上海外国语大学
SHANGHAI INTERNATIONAL STUDIES UNIVERSITY

Computational Journalism

Lecture 2: Software for Programming

Ting Wang

Outlines

- The Process of Software Development
- Python Programming Environment

上海外国语大学
SHANGHAI INTERNATIONAL STUDIES UNIVERSITY

a management approach to software engineering

The Process of Software Development

The Process of Software Development

Software Crisis

- The First NATO Software Engineering Conference in 1968, Germany.
- How to cope with the difficulty of writing useful and efficient computer programs in the required time.

The Process of Software Development

Difficulties in Software Development

1. Projects running over-budget
2. Projects running over-time
3. Software was very inefficient
4. Software was of low quality
5. Software often did not meet requirements
6. Projects were unmanageable and code difficult to maintain
7. Software was never delivered

The Process of Software Development

What is Software Engineering

– Software engineering is the application of engineering to the **design**, **development**, **implementation**, **testing** and **maintenance** of software in a systematic method.

From Wikipedia

The Process of Software Development

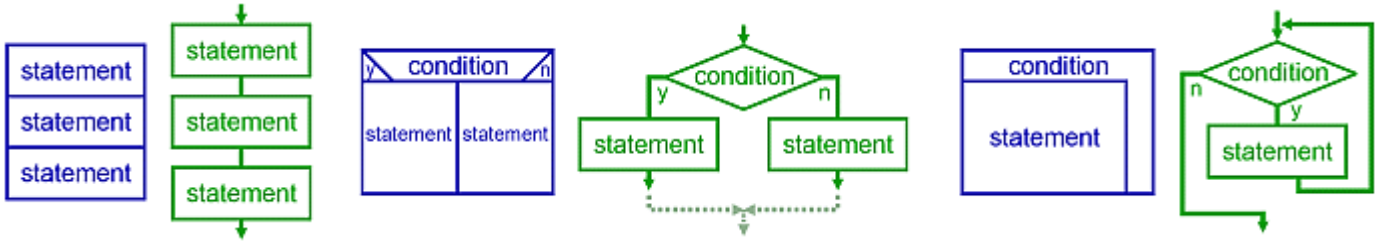
Basic Elements in Software Engineering

- Development Stages
- Management Pipeline
- Demands Changing
- Cooperative Team Work
- Professional Expert Participation

The Process of Software Development

Programming Paradigm 编程范式

- Structured Programming (1)结构化编程
  - Control Structure
    1. Sequence
    2. Selection: *if..then..else..endif, switch*
    3. Iteration: *while, repeat, for, do...until*

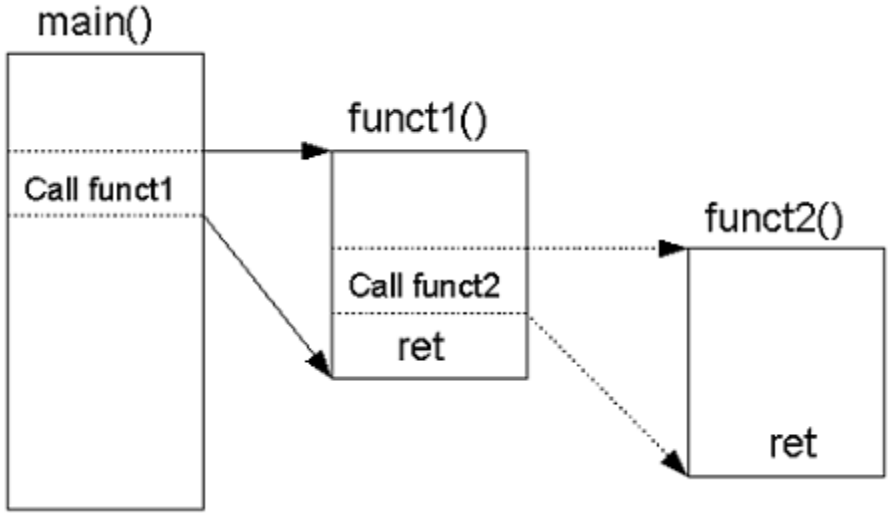


The Process of Software Development

Programming Paradigm 编程范式

- Structured Programming (2) 结构化编程
  - Subroutines

    子程序



The Process of Software Development

Programming Paradigm 编程范式

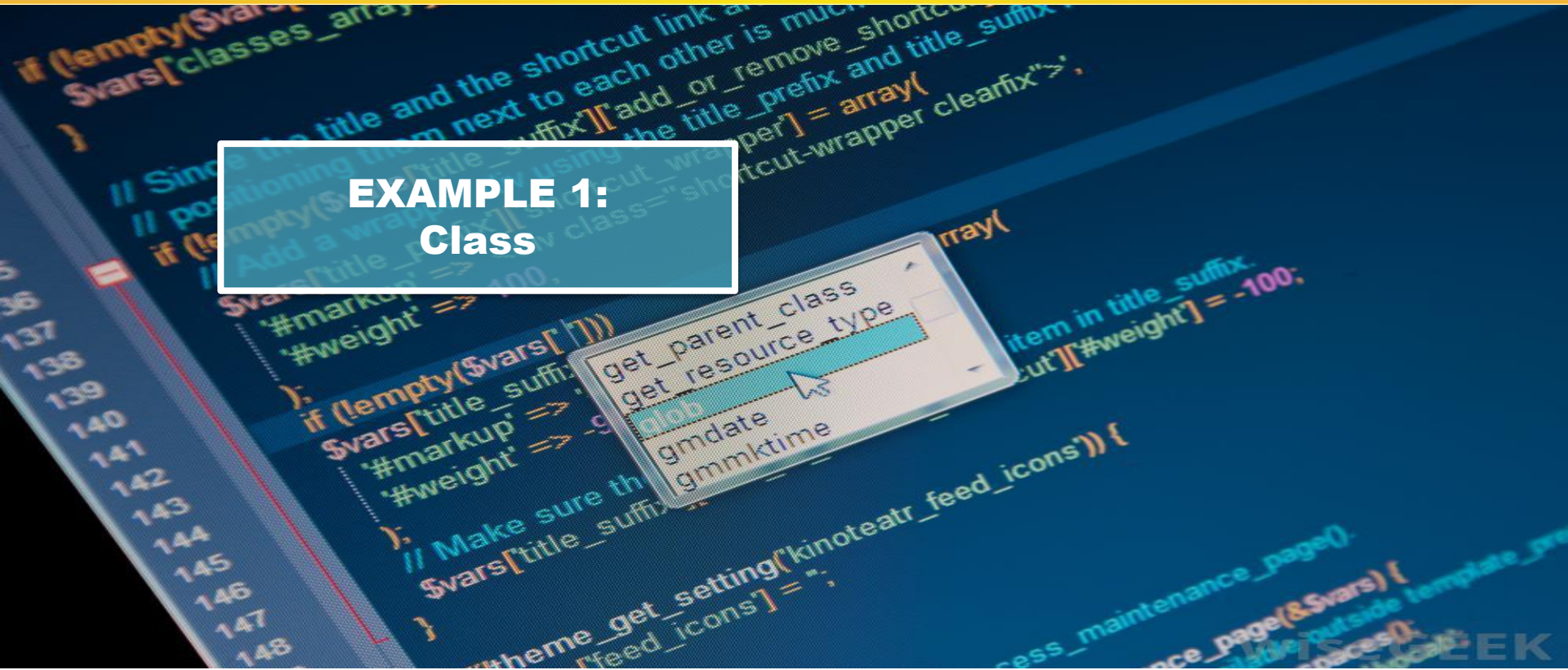
- Object Oriented Programming(1)
  面向对象编程　针对物件的编程
  - Object 对象
  - Class 类
  - Attribute 属性
  - Method 方法

The Process of Software Development

Programming Paradigm 编程范式

- Object Oriented Programming(2)
  面向对象编程
  - Encapsulation 封装
  - Inheritance 继承
  - Polymorphism 多态

The Process of Software Development



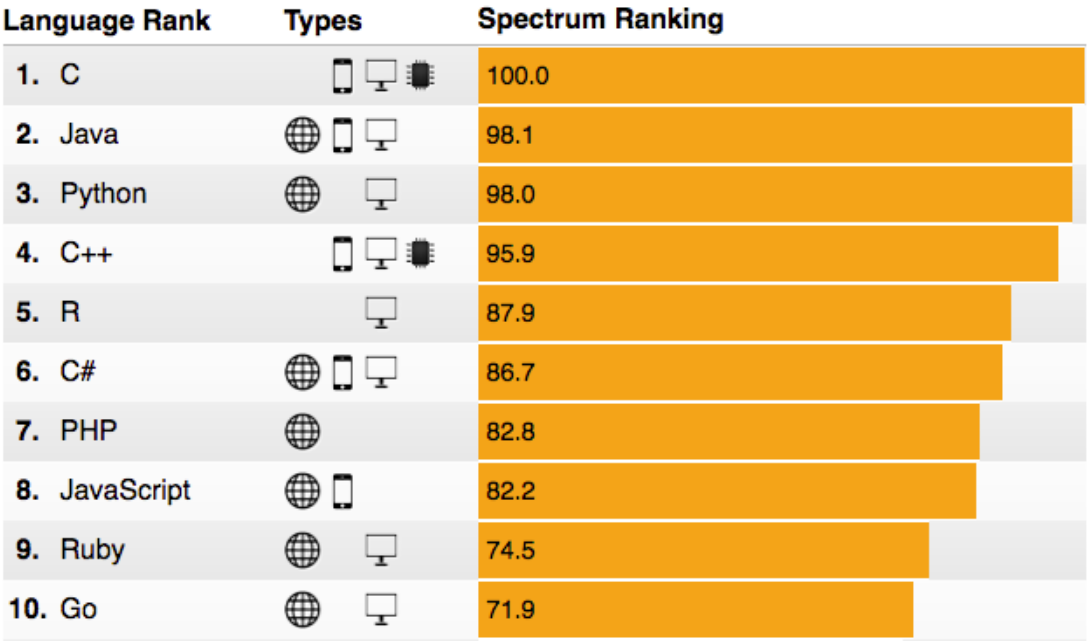
The Process of Software Development

```python
class Customer(object):

    name =''
    password =''

    def __init__(self, name, password):
        self.__name = name
        self.__password = password
        print ('Name: %s' %self.__name)

    def get_validation(self, password):
        if password=='sisu':
            return 1
        else:
            return 0

class Student(Customer):

    name =''
    password =''
    studentID = ''

    def __init__(self, name, password, studentID):
        Customer.__init__(self, name, password)
        self.studentID = studentID

    def print_studentID(self):
        return self.studentID

    def get_validation(self, password):
        if password=='shisu':
            return 'Passed'
        else:
            return 'Failed'

Thomas = Customer('Thomas Edison', 'sisu')
print('Thomas.get_validation() =', Thomas.get_validation('sisu'))
Albert = Student('Albert Einstein', 'sisu', '20160001')
print('Albert.print_studentID() =', Albert.print_studentID())
print('Albert.get_validation() =', Albert.get_validation('sisu'))
```

The Process of Software Development

IEEE Spectrum

http://spectrum.ieee.org/computing/software/the-2016-top-programming-languages

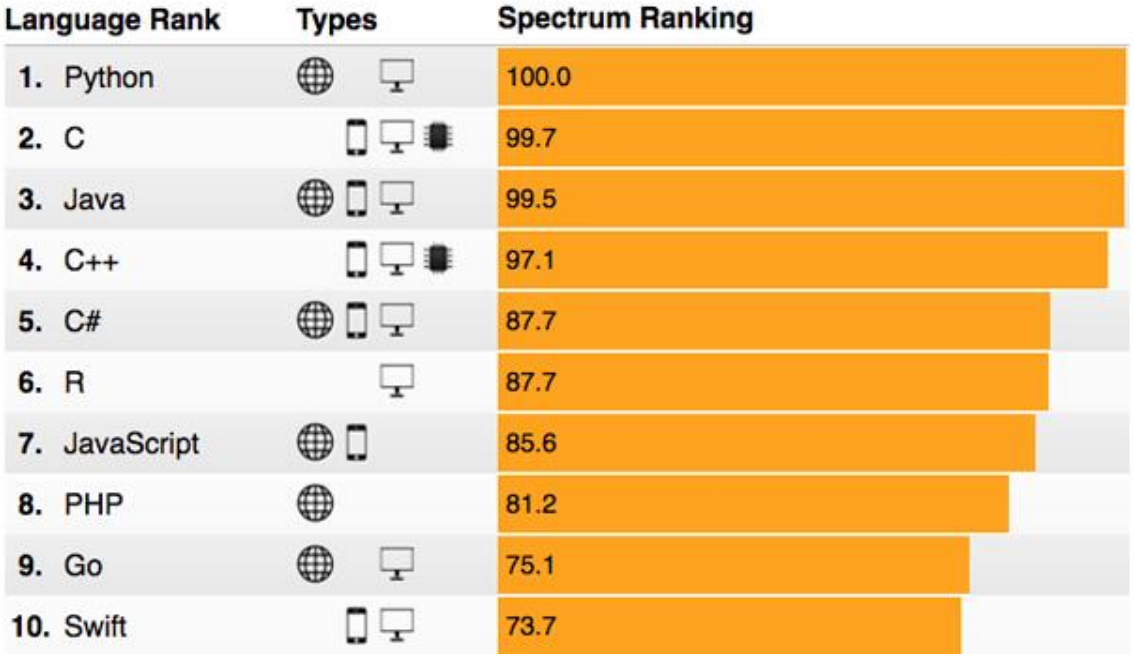


Language Rank	Types	Spectrum Ranking
1. C		100.0
2. Java		98.1
3. Python		98.0
4. C++		95.9
5. R		87.9
6. C#		86.7
7. PHP		82.8
8. JavaScript		82.2
9. Ruby		74.5
10. Go		71.9

The Process of Software Development

IEEE Spectrum

https://spectrum.ieee.org/computing/software/the-2017-top-programming-languages

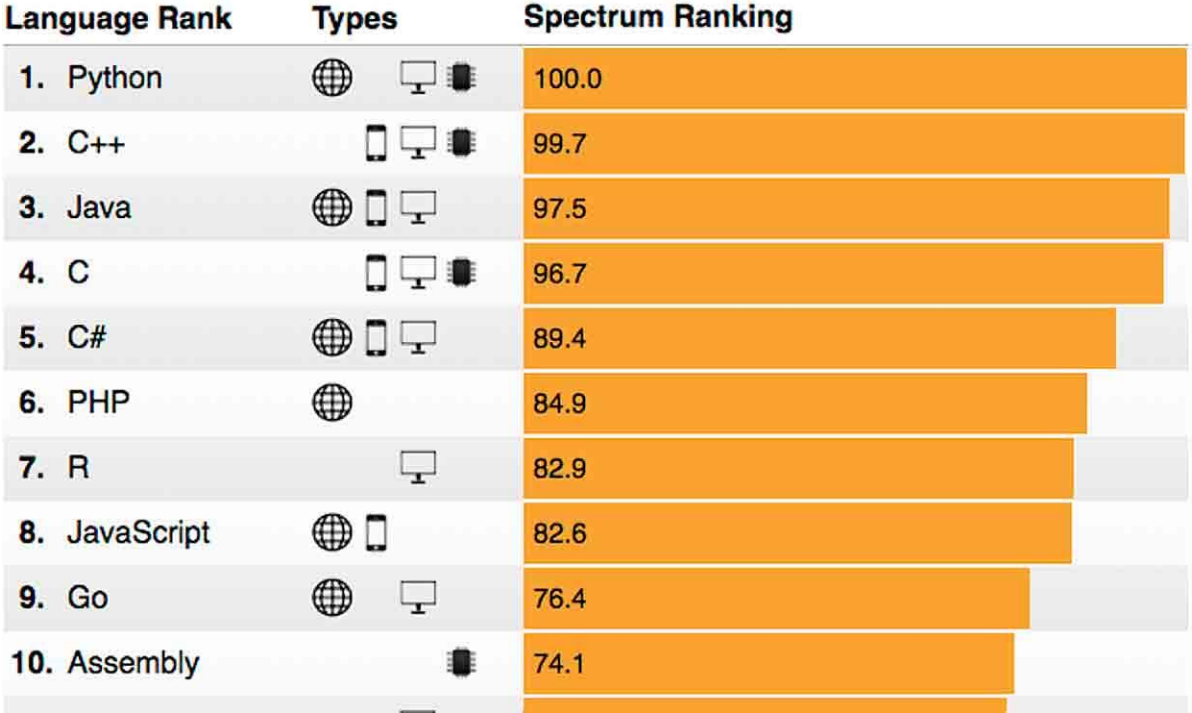


Language Rank	Types	Spectrum Ranking
1. Python		100.0
2. C		99.7
3. Java		99.5
4. C++		97.1
5. C#		87.7
6. R		87.7
7. JavaScript		85.6
8. PHP		81.2
9. Go		75.1
10. Swift		73.7

The Process of Software Development

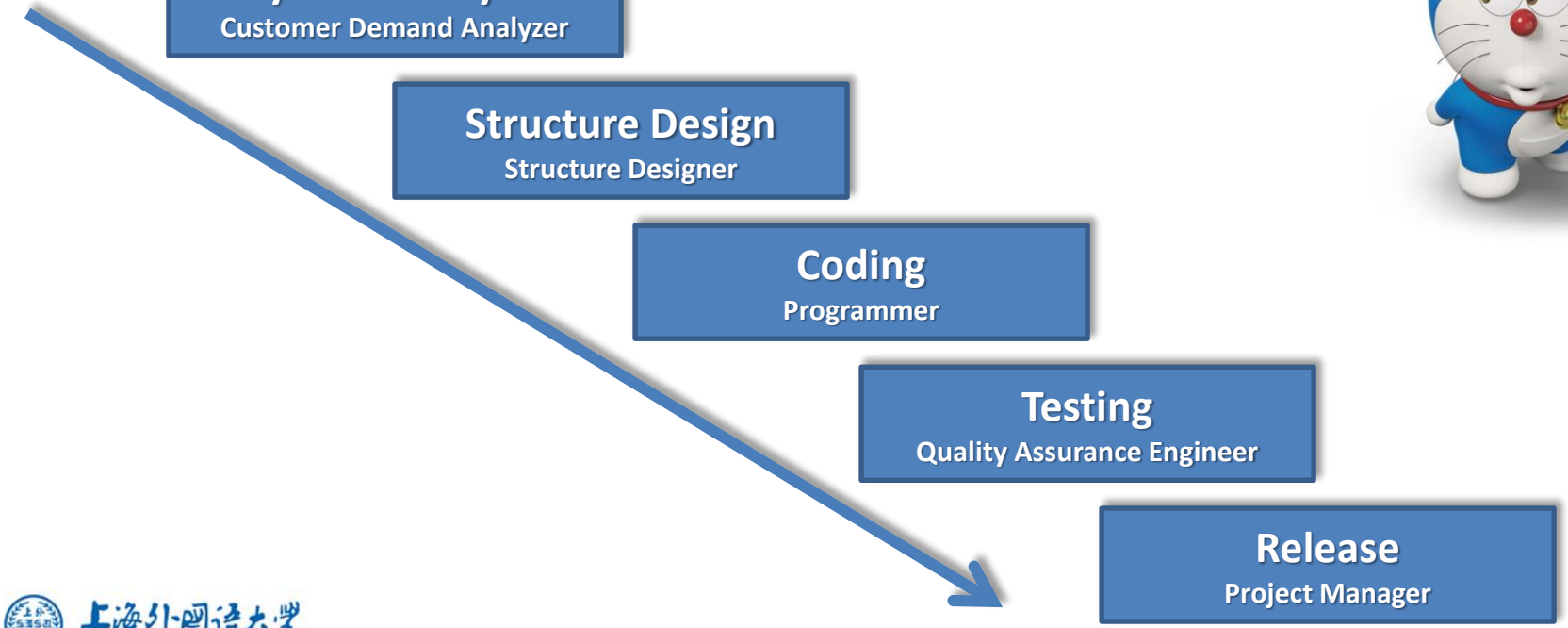
IEEE Spectrum

https://spectrum.ieee.org/at-work/innovation/the-2018-top-programming-languages



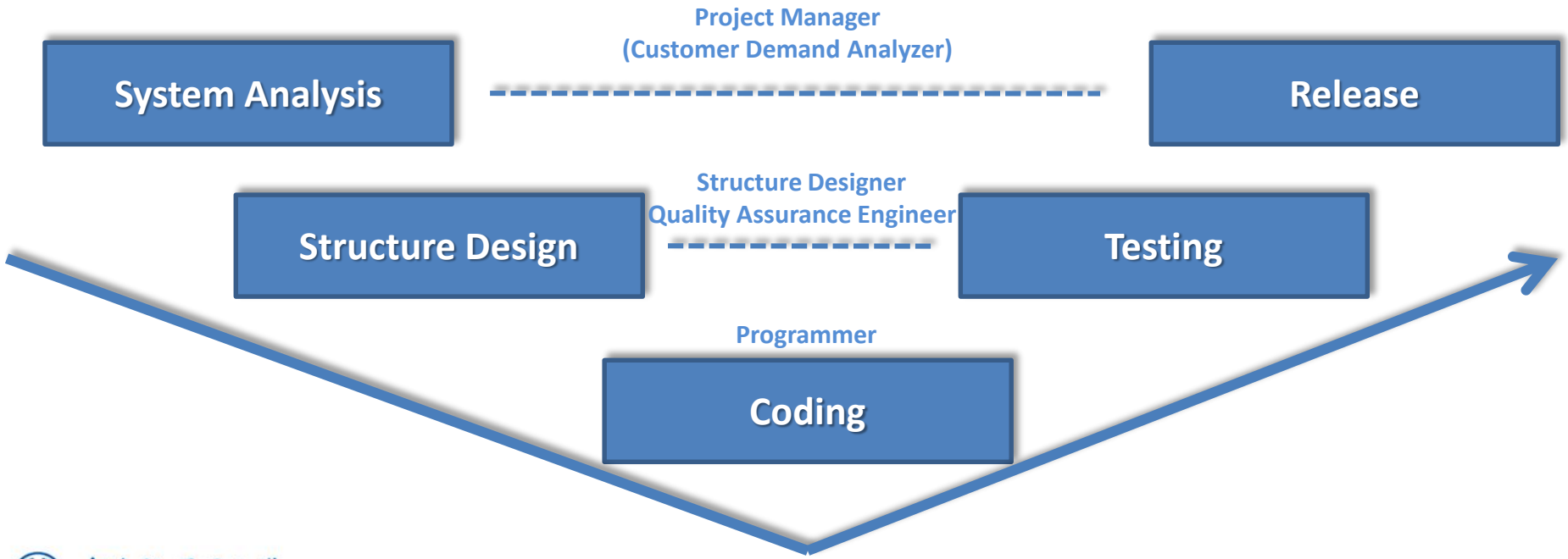
The Process of Software Development

- Water Fall Model



The Process of Software Development

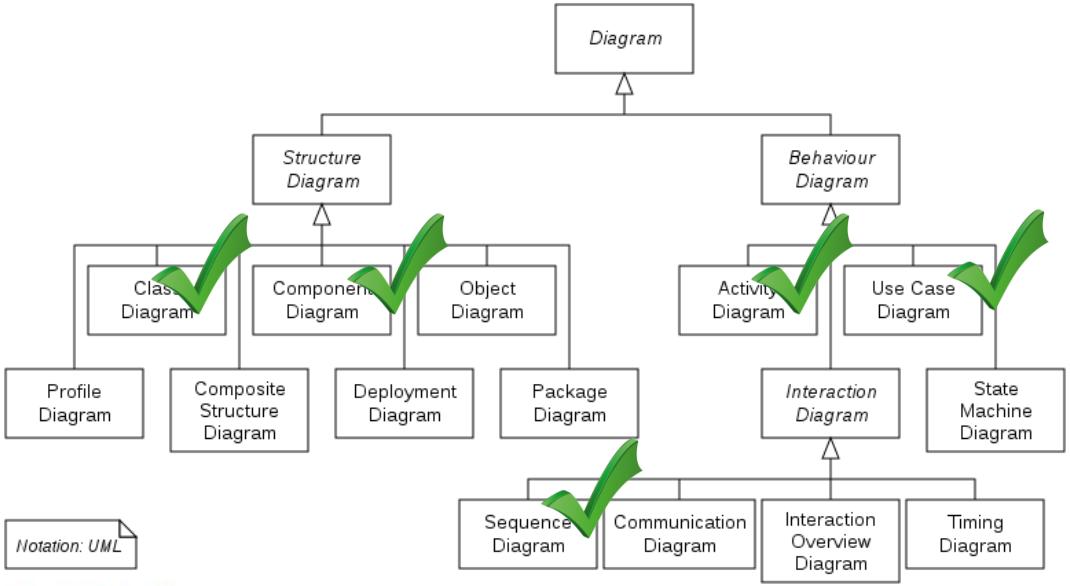
- V-Model



The Process of Software Development

Unified Modeling Language (UML)

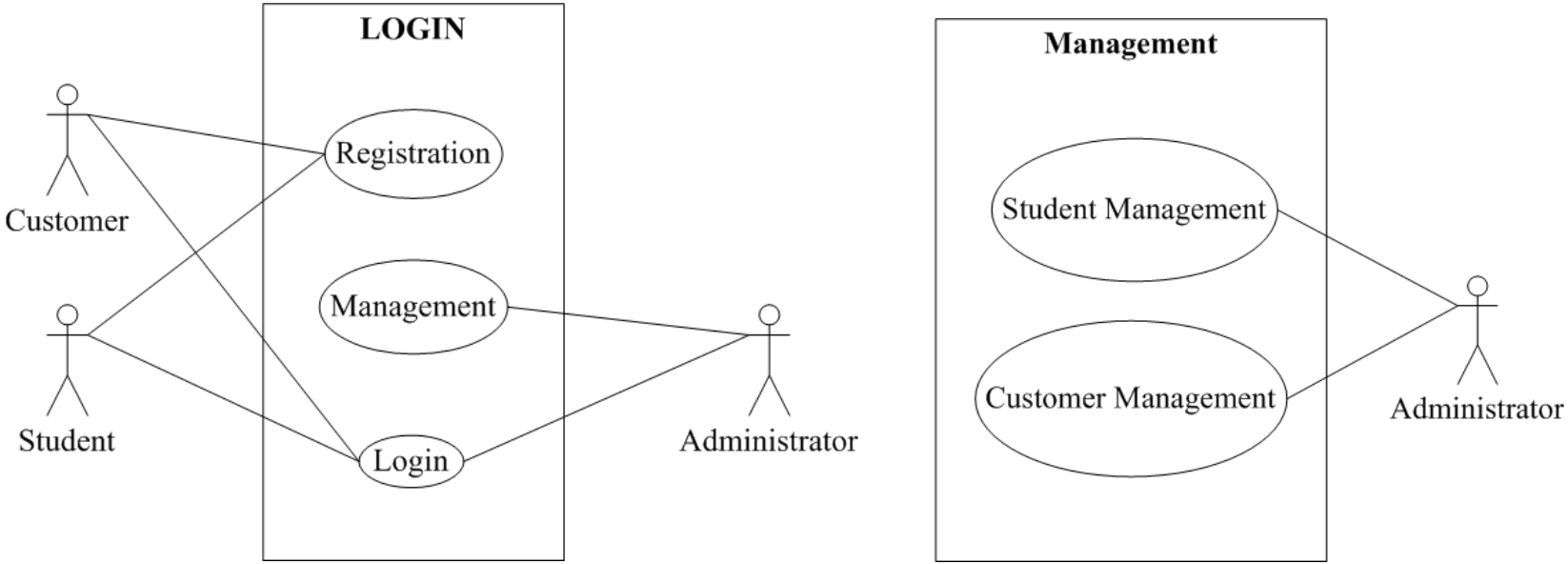
A general-purpose, developmental, modeling language in the field of software engineering, that is intended to provide a standard way to visualize the design of a system



UNIFIED MODELING LANGUAGE™

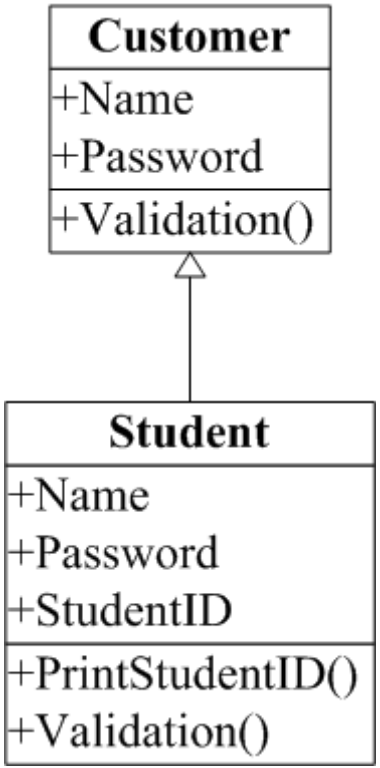
The Process of Software Development

- Use Case Diagram



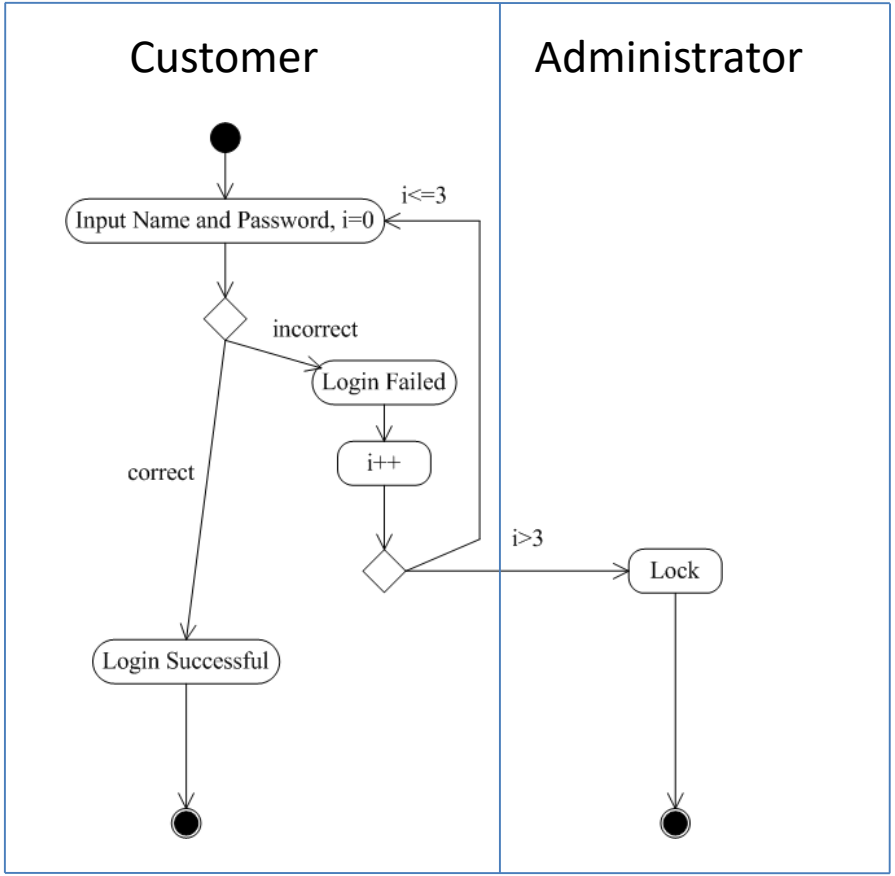
The Process of Software Development

- Class Diagram



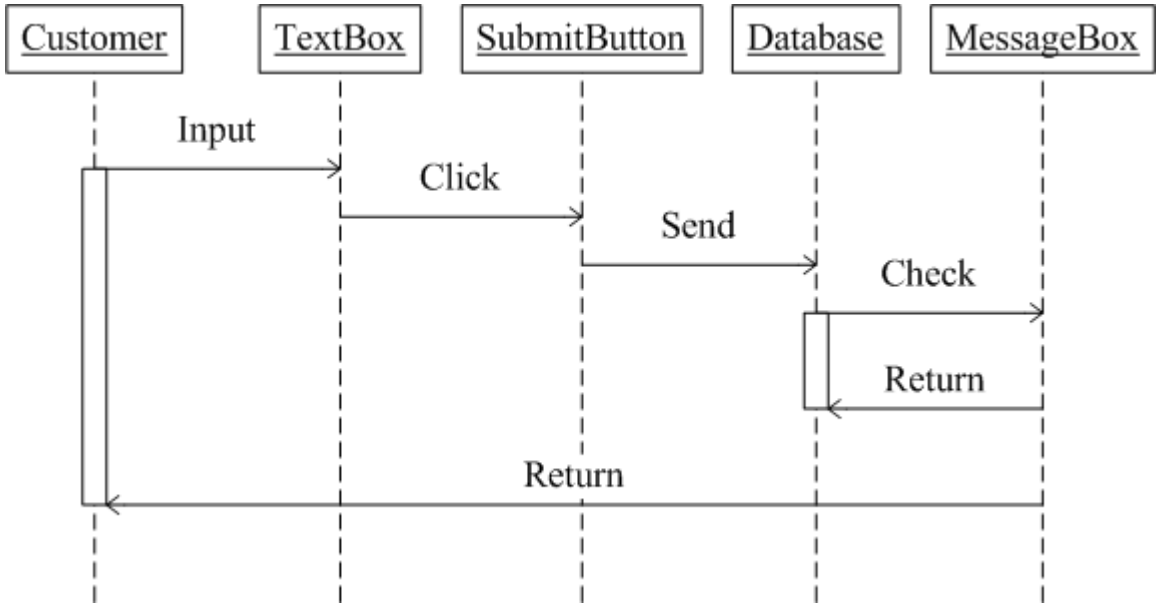
The Process of Software Development

- Activity Diagram



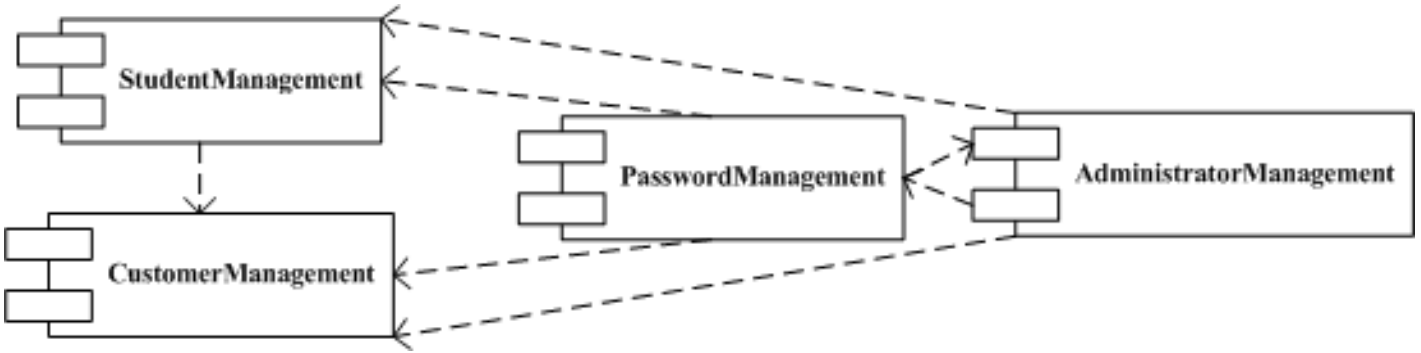
The Process of Software Development

- Sequence Diagram



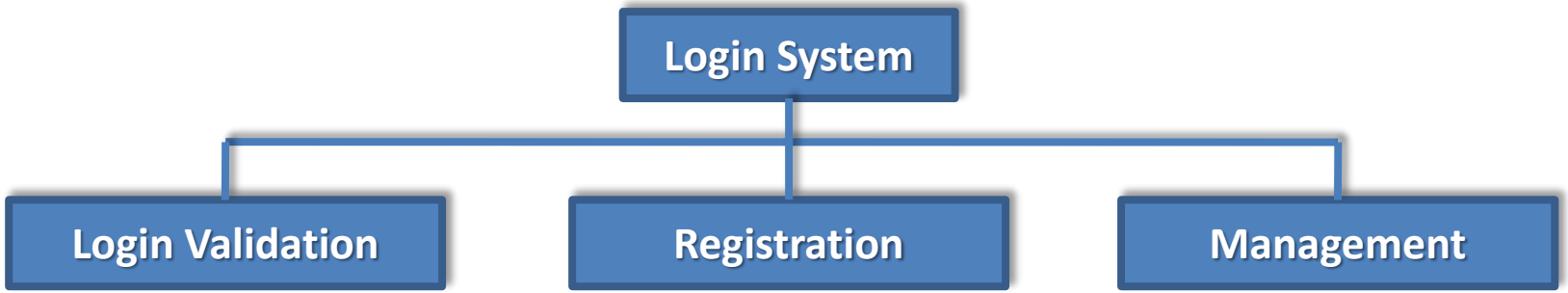
The Process of Software Development

- Component Diagram



The Process of Software Development

- Function Structure Diagram



The Process of Software Development

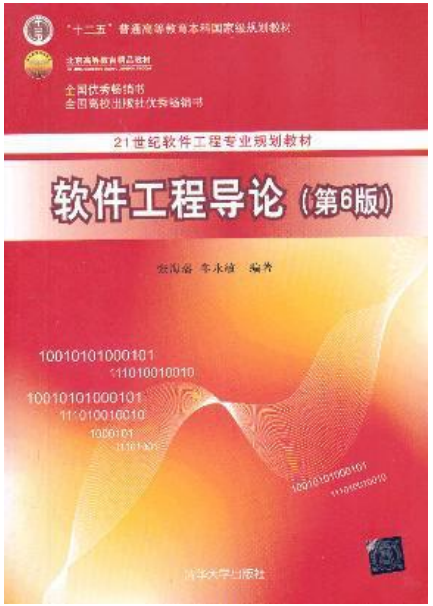
Time Estimation for Software Projects

- Man-Month
- Man-Day
- Basic Function: Insert, Delete, Update, Select
  - Slow: *1 Basic Function per day*
  - Common: *2 Basic Functions per day*
  - Fast: *4 Basic Functions per day*

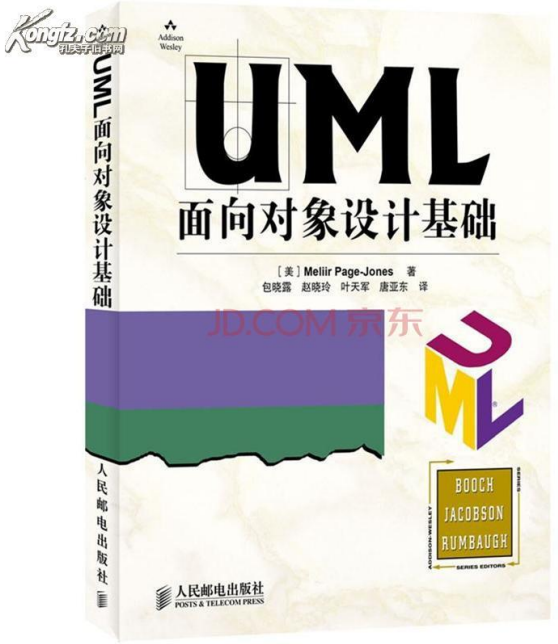
The Process of Software Development

References

软件工程导论（第6版）



UML面向对象设计基础



上海外国语大学
SHANGHAI INTERNATIONAL STUDIES UNIVERSITY

start my programming

Python Programming Environment

Outlines

- Installation
- Grammar
- Functions
- Debugging

上海外国语大学
SHANGHAI INTERNATIONAL STUDIES UNIVERSITY

to build an environment for running python

Installation

Installation

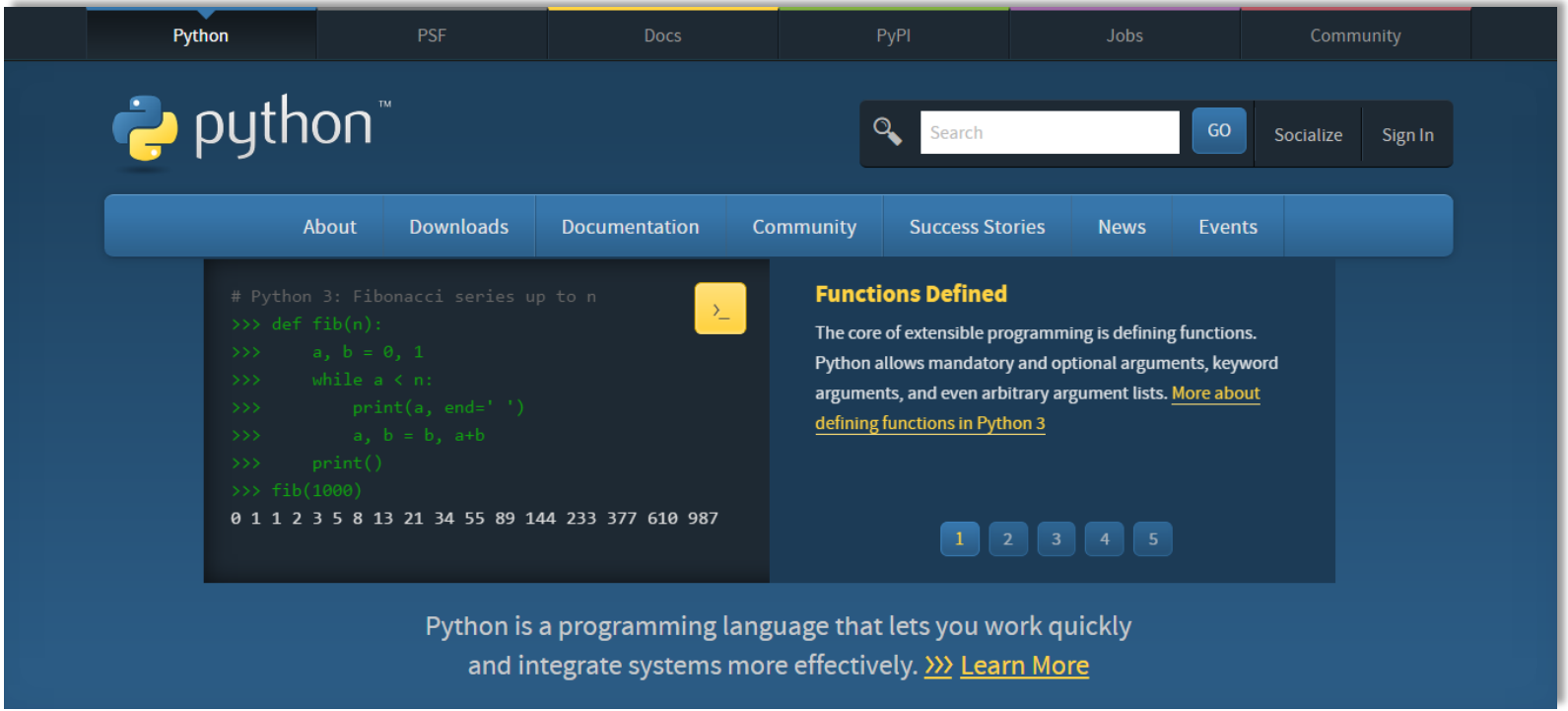
Guido van Rossum

The Birth of Python

Python is a widely used high-level, general-purpose, interpreted, dynamic programming language designed by Guido van Rossum in 1991.

Installation

Official Website of Python（https://www.python.org/）

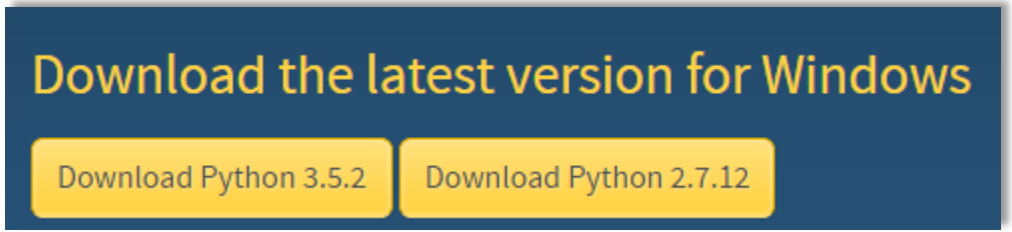


Python 2.X or 3.X?

Python 2.x is legacy, Python 3.x is the present and future of the language

Python 2.x is old, but mature;

Python 3.x is new, but slow.



https://wiki.python.org/moin/Python2orPython3

Download the Installation Package

https://www.python.org/downloads/

Python Releases for Windows

- Latest Python 2 Release - Python 2.7.12
- Latest Python 3 Release - Python 3.5.2

Python Releases for Mac OS X

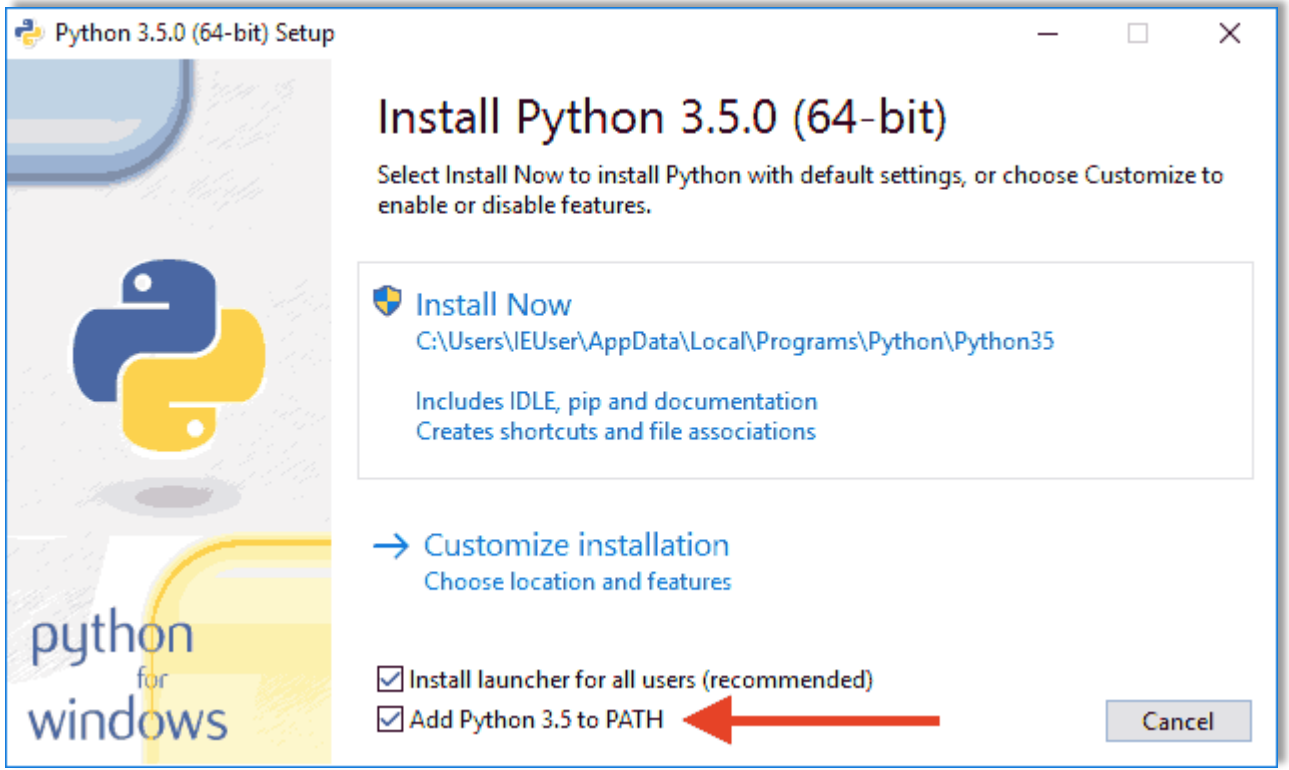
- Latest Python 2 Release - Python 2.7.12
- Latest Python 3 Release - Python 3.5.2

Python Source Releases

- Latest Python 2 Release - Python 2.7.12
- Latest Python 3 Release - Python 3.5.2

Installation

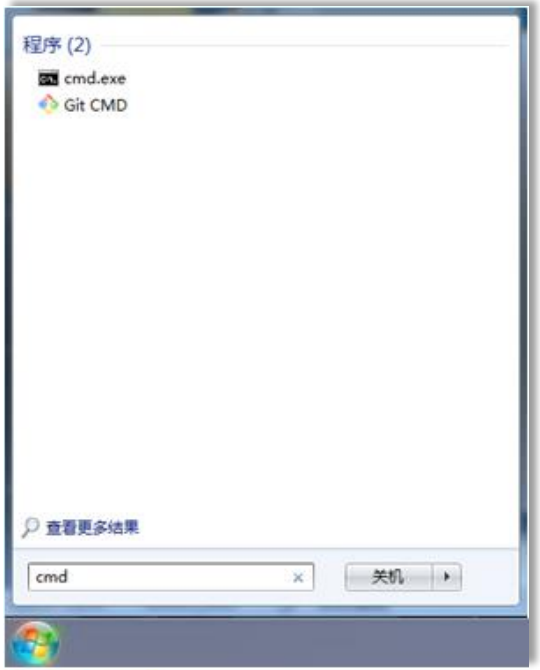
Next Step

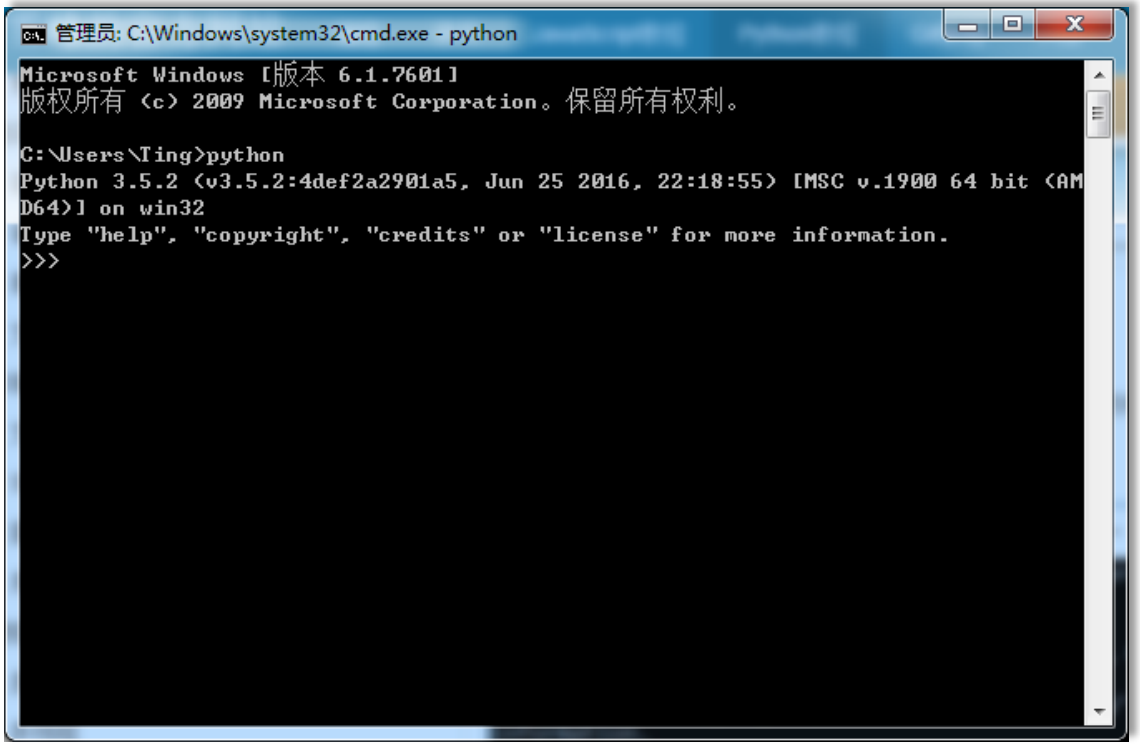


Installation

Success ！

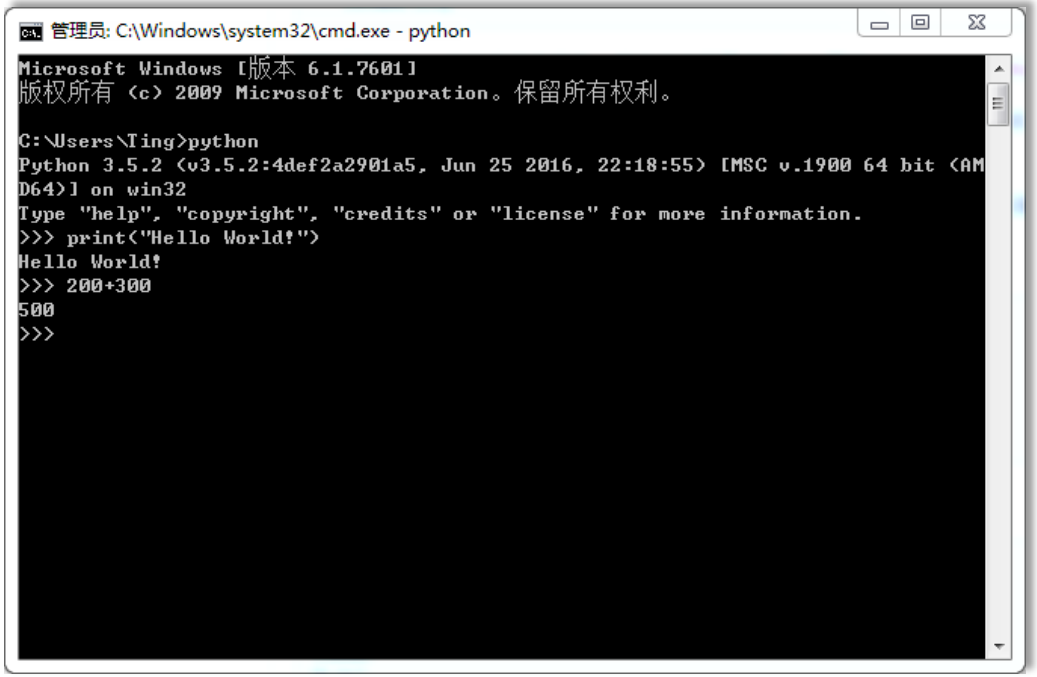
Start menu:





Installation

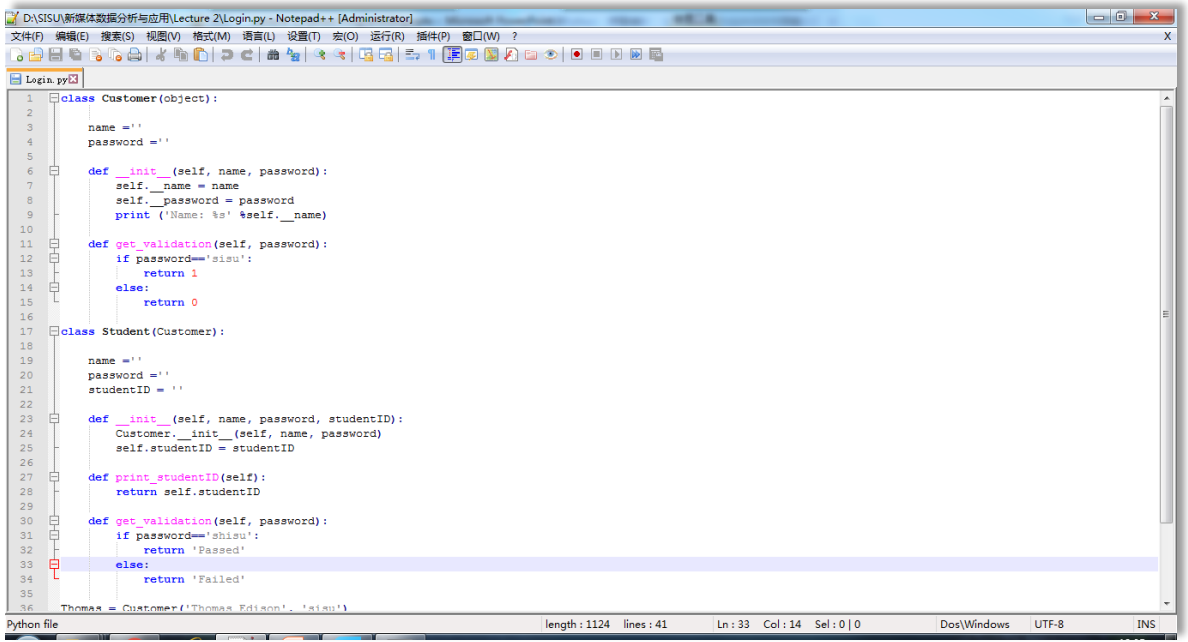
Example 1: Hello World, Python with CMD



IDE, Integrated Development Environment

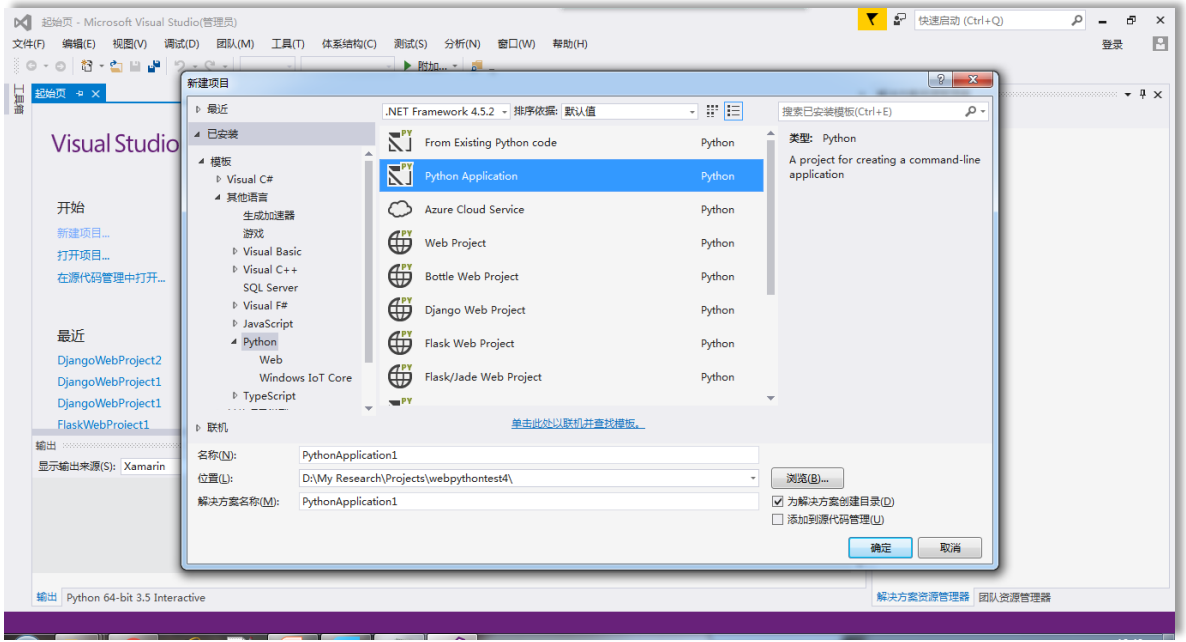
集成开发环境

1. Notepad++



IDE

2. Visual Studio 2015

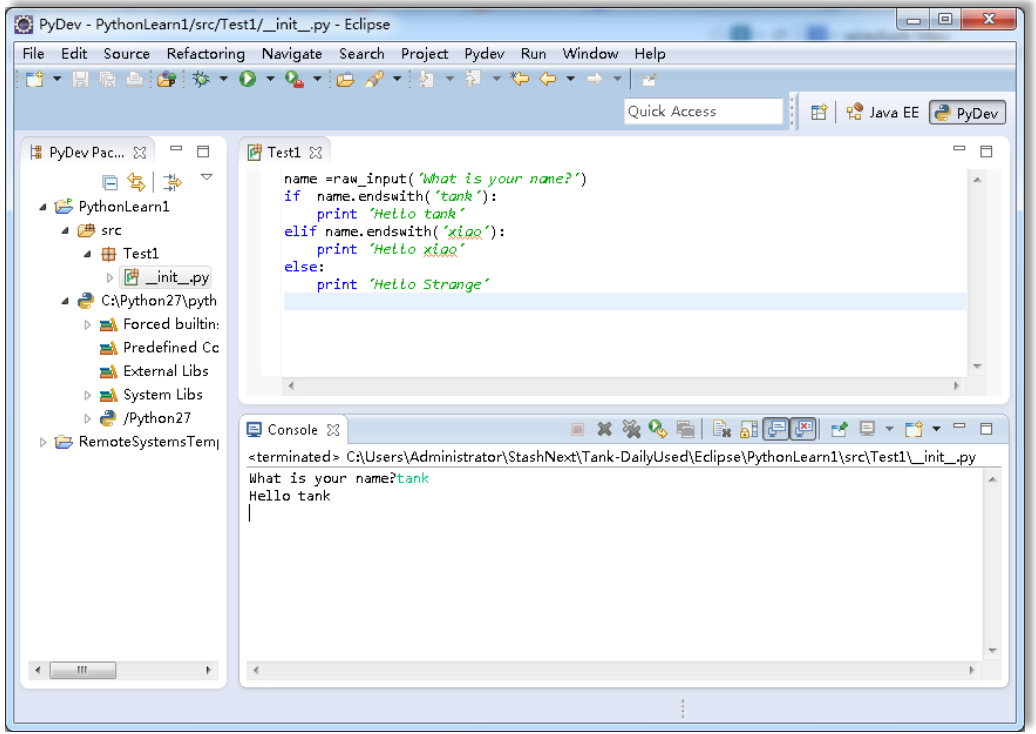


Installation

IDE

3. Eclipse + pydev

http://pydev.org/



http://www.cnblogs.com/Bonker/p/3584707.html

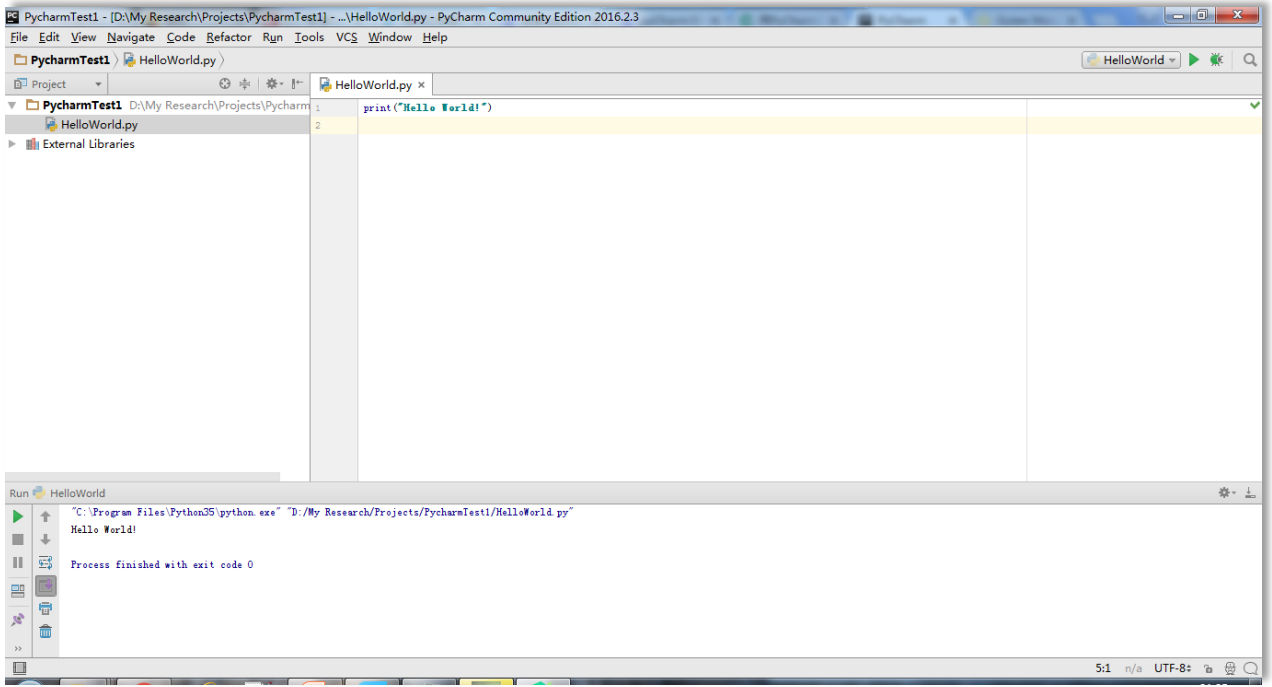
Installation

IDE

4. PyCharm

Professional Version

Community Version



http://www.jetbrains.com/pycharm/

Installation on Other Operation Systems

https://www.python.org/download/other/

- IBM AS/400 (OS/400)
- BeOS
- MS-DOS
- IBM OS/2
- IBM OS/390
- Series 60
- Oracle Solaris
- HP-UX

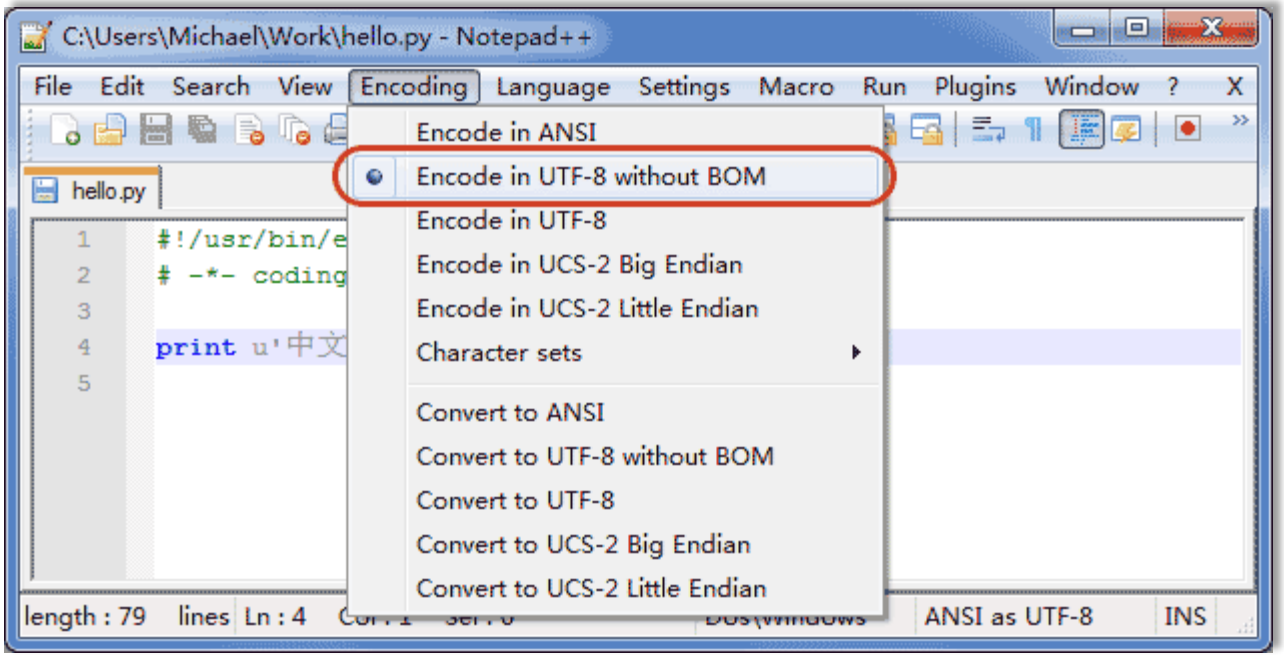
上海外国语大学
SHANGHAI INTERNATIONAL STUDIES UNIVERSITY

how to use python

Grammar

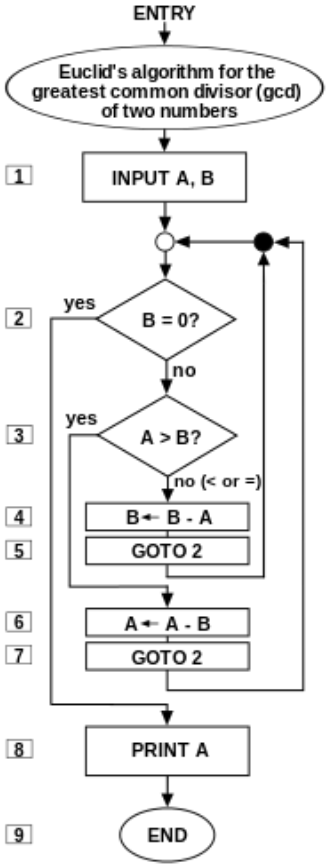
Character Encoding 字符编码

- Make sure that the encode is in UTF-8



Algorithm 算法

a self-contained step-by-step set of operations



Variables变量

a **storage location** paired with an associated symbolic **name** (an identifier), which contains some known or unknown quantity of information referred to as a **value**.

```
>>>x=2
>>>name="Thomas"
```

Case sensitive 大小写敏感 in python

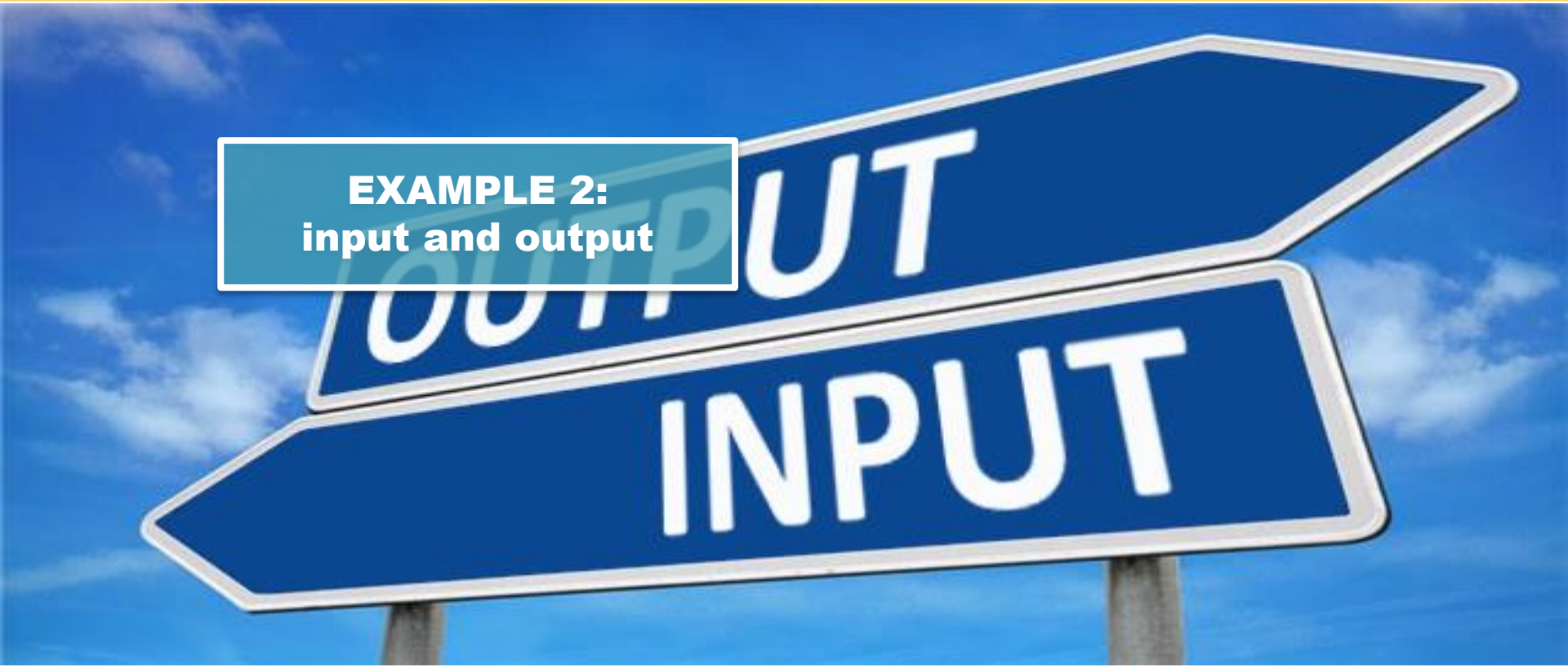
```
>>> x=2
>>> X=3
```

They are different variables!

Input and output 输入和输出

- input()
- print()

Grammar



Example 2: Hello 谁谁谁!

```
>>>name=input("What is your name?")
>>>print("Hello "+name+"!")
```

Data Structure of Variables 变量的数据结构

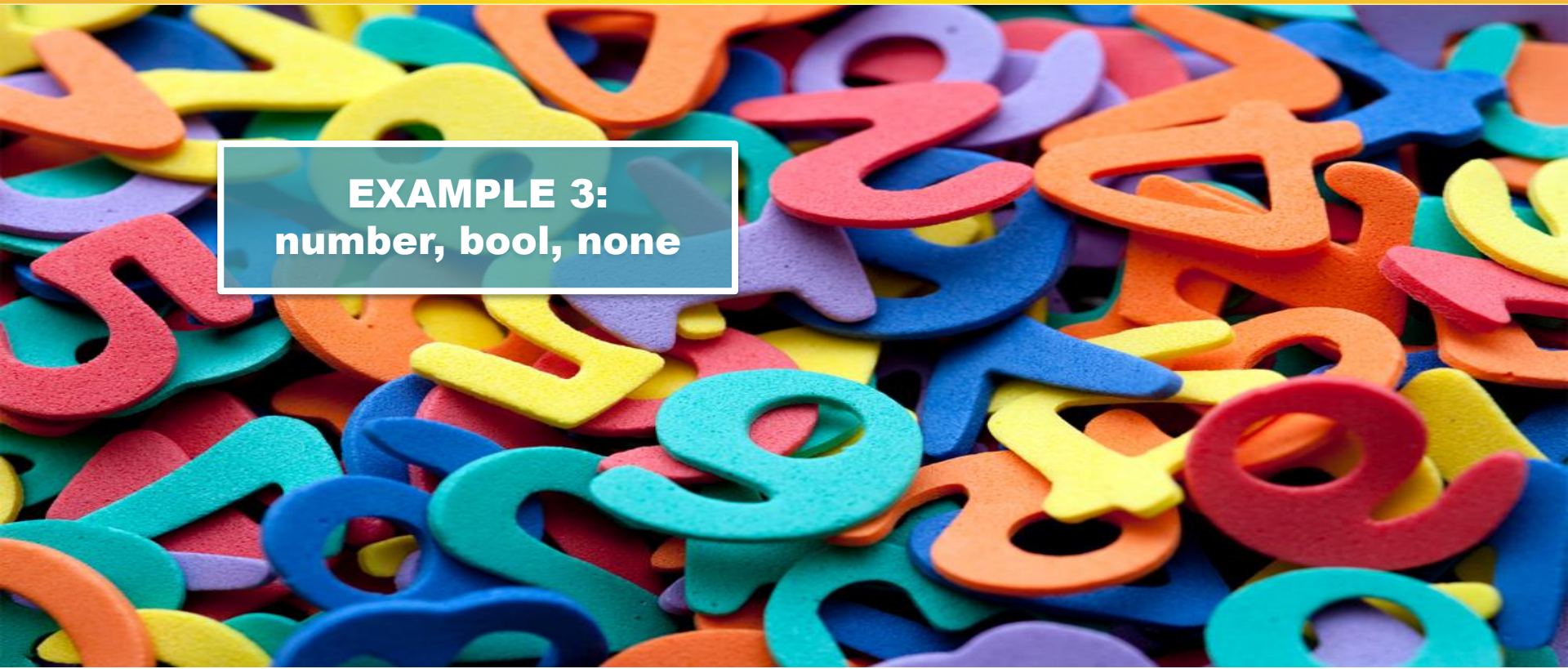
#	Type	#	Type
1	Number 数字	5	List 列表
2	Bool布尔逻辑	6	Tuple 元组
3	None 空值	7	Dict 字典
4	String 字符串	8	Datetime 日期

Reference: http://www.cnblogs.com/linjiqin/p/3608541.html

Number, bool, none 数字，布尔，空

Type_EN	Type_CN	Human_words	Example
integer	整型	整数	x=1
float	浮点型	小数	y=1.0
bool	布尔值 True, False, and, or, not	对错，与或非	print(x>y) print(x==y) print(not(x>y))
none	空值	二胎还没怀上，先把名字给取了，占位	x=None print(x)

Grammar



Operator Precedence

运算符	描述
lambda	Lambda表达式
or	布尔“或”
and	布尔“与”
not x	布尔“非”
in，not in	成员测试
is，is not	同一性测试
<，<=，>，>=，!=，==	比较
|	按位或
^	按位异或
&	按位与
<<，>>	移位
+，-	加法与减法

Low

Low High

High

运算符	描述
*，/，%	乘法、除法与取余
+x，-x	正负号
~x	按位翻转
**	指数
x.attribute	属性参考
x[index]	下标
x[index:index]	寻址段
f(arguments...)	函数调用
(experession,...)	绑定或元组显示
[expression,...]	列表显示
{key:datum,...}	字典显示
'expression,...'	字符串转换

String 字符串

– ***Convert***

强制转化

– ***ESC***

转义字符

- %d 整数
- %f 浮点数
- %s 字符串

```
def AddNumber(a, b):
    return int(a)+int(b)

def AddString(a, b):
    return a+b

InputOne=input("please input the first number: ")
InputTwo=input("please input the second number: ")
print("AddNumber: "+str(AddNumber(InputOne, InputTwo)))
print("AddString: "+AddString(InputOne, InputTwo))
```

```
( another style of Example 2):
>>>name=input("What is your name?")
>>>print("Hello, %s !" %name)
```

Grammar

Start from 0!

List列表，数组

```
>>>classmates = ['Michael', 'Bob', 'Tracy']
>>> classmates[0]
'Michael'
>>> classmates[1]
'Bob'
>>> classmates[2]
'Tracy'
>>> classmates[3]
Traceback (most recent call last): File "<stdin>", line 1, in <module>
IndexError: list index out of range
```

Use [ ]!

Tuple 元组

A list where values CANNOT be changed.

```
>>> classmates = ('Michael', 'Bob', 'Tracy')
```

Use ( )!

Grammar

dict 字典

d = {'key1':value1, 'key2':value2}

```
>>> d = {'Michael': 95, 'Bob': 75, 'Tracy': 85}
>>> d['Michael']
  95
```

Use {}!

set 集合

s = set([key1, key2, key3])

- Repetitions will be discarded
- No value, only key
- Actually, set is a function

```
>>> s = set([1, 1, 2, 2, 3, 3])
>>> s
 {1, 2, 3}
```

Datetime 时间

```
>>>import datetime
>>>print(datetime.datetime.now())
>>> dt = datetime.datetime (2018, 3, 5, 15, 30) # 用指定日期时间创建datetime
>>> print(dt)
            2018-03-05 15:30:00
```

Note:

1. “datetime” is a module. It should be imported before it is employed.
2. Python has many modules for different usages. Moreover, there are also a great number of **third-party modules**, which can be installed by Python command “pip”.

Conditional Statement 条件判断

```
if <condition 1>:
    <statement1>
elif < condition 2>:
    <statement2>
elif < condition 3>:
    <statement3>
else:
    <statement4>
```

Iteration 循环

for <counter> in <range>:
    <statement>

while <condition>:
    <statement>

break: stop the whole iteration

continue: stop this round, but continue to start the next round of this iteration

上海外国语大学
SHANGHAI INTERNATIONAL STUDIES UNIVERSITY

reuse and encapsulation

Functions

Functions

Define Functions

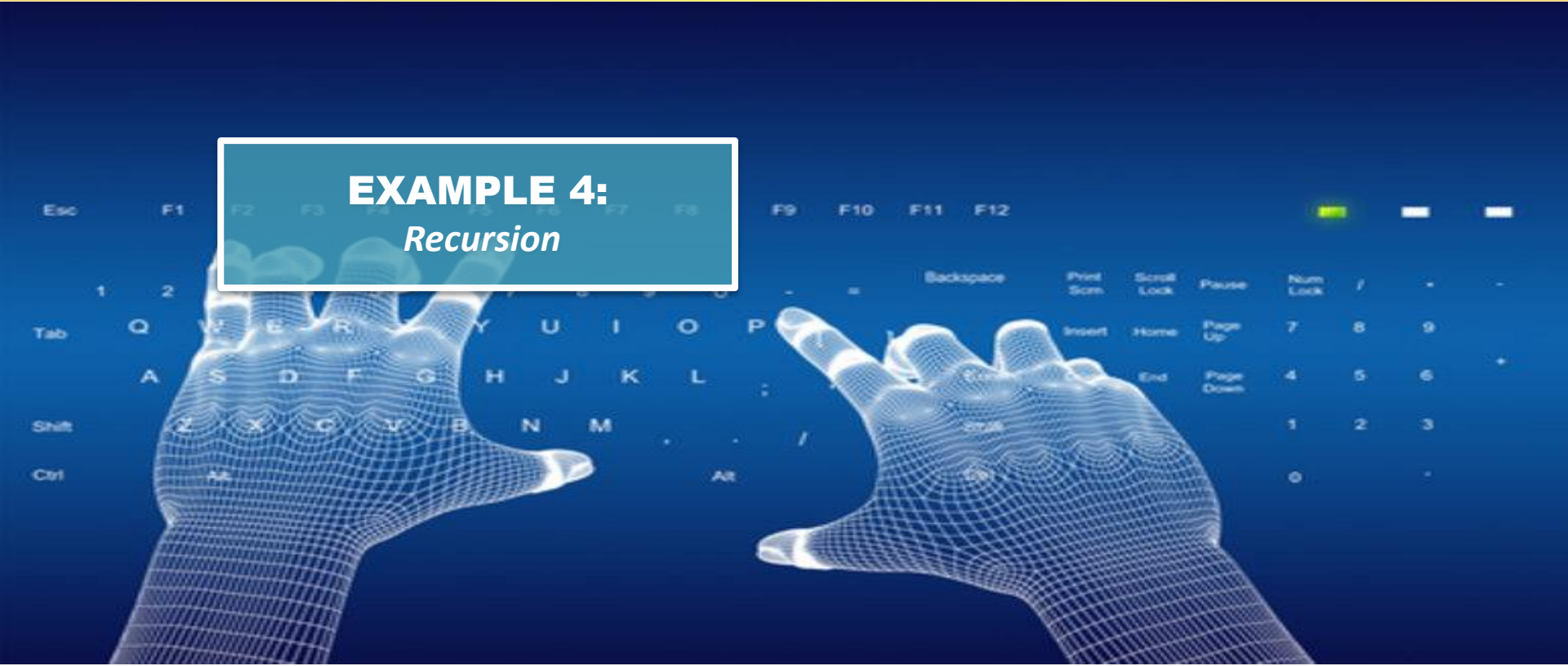
```
def FunctionName(parameter1, parameter2,…):
    <statement>
    [return value]
```

Optional

Call Functions

```
FunctionName(para1, para2,…)
```

Functions



Recursion: the function call itself 递归

Example: factorial 阶乘

n! = n*...*4*3*2*1

n! = n*(n-1)!

Assume that, F(n)=n!

Then F(n-1)=(n-1)!

∴ F(n)=n*F(n-1)

```
def factorial(n):
    if n==1:
        return 1
    else:
        return n*factorial(n-1)

number=input("Please input the number:")
print(factorial(int(number)))
```

Functions

Object Oriented Programming

- Class
- Object
- Attribute
- Method

上海外国语大学
SHANGHAI INTERNATIONAL STUDIES UNIVERSITY

testing, exception and modification

Debugging

Test-Driven Development

- STEP:
  1. print() it!
  2. Do NOT forget to delete print().

try...except...finally...

If we are not sure whether there are some errors in our code, we can use this statement.

Step 1: “try”

Step 2: Errors occur, stop “try”;

Step 3: go to “except”, and finish this part

Step 4: if there is a “finally” part then execute it;

Step 5: finish

```
try:
    print('try...')
    r = 10 / 0
    print('result:', r)
except ZeroDivisionError as e:
    print('except:', e)
finally:
    print('finally...')
print('END')
```

References

References

廖雪峰的官方网站（Python教程）

http://www.liaoxuefeng.com/wiki/0014316089557264a6b348958f449949df42a6d3a2e542c000

References

Microsoft Virtual Academy

https://mva.microsoft.com/zh-cn/training-courses/-python--8360?l=EK9zuOO8_2604984382

Home Work

Home Work

1. Print **all** the **Prime Numbers** smaller than **10,000**.

2. Print the first **30** numbers of ***Successione di Fibonacci***

You should do this homework by yourself and submit the report and the code to me individually before Oct. 31 via email attachment with the title "2018M + Your Chinese Name+ID"

上海外国语大学
SHANGHAI INTERNATIONAL STUDIES UNIVERSITY

The End of Lecture 2

Thank You

http://www.wangting.ac.cn