



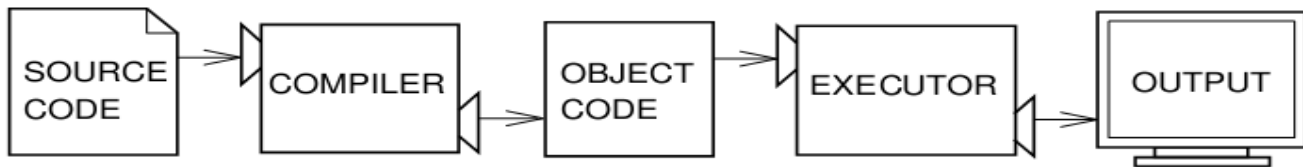
Python vs. C

车万翔

哈尔滨工业大学



❖ 编译型语言 (C/C++ 等)



❖ 解释型语言 (BASIC、Python等)





Python 语言介绍



哈爾濱工業大學
HARBIN INSTITUTE OF TECHNOLOGY

- ❖ 诞生于1989年，英国发音： $/\text{'paɪ}\theta\text{ə}n/$ ，美国发音： $/\text{'paɪ}\theta\text{ɑ:n}/$
- ❖ 创始人吉多·范罗苏姆（Guido van Rossum）





Python 语言的特点



哈爾濱工業大學
HARBIN INSTITUTE OF TECHNOLOGY

- ❖ 解释型语言
- ❖ 设计哲学是“优雅”、“明确”、“简单”
 - 易学、易用
 - 可读性高
- ❖ 开发哲学是“用一种方法，最好是只用一种方法来做一件事”
- ❖ 现代编程语言
 - 面向对象
 - 支持泛型设计
 - 支持函数式编程
- ❖ 丰富的数据结构和第三方函数库
 - 功能强大



Atlassian



Bitbucket





Hello, World!



哈爾濱工業大學
HARBIN INSTITUTE OF TECHNOLOGY

❖ 它是编程之神的传统咒语，可以帮助你开始这段感情.....

```
# 打印Hello, World!到屏幕
```

注释

```
print ('Hello, World!')
```

执行语句



Hello World!



哈爾濱工業大學
HARBIN INSTITUTE OF TECHNOLOGY

Python

```
print ('Hello World! ')
```

```
$ python hello.py
```

C

```
#include <stdio.h>
```

```
int main()
```

```
{
```

```
    printf("Hello World!\n");
```

```
    return 0;
```

```
}
```

```
$ gcc hello.c -o hello
```

```
./hello
```



两种编程方式：控制台和脚本



哈爾濱工業大學
HARBIN INSTITUTE OF TECHNOLOGY

控制台

❖ 优点

- 无需创建文件
- 立即看到运行结果

❖ 适用场景

- 语句功能测试

```
Wanxiangs-iMac:~ wanxiang$ python3
Python 3.6.1 (v3.6.1:69c0db5050, Mar 21 2017
[GCC 4.2.1 (Apple Inc. build 5666) (dot 3)]
Type "help", "copyright", "credits" or "lice
>>> print('Hello World')
Hello World
>>> 
```

脚本

❖ 优点

- 反复运行
- 易于编辑

❖ 适用场景

- 编写大型程序

```
hello.py x
1 ▶ print 'Hello World'
2
```



Python

- ❖ 行尾无分号
- ❖ 严格的缩进规范
 - 缩进表示程序块
- ❖ 没有++、--运算符
- ❖ 逻辑运算符
 - and, or, not
- ❖ 支持串联比较运算
 - `3 < 5 < 7 : True`
 - `3 < 5 < 2 : False`

C

- ❖ 行尾必须有分号
- ❖ 缩进任意
 - {} 表示程序块
- ❖ 没有**运算符
- ❖ 逻辑运算符
 - &&, ||, ~
- ❖ 不支持串联比较运算
 - `3 < 5 < 7 : True`
 - `3 < 5 < 2 : True`



Python

- ❖ if condition:
 pass
- ❖ for i in range(10):
- ❖ while condition:
 pass
- ❖ 没有 do while
- ❖ while 和 for 可以跟 else
- ❖ 注释
 - # ...

C

- ❖ if(condition){ }
- ❖ for (i = 0; i < 10; i++){ }
- ❖ else if 等价于 Python 的 elif
- ❖ 注释
 - // ...
 - /* ... */



语法层面 (3)



Python

```
a = input('Input a char: ')
if a == 'a':
    print 'you pressed a'
elif a == 'b':
    print 'you pressed b'
elif a == 'c':
    print 'you pressed c'
else:
    print 'not a, b, c'
```

C

```
char c;
printf( "Input a char:" );
scanf( "%c", &c );

switch( c )
{
    case 'a':
        printf(" you pressed a ");
        break;
    case 'b':
        printf(" you pressed b ");
        break;
    case 'c':
        printf(" you pressed c ");
        break;
    default:
        printf(" not a, b, c ");
        break;
}
```



Input and Output



哈爾濱工業大學
HARBIN INSTITUTE OF TECHNOLOGY

Python

❖ Input

- `a = int(input('Input something'))`
- 输入的是字符串
- 再转化为相应的类型

❖ Output

- `print a`
- 输出 a 的值和回车

C

❖ Input

- `int a;`
- `scanf("%d", &a);`

❖ Output

- `printf("%d\n", a);`



Dynamic vs. Static Typing



哈爾濱工業大學
HARBIN INSTITUTE OF TECHNOLOGY

Python

```
>>> a = 123
>>> type(a)
<type 'int'>
>>> a = 'abc'
>>> type(a)
<type 'str'>
```

变量是一个对象的别名、指针或引用，
赋值是重新绑定

C

```
int a;
a = 123; ✓
a = "abc"; ✗
```

变量是数据的内存地址名，赋值是改写
内容

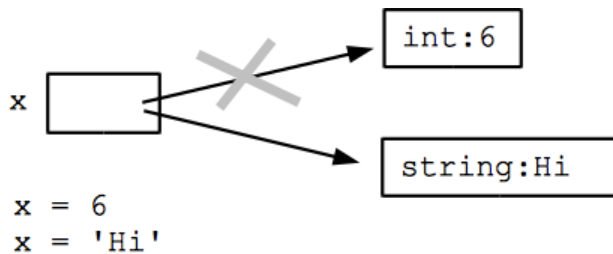


Dynamic vs. Static Typing



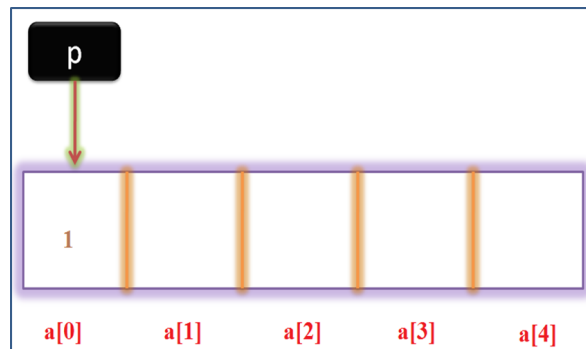
哈爾濱工業大學
HARBIN INSTITUTE OF TECHNOLOGY

Python



Object Store

C



Address Resolving



Dynamic vs. Static Typing



哈爾濱工業大學
HARBIN INSTITUTE OF TECHNOLOGY

Python

```
n = 1  
print(id(n))  
n += 1  
print(id(n))
```

C

```
int a = 1;  
printf( "0x%x\n", &a);  
a += 1;  
printf( "0x%x\n", &a);
```

数值类型不可变



Dynamic vs. Static Typing



Python

```
# a and b can be any type
def max(a, b):
    if a > b:
        return a
    else:
        return b
```

C

```
// a and b can only be int
int max(int a, int b)
{
    if (a > b)
        return a;
    else
        return b;
}
```



Dynamic vs. Static Typing



哈爾濱工業大學
HARBIN INSTITUTE OF TECHNOLOGY

$$\pi = 4 \times (1 - 1/3 + 1/5 - 1/7 \dots)$$

Python

```
n = 1000000000
n += 0 if n % 2 == 1 else 1
sign = 1
t = 0
for i in range(1, n + 2, 2):
    t += (1.0 / i) * sign
    sign = -sign

s = 4 * t
print(s)
```

24.111s

C

```
#include <stdio.h>

int n,i;
int sign = 1;
double t, s;

int main(int argc, char ** argv) {
    n = 1000000000;
    n += (n % 2 == 1 ? 0 : 1);
    for(i = 1; i < n + 2; i += 2) {
        t += (1.0 / (double) i) * sign;
        sign = -sign;
    }
    s = 4 * t;
    printf("pi = %f\n", s);
}
```

0.581s



High vs. Low Level



Python

- ❖ int 可以 “任意” 大
- ❖ string, list, tuple, dict, set等是内置数据类型

C

- ❖ int 存在溢出和符号问题
- ❖ 仅有int, float, double, char类型



Memory Management



哈爾濱工業大學
HARBIN INSTITUTE OF TECHNOLOGY

Python

- ❖ 无指针的概念?
- ❖ 自动垃圾回收机制

```
node = (1, None)
node = (2, node)
node = (3, node)
while node != None:
    print(node[0])
    node = node[1]
```

C

- ❖ 额外指针的概念
- ❖ 内存需要自行申请和释放



Python

- ❖ 面向过程
- ❖ 面向对象
- ❖ 函数式编程

C

- ❖ 面向过程



Python

```
teststr = '1234'
```

```
teststr[1] = '1' x
```

```
teststr = teststr.replace('2', '1') ✓
```

```
teststr = teststr[:1] + '1' + teststr[2:]  
✓
```

C

```
char teststr[] = "1234";
```

```
teststr[1] = '1';
```

字符串不可变



不可变对象

- ❖ 简单类型
- ❖ 整数
- ❖ 字符串
- ❖ Tuple

- ❖ 对于不可变对象，“=”意味着绑定新对象，与原对象无关

可变对象

- ❖ List
- ❖ Dict
- ❖ Object

- 对于不可变对象，其改变一般也是通过调用响应的方法，例如append



Python

class

C

```
struct sStruct{  
    int number;  
    char[4] name;  
}
```

```
union sUnion{  
    int number;  
    char[4] code;  
}
```



Function



哈爾濱工業大學
HARBIN INSTITUTE OF TECHNOLOGY

Python

```
def printArea( width = 1, height = 2 ):
    print(width * height)
```

```
printArea()
printArea(4, 2.5)
printArea(height =5, width =3)
printArea(width = 1.2)
printArea(height =6.2)
```

C

```
void printArea( float width, float height)
{
    printf( "%f\n", width * height );
}

int main( int argc, char* argv )
{
    printArea(1, 2);
}
```

面向对象的函数定义



Function(参数传递)



哈爾濱工業大學
HARBIN INSTITUTE OF TECHNOLOGY

Python

```
def swap(x, y):
```

```
    t = x
```

```
    x = y
```

```
    y = t
```

```
a = 1
```

```
b = 2
```

```
swap(a, b)
```

C

```
void swap( int a, int b )
```

```
{
```

```
    int t;
```

```
    t = b;
```

```
    b = a;
```

```
    a = t;
```

```
}
```

```
int main( int argc, char* argv )
```

```
{
```

```
    int a = 1;
```

```
    int b = 2;
```

```
    printf( "a=%d\tb=%d\n", a, b);
```

```
    swap(a, b);
```

```
    printf( "a=%d\tb=%d\n", a, b);
```

```
    return 0;
```

```
}
```

C

```
void swap( int* a, int* b )
```

```
{
```

```
    int t;
```

```
    t = *b;
```

```
    *b = *a;
```

```
    *a = t;
```

```
}
```

```
int main( int argc, char* argv )
```

```
{
```

```
    int a = 1;
```

```
    int b = 2;
```

```
    printf( "a=%d\tb=%d\n", a, b);
```

```
    swap(&a, &b);
```

```
    printf( "a=%d\tb=%d\n", a, b);
```

```
    return 0;
```

```
}
```




Python

```
a, b = 1, 2  
print a, b  
a, b = b, a  
print a, b
```

```
def f(x, y):  
    return x + y, x - y, x * y, x / y;  
t1, t2, t3, t4 = f(9, 5)
```

C

```
int a, b = 1, 2 x
```

```
int returnValue()  
{  
    return 1, 2, 10;  
}  
  
int main( int argc, char* argv )  
{  
    int a, b;  
    a, b = returnValue();  
    printf( "%d\n", b);  
}
```



Python

```
try:
    number1, number2 = eval(input("two numbers: "))
    result = number1 / number2
    print "Result is", result
except ZeroDivisionError:
    print "Division by zero!"
except SyntaxError:
    print "A comma may be missing in the input"
except:
    print "Something wrong in the input"
else:
    print "No exceptions"
finally:
    print "at the end"
```

C

1. 使用标准C库提供了abort()和exit()两个函数，它们可以强行终止程序的运行
2. 使用assert(断言)宏调用，当程序出错时，就会引发一个abort()
3. 使用errno全局变量
4. 使用goto语句，当出错时跳转



String



哈爾濱工業大學
HARBIN INSTITUTE OF TECHNOLOGY

Python

```
a = 'abc'  
b = 'def'  
a < b #(True)  
c = a + b #('abcdef')
```

C

```
#include <string.h>  
char *a = "abc";  
char *b = "def";  
strcmp(a, b); //(-1)  
char c[7];  
strcpy(c, a);  
strcat(c, b);
```



List vs. Array



哈爾濱工業大學
HARBIN INSTITUTE OF TECHNOLOGY

Python

- ❖ list
- ❖ 存储类型任意
- ❖ 可扩充和删除
- ❖ 索引值
 - 0 ... (len-1)
 - -len ... -1
 - 不能越界，否则抛异常
- ❖ 切片 (slicing)
 - a[2:5]

C

- ❖ array
- ❖ 存储类型统一
- ❖ 不可扩充和删除
- ❖ 索引值
 - 0 ... (len-1)
 - 不能越界，越界难察觉



Compatibility



哈爾濱工業大學
HARBIN INSTITUTE OF TECHNOLOGY

Python

- ❖ 基于虚拟机
- ❖ 具有很强的可移植性
- ❖ 但Python 2和3兼容不好

C

- ❖ 源代码“可移植”



Applications



哈爾濱工業大學
HARBIN INSTITUTE OF TECHNOLOGY

Python

- ❖ Prototype system
- ❖ Data analyses
- ❖ Web application
- ❖ GUI

C

- ❖ Operating system
- ❖ Fast core algorithm
 - Python module
- ❖ Embedded system
- ❖ Hardware driver
- ❖ Real-time applications



Python

- ❖ `a = 1`
- ❖ `fileHandle = open('testBinary.txt', 'wb')`
- ❖ `fileHandle.write(a)`
- ❖ `fileHandle.close()`

C

- ❖ `fread(void * ptr, size_t size, size_t nmemb, FILE * stream)`
- ❖ `fwrite(const void * ptr, size_t size, size_t nmemb, FILE * stream)`

文件二进制读写



LOC & Performace



哈爾濱工業大學
HARBIN INSTITUTE OF TECHNOLOGY

Python

50 LOC

1 min runtime

C

100 LOC

0.5 min runtime