**CSIT110 / CSIT810**

# Python

Lecture 2

Dr. Joseph Tonien

School of Computing and Information Technology
University of Wollongong

# Objectives

Understanding of:

- Variables
- Strings, numbers
- Convert a number to string
- `print` function
- Comments
- Naming convention

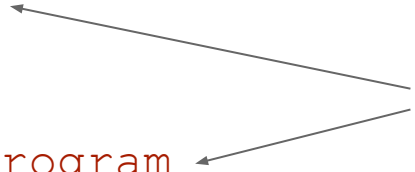
# Comments

# Let's look at the first example:

```
# hello.py
# By Joseph Tonien

# My first Python program

print("Hello World")

a = 1
b = 2
c = a + b
print(c)
```

What do you think this program will do?

```
# hello.py
# By Joseph Tonien

# My first Python program

print("Hello World")

a = 1
b = 2
c = a + b
print(c)
```



These are called comments

We can put comments anywhere in the program:

- to make the program clearer for people to read and maintain
- to help people understand our program better, especially, if our program has a special logic that needs explanation
- comments are not code, so they will NOT be executed

# The `print` function

```
# hello.py
# By Joseph Tonien

# My first Python program

print("Hello World")

a = 1
b = 2
c = a + b
print(c)
```

What do you think these codes do?

`print` is called a **function**, we will learn about function later

What is the difference between

```
print("Hello World")
print(c)
```

```
# hello.py
# By Joseph Tonien

# My first Python program

print("Hello World")

a = 1
b = 2
c = a + b
print(c)
```

What do you think these codes do?

`print` is called a **function**, we will learn about function later

What is the difference between

```
print("Hello World")
print(c)
```

this is a string

this is a number

```
# hello.py
# By Joseph Tonien

# My first Python program

print("Hello World")

a = 1
b = 2
c = a + b
print(c)
```

What do you think these codes do?

`print` is called a **function**, we will learn about function later

What is the difference between

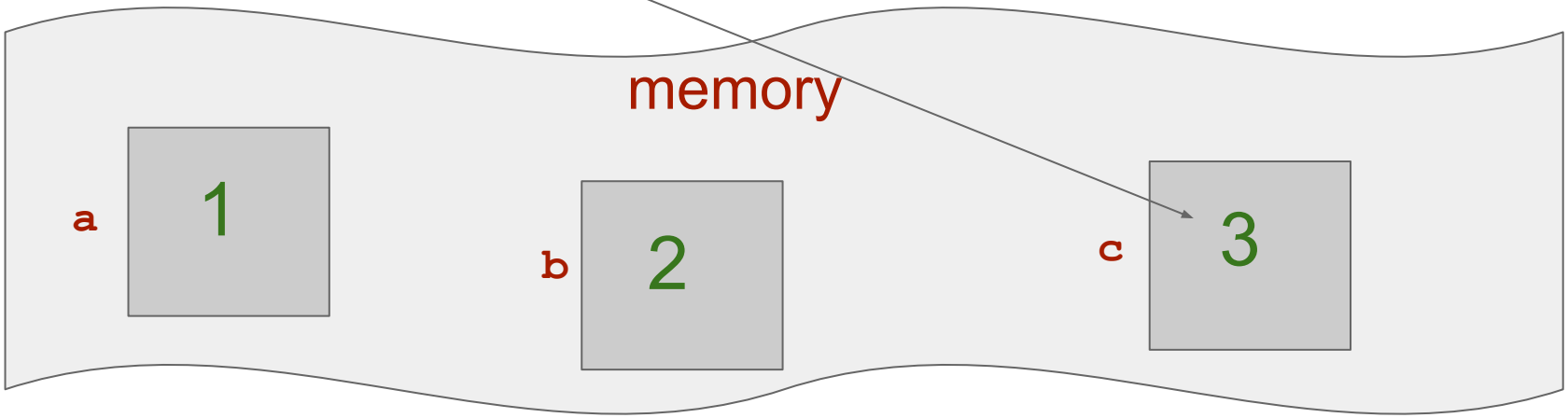
```
print(c)
print("c")
```

# Variables

```
# hello.py
# By Joseph Tonien

# My first Python program

print("Hello World")

a = 1
b = 2
c = a + b
print(c)
```

`a, b, c` are called variables

variables are reserved memory locations to store values



# The `print` function

# The `print` function

```
# diamond.py
# By Joseph Tonien

# A program to demonstrate the use of print function
# Print out a shape of a diamond

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

What do you think this program will do?

# The `print` function

```
# unimovies.py
# By Joseph Tonien

# A program to demonstrate the use of print function
# Print out Unimovies program

print("Welcome to Unimovies!")
print()
print("Thursday July 30 at 7.15pm: Inside Out")
print()
print("Starring: Diane Lane, Amy Poehler, Mindy Kaling")
print("Released: June 18, 2015")
print("Rating:   PG")
print("Runtime:  102 minutes")
print("Websites: http://movieweb.com/movie/inside-out-2015")
```

this code prints a newline

`print()` will prints a new empty line

# String concatenation (addition)

# String concatenation (addition)

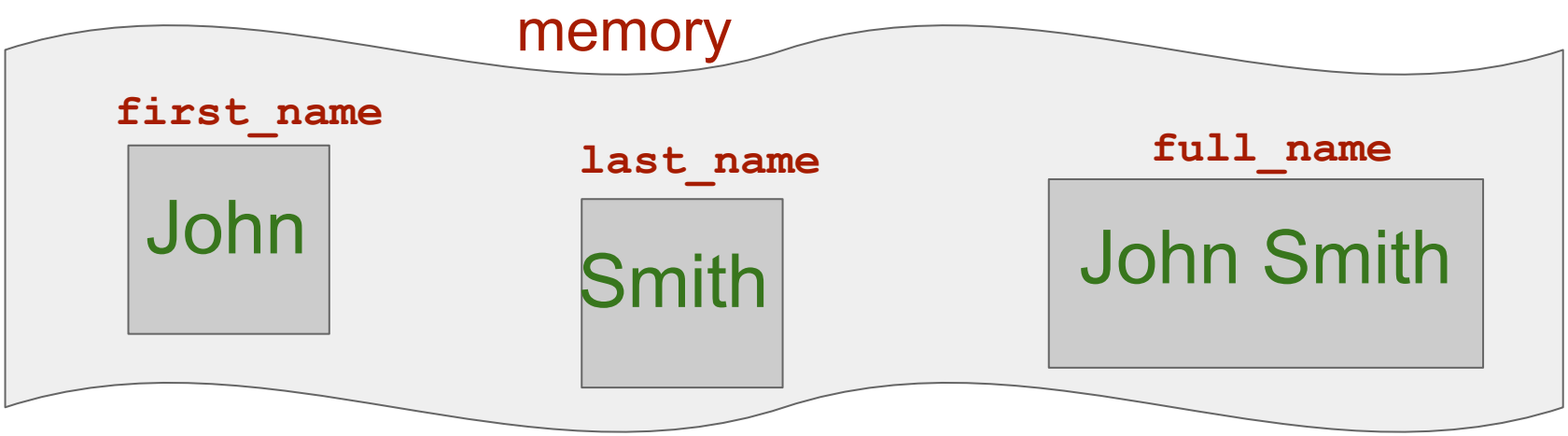
```
# fullname.py
# By Joseph Tonien

first_name = "John"
last_name = "Smith"

full_name = first_name + " " + last_name

print("My name is " + full_name + ".")
```

What do you think this program will do?

# String concatenation (addition)

```
# fullname.py
# By Joseph Tonien

first_name = "John"
last_name = "Smith"

full_name = first_name + " " + last_name

print("My name is " + full_name + ".")
```

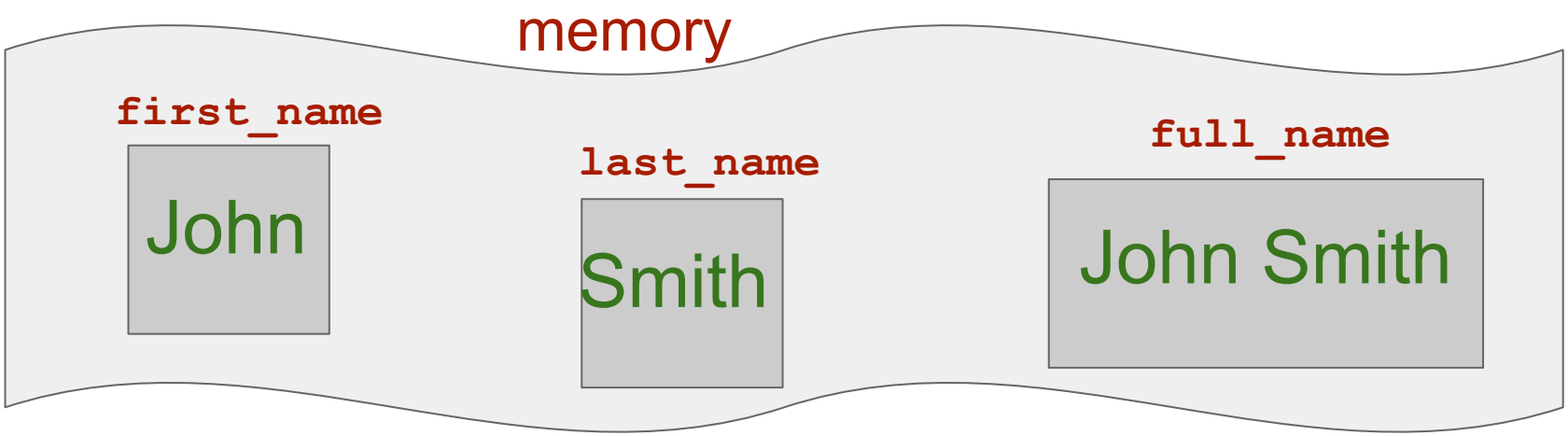
3 variables in the memory



# String concatenation (addition)

```
# fullname.py
# By Joseph Tonien

first_name = "John"
last_name = "Smith"

full_name = first_name + " " + last_name

print("My name is " + full_name + ".")
```

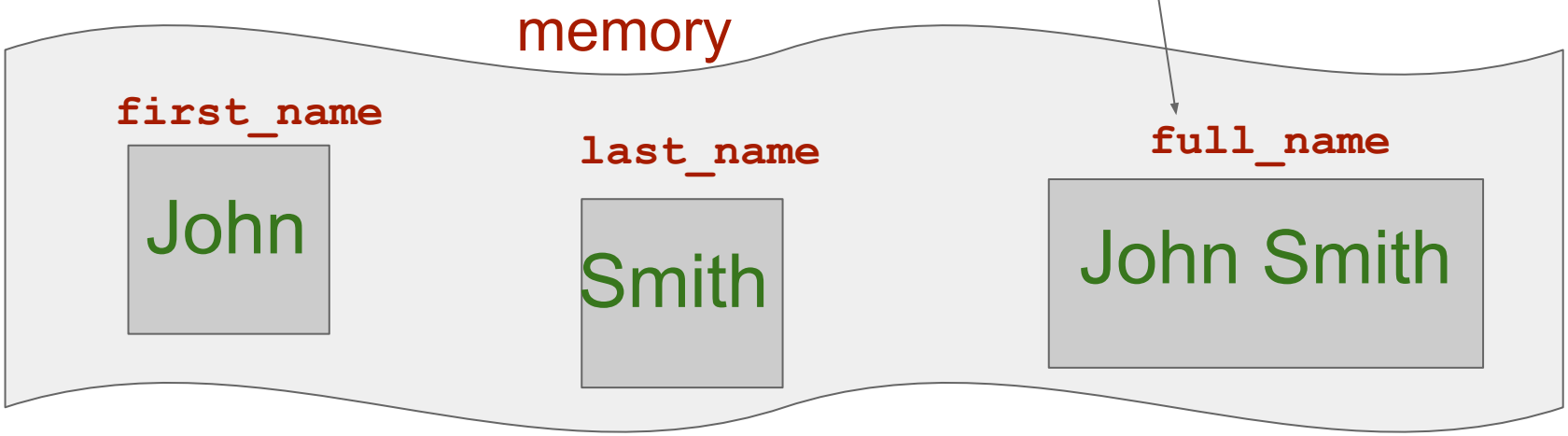
this is called
string concatenation
or string addition



# String concatenation (addition)

```
# fullname.py
# By Joseph Tonien

first_name = "John"
last_name = "Smith"

full_name = first_name + " " + last_name

print("My name is " + full_name + ".")
```

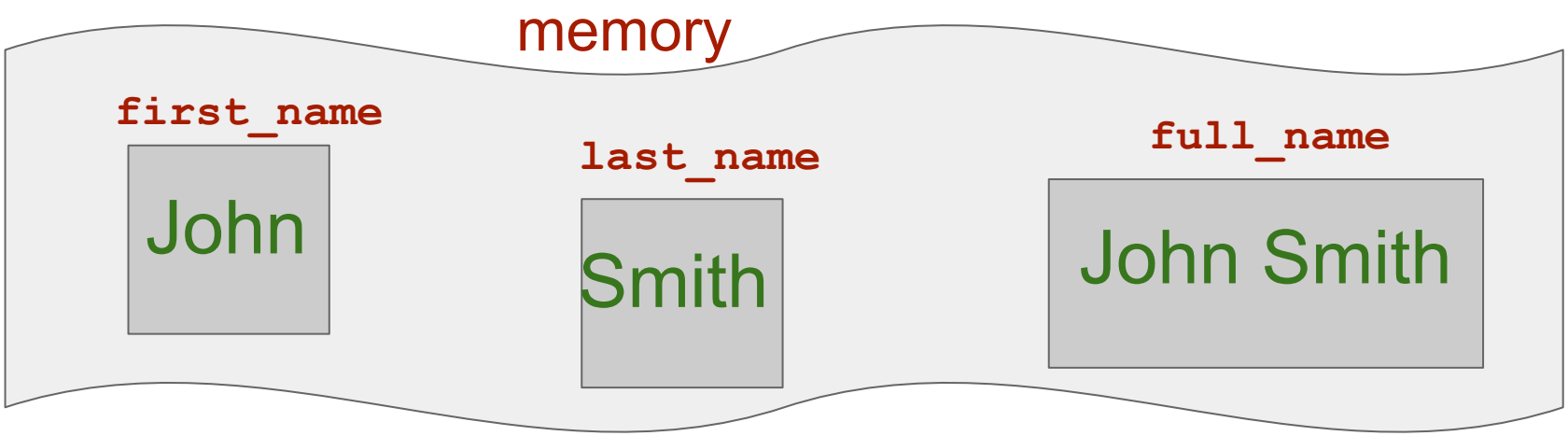
"John" + " " + "Smith"



# String concatenation (addition)

```
# fullname.py
# By Joseph Tonien

first_name = "John"
last_name = "Smith"

full_name = first_name + " " + last_name

print("My name is " + full_name + ".")
```

string concatenation → `"My name is " + "John Smith" + "."`



# String concatenation (addition)

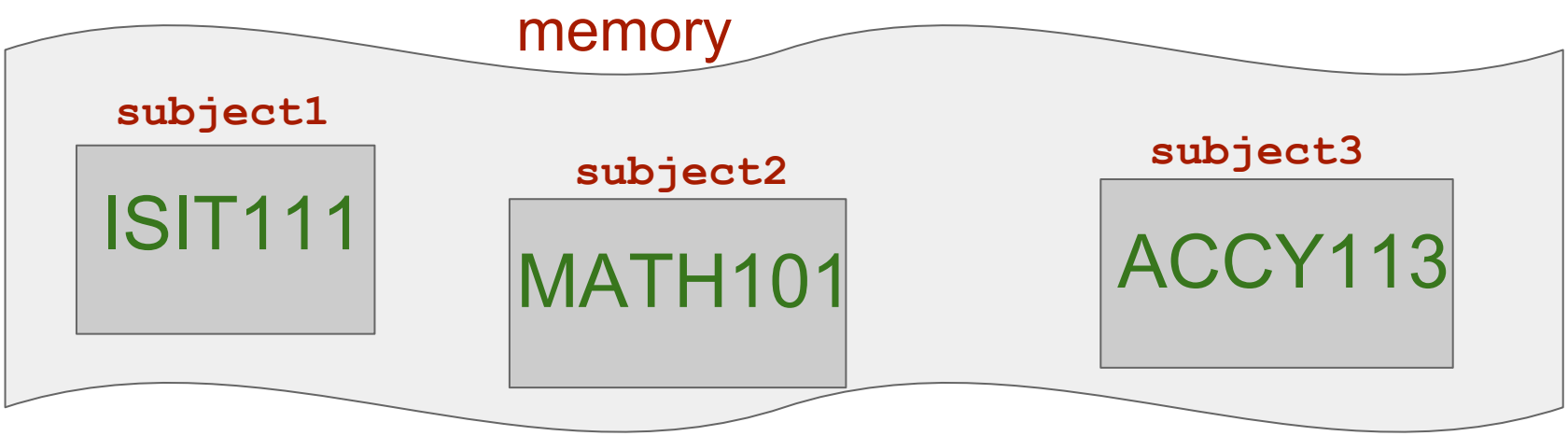
```
# subject.py
# By Joseph Tonien

subject1 = "ISIT111"
subject2 = "MATH101"
subject3 = "ACCY113"
print("My enrolled subjects are: " + subject1 + ", " + subject2
+ ", " + subject3 + ".")
```

What do you think this program will do?

# String concatenation (addition)

```
# subject.py
# By Joseph Tonien

subject1 = "ISIT111"
subject2 = "MATH101"
subject3 = "ACCY113"
print("My enrolled subjects are: " + subject1 + ", " + subject2
+ ", " + subject3 + ".")
```

```
"My enrolled subjects are: " + "ISIT111" + ", " + "MATH101" + ", " + "ACCY113" + "."
```



# Translate number into string

```
# favorite_number.py
# By Joseph Tonien

fav_number = 7

print("My favorite number is " + fav_number)
```

What is wrong with this program?

```
# favorite_number.py
# By Joseph Tonien

fav_number = 7

print("My favorite number is " + fav_number)
```

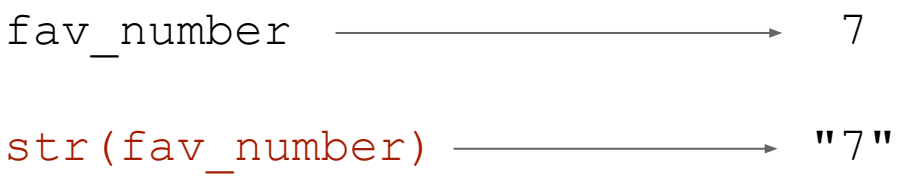


Python cannot add a string to a number

(some other programming languages can)

```
# favorite_number.py
# By Joseph Tonien

fav_number = 7

print("My favorite number is " + str(fav_number))
```

convert a number to a string




now we can do string concatenation

```
"My favorite number is " + "7"
```

# Convert a number to a string

```
# sum_of_two_numbers.py
# By Joseph Tonien

number1 = 10
number2 = 20
sum = number1 + number2
print("The sum of " + str(number1) + " and " + str(number2)
+ " is " + str(sum) + ".")
```

What do you think this program will do?

# Convert a number to a string

```
# sum_of_two_numbers.py
# By Joseph Tonien

number1 = 10
number2 = 20
sum = number1 + number2
print("The sum of " + str(number1) + " and " + str(number2)
+ " is " + str(sum) + ".")
```

```
"The sum of " + "10" + " and " + "20" + " is " + "30" + "."
```

# Naming convention

# Constant

```
# second_minute.py
# By Joseph Tonien

SECOND_PER_MINUTE = 60

minute = 5
second = minute * SECOND_PER_MINUTE

print(str(minute) + " minutes has " + str(second) + " seconds")
```

What do you think this program will do?

# Naming convention

```
first_name = "John"
last_name = "Smith"
full_name = first_name + " " + last_name

fav_number = 7

subject1 = "ISIT111"
subject2 = "MATH101"
subject3 = "ACCY113"

SECOND_PER_MINUTE = 60

minute = 5
second = minute * SECOND_PER_MINUTE
```

`lower_case_with_underscores` for normal variables

`UPPER_CASE_WITH_UNDERSCORES` for constant

# Keywords

The following list shows the Python keywords. These are reserved words and we CANNOT use them as constant or variable or any other identifier names.

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