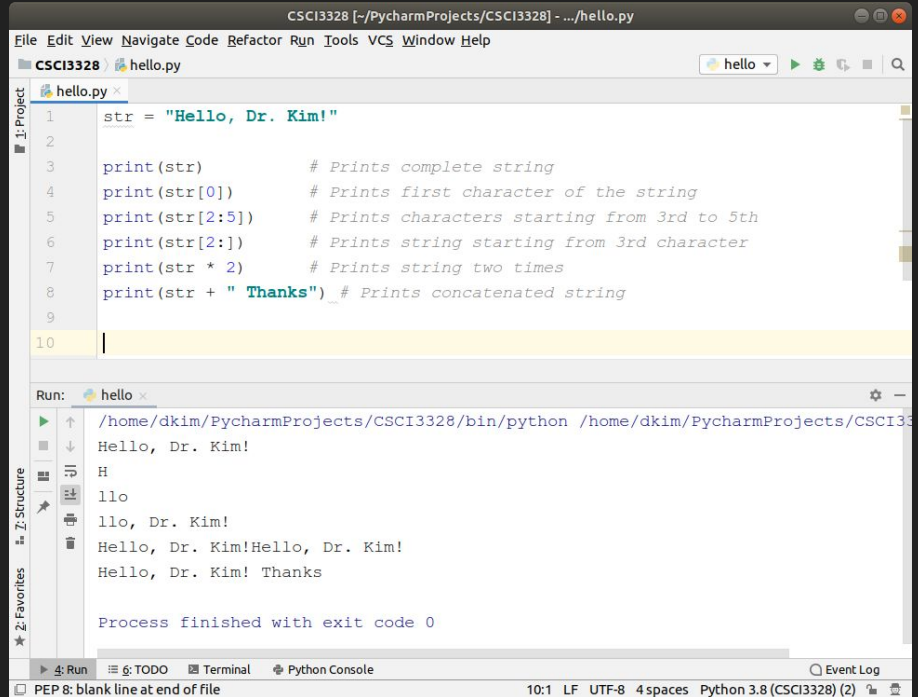
Process finished with exit code 0
```

The status bar at the bottom indicates the file encoding is UTF-8, the indentation is 4 spaces, and the Python version is 3.8 (CSCI3328).

String

Subsets of strings can be taken using the slice operator (`[]` and `[:]`) with indexes starting at `0` in the beginning of the string and working their way from `-1` at the end.

The plus (+) sign is the string concatenation operator and the asterisk (*) is the repetition operator.



The screenshot shows a PyCharm IDE window titled "CSCI3328 [-/PycharmProjects/CSCI3328] - .../hello.py". The editor displays the following Python code:

```
1 str = "Hello, Dr. Kim!"
2
3 print(str)           # Prints complete string
4 print(str[0])       # Prints first character of the string
5 print(str[2:5])     # Prints characters starting from 3rd to 5th
6 print(str[2:])      # Prints string starting from 3rd character
7 print(str * 2)      # Prints string two times
8 print(str + " Thanks") # Prints concatenated string
9
10
```

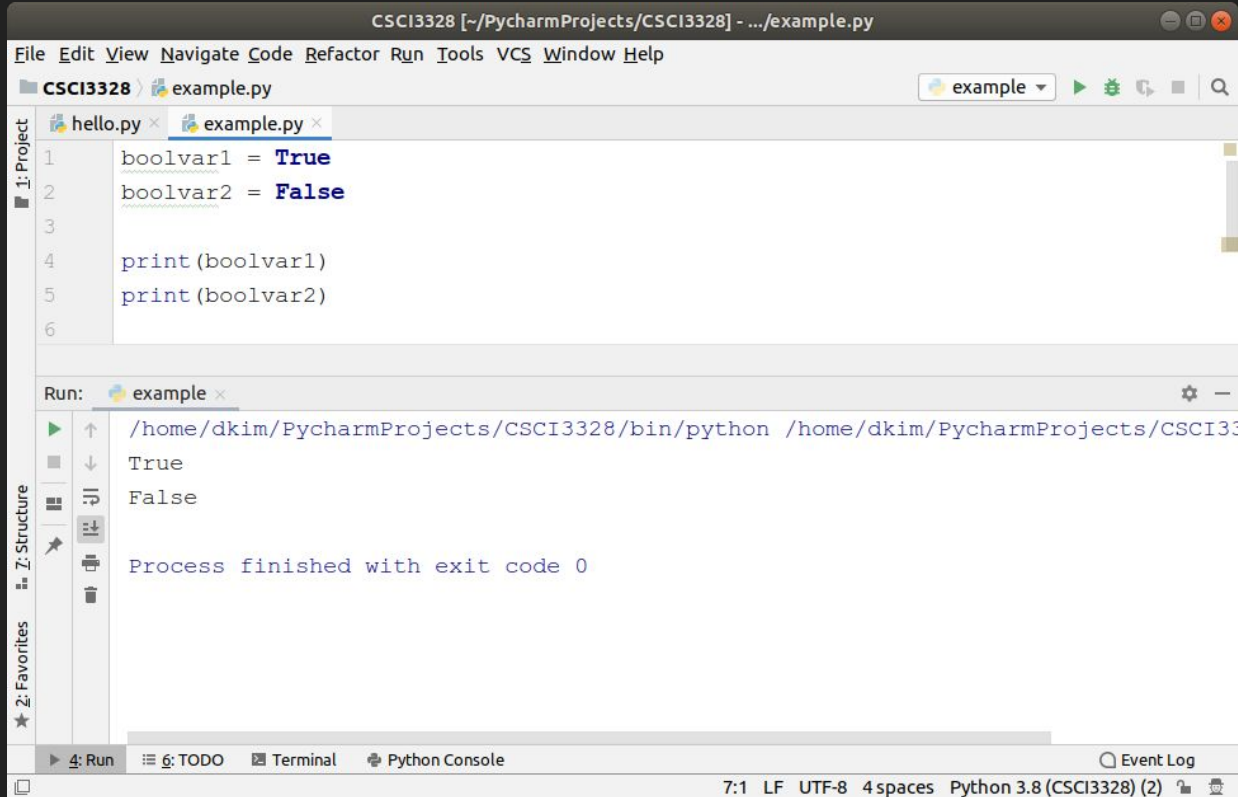
The Run window below shows the execution output:

```
hello
/home/dkim/PycharmProjects/CSCI3328/bin/python /home/dkim/PycharmProjects/CSCI3328/bin/python
Hello, Dr. Kim!
H
llo
llo, Dr. Kim!
Hello, Dr. Kim!Hello, Dr. Kim!
Hello, Dr. Kim! Thanks

Process finished with exit code 0
```

The status bar at the bottom indicates "PEP 8: blank line at end of file" and "10:1 LF UTF-8 4 spaces Python 3.8 (CSCI3328) (2)".

Boolean values in python



The screenshot shows the PyCharm IDE interface. The main editor window displays a Python file named `example.py` with the following code:

```
1 boolvar1 = True
2 boolvar2 = False
3
4 print(boolvar1)
5 print(boolvar2)
6
```

Below the editor, the Run console shows the execution of the script. The command used is `/home/dkim/PycharmProjects/CSCI3328/bin/python /home/dkim/PycharmProjects/CSCI3328/...`. The output is:

```
True
False

Process finished with exit code 0
```

The status bar at the bottom indicates the file encoding is UTF-8 with 4 spaces, and the Python version is 3.8 (CSCI3328) (2).

Lab 5: Define and Display Variables

1. Objective:
 - Define six variables and assign appropriate values to them.
 - Display the variables along with their values using the `print` function.
2. Variables:
 - Your Name
 - Email
 - Expected GPA (0.0~4.0)
 - Level of Programming Skill (0~10)
 - Interest in Machine Learning (T/F)
 - Desired Job/Area After Graduation
3. Assignment Example:
 - Define and assign values to the variables.
 - Use the `print` function to display the variables and their values.
4. Submission:
 - Capture your code and the output together in a single image.
 - Upload the image to Blackboard.

Lab 5

Variable Definition (40 Points):

Correctly defining six variables: Name, Email, Expected GPA, Programming Skill Level, Interest in Machine Learning, Desired Job/Area After Graduation (40 Points)

Missing or incorrectly defined variables (0 Points)

Display Variables Using Print Function (30 Points):

Successfully using the print function to display each variable and its assigned value (30 Points)

Incomplete or incorrect use of the print function (0 Points)

Screenshot and Submission (30 Points):

Submitting a clear image capturing both the code and its output (30 Points)

Incomplete, unclear, or incorrect screenshot (0 Points)

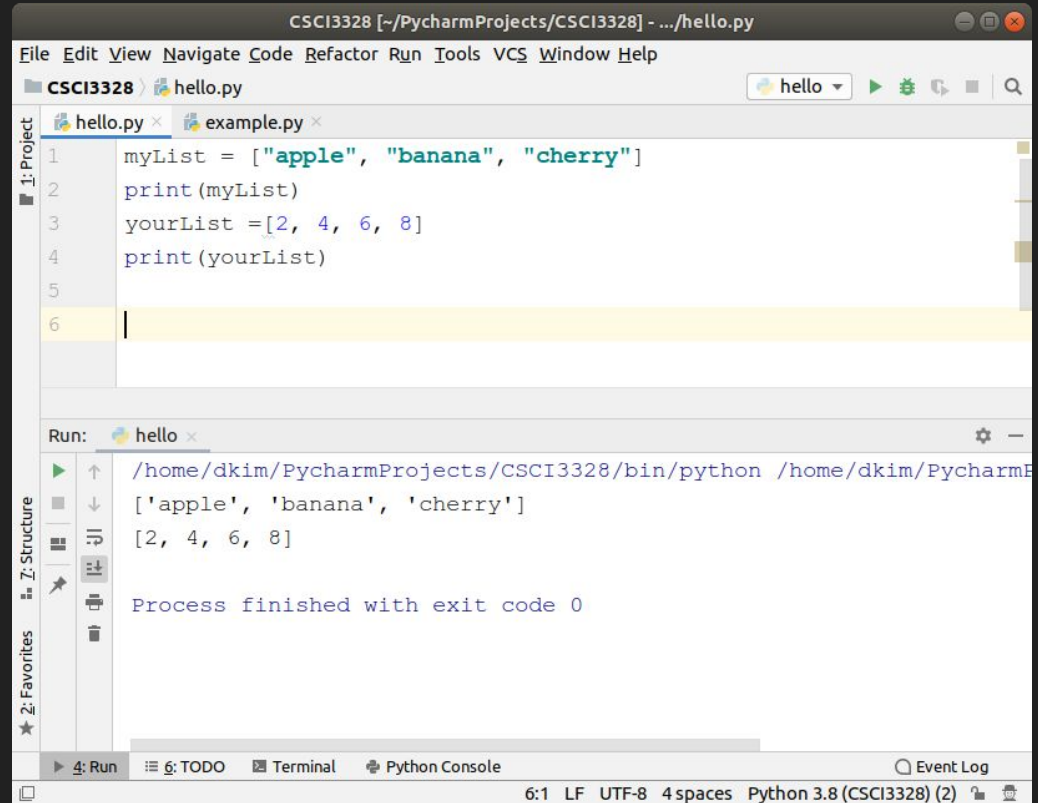
Total Points: 100

List

A list is an ordered and changeable collection in Python.

It allows duplicate members.

More details about lists will be covered later.



The screenshot shows the PyCharm IDE interface. The main editor window displays a Python script named `hello.py` with the following code:

```
1 myList = ["apple", "banana", "cherry"]
2 print(myList)
3 yourList = [2, 4, 6, 8]
4 print(yourList)
5
6
```

The `print(myList)` and `print(yourList)` lines are highlighted in yellow. Below the editor, the `Run` window shows the execution output:

```
Run: hello x
/home/dkim/PycharmProjects/CSCI3328/bin/python /home/dkim/PycharmE
['apple', 'banana', 'cherry']
[2, 4, 6, 8]

Process finished with exit code 0
```

The status bar at the bottom indicates the file encoding is UTF-8, 4 spaces, and Python 3.8 (CSCI3328) (2).