雨课堂
Rain Classroom

本次课程是

线上+线下

融合式教学

请**现场**的同学们：

1. 打开雨课堂，点击页面右下角喇叭按钮调至静音状态

请**远程上课**的同学们：

1. 打开雨课堂，点击页面右下角喇叭按钮调至静音状态
2. 打开“瞩目”（会议室：182 943 865；密码：见学堂公告），进入会议室，并关闭麦克风

Database Concepts (V)

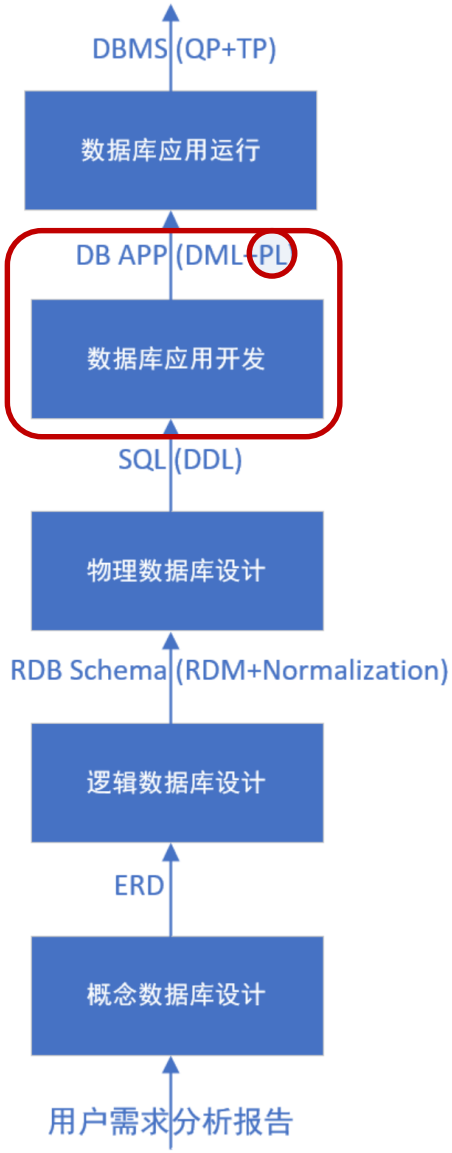
Database Connectivity

Chaokun Wang

School of Software, Tsinghua University

chaokun@tsinghua.edu.cn

May 2, 2022



Outline

- Python in a Nutshell

Introduction to Python

- Open source general-purpose language
- Object Oriented, Procedural, Functional
- Easy to interface with C/C++/ObjC/Java/Fortran
- Great interactive environment

Introduction to Python

- Environment
  - Python 2.x vs Python 3.x
  - Package managers and distributions
    - Anaconda/Pip/…
  - Recommend for Windows user
    - Install Anaconda2/Anaconda3
      - https://www.anaconda.com/download/ or
      - https://mirrors.tuna.tsinghua.edu.cn/anaconda/archive/
    - Default install path
      - C:\ProgramData\AnacondaX
        - » `C:\ProgramData\AnacondaX\Scripts`
          - conda install *packagename*
          - pip install *packagename*

First Python Program

- Interactive mode

```
(base) H:\Anaconda3>python
Python 3.7.3 (default, Mar 27 2019, 17:13:21) [MSC v.1915 64 bit (AMD64)] :: An
aconda, Inc. on win32
Type "help", "copyright", "credits" or "license" for more information.
>>> a = 'Hello World!'
>>> print(a)
Hello World!
>>>
```

- Script mode

```
(base) H:\Anaconda3>python D:\Desktop\hello.py
Hello World!
```

Basic Python Syntax

- Statements style

```
balance = 200
withdraws = 150
if withdraws % 100 == 0:
   if balance >= withdraws:
       print("Withdraws %d successfully, current balance is %d" % (withdraws, (balance - withdraws)))
else:
   print("Only notes in 100 yuan is available!")
```

- Multi-Line Statements

```
total = item_one + \
        item_two + \
        item_three

days = ['Monday', 'Tuesday', 'Wednesday',
        'Thursday', 'Friday']
```

- Quotation & comment

```
In [29]: # First comment
    ...: print("Hello, 'Python'!") # second comment
    ...: '''This is a multi line comment
    ...: print("You can't see me!")
    ...: In fact it is a paragraph'''
    ...: print('Have fun!')
    ...:
Hello, 'Python'!
Have fun!
```

Basic Python Syntax

- Operators
  - +- * / // % **
  - < <= > >= == !=
  - and or not

```
>>> a = 5
>>> -2 * 4 + a ** 2
17
>>> a / 2
2.5
>>> a / 2.0
2.5
>>> a // 2.0
2.0
>>> 77 > 66 == 66  # same as (77 > 66) and (66 == 66)
True
>>> a == a / 2 * 2 + a % 2
False
>>> a == a // 2 * 2 + a % 2
True
>>> a = 'Hello' + " " + 'World!'
>>> print(a)
Hello World!
```

Python Data Types

- Numbers and assignment

```
In [1]: a = 1       # An integer assignment
        b = 2
        c = 0x1A    # A hex integer
        d = 1.0     # A floating point
        e = 1 + 2j  # A complex number
        f = 1 - 2j
```

```
In [2]: c
Out[2]: 26
```

```
In [3]: d + e
Out[3]: (2+2j)
```

```
In [4]: e * f
Out[4]: (5+0j)
```

```
In [5]: for k in range(1, 6):
            a *= k
        a
Out[5]: 120
```

```
In [6]: for k in range(1, 200):
            a *= k
        a
Out[6]: 473194720418874302131417928359311037377081586612303957976
84551994661566997804200575218257040719077434250632728335l
78329815458567414558226303715597497169795866807771436245 7
99034243819336635346737287099854415241997511847873645697 6
91196651454544454434194419275444334774119613561914250219 4
43741102051527770735333367377422233842483200000000000000 0
00000000000000000000000000000000000
```

```
In [7]: x, y = b, c
        m = n = x
        m
Out[7]: 2
```

```
In [8]: del(m)
```

```
In [9]: m
```

```
NameError                                 Traceback (mos
t recent call last)
<ipython-input-9-9a40b379906c> in <module>
----> 1 m

NameError: name 'm' is not defined
```

Python Data Types

- String

In [10]:
```
string = 'Hello World'
print(string)              # Prints complete string
print(string[0])           # Prints first character of the string
print(string[1:5])         # Prints characters starting from 2rd to 5th
print(string[:5])          # Prints string starting from start to 5th character
print(string[6:])          # Prints string starting from 3rd character
print(string[-5:])         # Prints string starting from 3rd charcater
print((string + " ") * 3)  # Prints strting (concatenated with a space) three times
```

```
Hello World
H
ello
Hello
World
World
Hello World Hello World Hello World
```

In [12]:
```
c = 0xDA
dec = "decimal"
hex = "hexadecimal"
print("%x in %s equal to %d in %s" % (c , hex, c, dec))
print("%.2f" % 12.34567)
print("%8.2f" % 12.34567)
```

```
da in hexadecimal equal to 218 in decimal
12.35
   12.35
```

```
>>> str = 'Hello python, hello world'
>>> str.upper()
'HELLO PYTHON, HELLO WORLD'
>>> str.endswith("world")
True
>>> str.endswith("World")
False
>>> str.find("world")
20
>>> ".".join(str.split())
'Hello.python,.hello.world'
```

Python Data Types

- List

```
In [14]: alist = ['abcd', 786, 1 + 3j, 'mary']
         print(alist[2])
         print(alist[2:4])

         (1+3j)
         [(1+3j), 'mary']
```

```
In [15]: blist = [123, 'john']
         alist + blist * 2

Out[15]: ['abcd', 786, (1+3j), 'mary', 123, 'john', 123, 'john']
```

```
In [16]: alist.append('john')
         alist

Out[16]: ['abcd', 786, (1+3j), 'mary', 'john']
```

```
In [17]: alist[3] = blist
         alist

Out[17]: ['abcd', 786, (1+3j), [123, 'john'], 'john']
```

```
In [18]: del(alist[3])
         alist

Out[18]: ['abcd', 786, (1+3j), 'john']
```

```
In [19]: alist.remove(alist[3])
         alist

Out[19]: ['abcd', 786, (1+3j)]
```

```
In [20]: alist.pop()

Out[20]: (1+3j)
```

```
In [21]: alist

Out[21]: ['abcd', 786]
```

Python Data Types

- Tuple
  - Tuples can be thought of as **read-only** lists

```
In [23]: atup = ('abcd', 786, 1 + 3j)
         print(atup[1])
         print(atup[1:])

         786
         (786, (1+3j))
```

```
In [25]: btup = ([1, 2], 3)
         atup[3] = btup

         ---------------------------------------------------------------------------
         TypeError                                 Traceback (most recent call last)
         <ipython-input-25-21b0e3f2531b> in <module>
               1 btup = ([1, 2], 3)
         ----> 2 atup[3] = btup

         TypeError: 'tuple' object does not support item assignment
```

```
In [26]: atup = atup[0:2] + (20.4, ) + atup[3:]
         atup

Out[26]: ('abcd', 786, 20.4)
```

```
In [27]: atup + ('john', )

Out[27]: ('abcd', 786, 20.4, 'john')
```

```
In [28]: btup = ([1, 2], 3)
         btup[0][1] = 4
         btup

Out[28]: ([1, 4], 3)
```

Python Data Types

- Dictionary

In [29]:
```
dict = {}
dict['one'] = "This is one"
dict[2] = "This is two"
print(dict)
print(dict["one"])  # Prints value for 'one' key
print(dict[2])      # Prints value for 2 key

dict[2] = "A new two"
print(dict)
```

```
{'one': 'This is one', 2: 'This is two'}
This is one
This is two
{'one': 'This is one', 2: 'A new two'}
```

In [30]:
```
tinydict = {'name': 'john', 'code': 6734, 'dept': 'sales'}
print(list(tinydict.keys()))    # Prints all the keys
print(list(tinydict.values()))  # Prints all the values
print(list(tinydict.items()))   # Prints all the keys and values
```

```
['name', 'code', 'dept']
['john', 6734, 'sales']
[('name', 'john'), ('code', 6734), ('dept', 'sales')]
```

Python Data Types

- Date & Time

```
In [1]: from datetime import date

In [2]: today = date.today()
        today
Out[2]: datetime.date(2021, 5, 6)

In [3]: my_birthday = date(today.year, 1, 24)
        if my_birthday < today:
            my_birthday = my_birthday.replace(year=today.year + 1)
        my_birthday
Out[3]: datetime.date(2022, 1, 24)

In [4]: from dateutil.relativedelta import relativedelta

In [5]: today + relativedelta(days=21)
Out[5]:

In [6]: later = today + relativedelta(days=21) + relativedelta(months=3)
        later
Out[6]:

In [7]: relativedelta(later, today)
Out[7]:

In [8]: (later - today).days
Out[8]:
```

Python Data Types

- Date & Time

```
In [9]: from datetime import datetime

In [10]: datetime.now()
Out[10]: datetime.datetime(2021, 5, 6, 11, 34, 29, 191925)

In [11]: datetime.utcnow()
Out[11]: datetime.datetime(2021, 5, 6, 3, 34, 29, 624446)

In [12]: now = datetime.now()

In [13]: now.date()
Out[13]: datetime.date(2021, 5, 6)

In [14]: now.time()
Out[14]: datetime.time(11, 34, 30, 227380)

In [15]: now.minute
Out[15]: 34
```

Python Statements

- Conditions
  - if… elif… else…

```
if user.cmd== 'create':
  action = "create item"
elif user.cmd == 'delete':
  action = "delete item"
elif user.cmd == 'update':
  action = "update item"
else:
  action = "invalid command, try again!"
```

```
if user.cmd in ('create', 'delete', 'update'):
  action = "%s item" % user.cmd
else:
  action = "invalid command, try again!"
```

  - Ternary conditional operator

```
In  [48]: x, y = 4, 3
          smaller = x if x < y else y
          smaller

Out[48]: 3
```

Python Statements

- Loops
  - while

```
In [49]: p = k = 1
         while k <= 10:
             p *= k
             k += 1
         print(p)

3628800
```

  - for

```
In [50]: ages = {'john': 26, 'mary': 18, 'david': 27}
         for name in ages:
             print("%s's age is %d" % (name, ages[name]))

john's age is 26
mary's age is 18
david's age is 27
```

```
In [51]: list(range(2, 19, 3))

Out[51]: [2, 5, 8, 11, 14, 17]
```

```
In [52]: p = 1
         for k in range(1, 10):
             p *= k
         print(p)

362880
```

Python Statements

- break & continue

In [54]:
```
passwdList = ["one", "two", "three"]
valid = False
count = 3
while count > 0:
    input_passwd = input("enter password: ")
    for passwd in passwdList:
        if input_passwd == passwd:
            valid = True
            print("Welcome!")
            break
    if valid == False:
        print("invalid password")
        count -= 1
        continue
    else:
        break
```

```
enter password: four
invalid password
enter password: two
Welcome!
```

Python Statements

- List comprehensions

```
In [55]: [x ** 2 for x in range(6)]
Out[55]: [0, 1, 4, 9, 16, 25]

In [56]: import random
         seq = [random.randint(0, 1000) for x in range(0, 8)]
         seq
Out[56]: [191, 453, 618, 384, 280, 734, 564, 54]

In [57]: [x for x in seq if x % 2]
Out[57]: [191, 453]

In [59]: [(x + 1, y + 1, z + 1) for x in range(2) for y in range(2) for z in range(2)]
Out[59]: [(1, 1, 1),
          (1, 1, 2),
          (1, 2, 1),
          (1, 2, 2),
          (2, 1, 1),
          (2, 1, 2),
          (2, 2, 1),
          (2, 2, 2)]
```

Python Functions

```
In [60]: def printInfo(name, age=35):
             "This prints a passed info into this function"
             print("%s's age is: %d" % (name, age))
```

```
In [61]: printInfo("miki", 50)
         printInfo(age=20, name="mark")
         printInfo("john")
```

```
miki's age is: 50
mark's age is: 20
john's age is: 35
```

```
In [62]: def factorial(x):
             return x * factorial(x - 1) if x >= 1 else 1
```

```
In [63]: factorial(10)
```

```
Out[63]: 3628800
```

```
In [66]: total = 0  # This is a global variable.
         def sum(arg1, arg2):
             total = arg1 + arg2  # Here total is a local variable.
             print("Inside the function local total:", total)
         sum(10, 20)
         print("Outside the function global total:", total)
```

```
Inside the function local total: 30
Outside the function global total: 0
```

Python Encoding & Decoding

- In python3, the encoding is always “utf8”.

```
In [67]: import sys
         sys.stdout.encoding
Out[67]: 'UTF-8'

In [68]: "你好"
Out[68]: '你好'
```

- However, you can encode into other encoding.

```
In [69]: "你好".encode()
Out[69]: b'\xe4\xbd\xa0\xe5\xa5\xbd'

In [70]: "你好".encode("gbk")
Out[70]: b'\xc4\xe3\xba\xc3'
```

- Or decode after encoding.

```
In [71]: print("你好".encode().decode())
         print("你好".encode('gbk').decode('gbk'))
         print("你好".encode().decode('gbk'))
你好
你好
浣犲ソ
```

Python Modules

- import
  - import random
  - from datetime import date, time, datetime
- Installed modules
  - C:\ProgramData\AnacondaX\Lib
  - C:\ProgramData\AnacondaX\Lib\site-packages

run.py

```
1
2  def run():
3      global run
4      del run
5
6      from win_unicode_console import runner
7
8      runner.run_arguments()
9
10 if __name__ == "__main__":
11     run()
```

Conclusions

- Python in a Nutshell
  - Introduction to Python
  - First Python Program
  - Basic Python Syntax
  - Python Data Types
  - Python Statements
  - Python Functions
  - Python Encoding & Decoding
  - Python Modules

References

- Python核心编程, Wesley Chun (著), 宋吉广 (译). 人民邮电出版社, 2008
- Python Tutorial: https://www.tutorialspoint.com/python/python_overview.htm

Thank you!