

8. Problem Set



《Python programming》 / Lecturer : Zhiyi Luo (罗志一)

School of Computer Science and Technology
计算机科学与技术学院



Determine the following statement is true or false.

- Python variable names are case sensitive, so `student` and `Student` are not the same variable.
- The '+' operator can be used to concatenate strings and generate a new string.
- A function must have at least one parameter.
- Python use indentation to determine the code structure.
- Given `lst=[1,3,5,7]`, function `lst.remove(1)` will remove the element with subscript 1.
- Given `a=(1,2,3,4)`, `a[0]=0` will modify `a` to `(0,2,3,4)`.



Determine the following statement is true or false.

- The expression $\{1,3,2\} > \{1,2,3\}$ has the value True.
- The expression $\{1,3,2\} >= \{1,2,3\}$ has the value True.
- Suppose x is a list with 5 elements, then the slice operation $x[10:]$ cannot be executed and an exception is thrown.
- Local variables defined inside the function are automatically deleted when the function call ends.



Multiple choices

- How many iteration times are there in statements `for i in range(20)` ?
- Statement `a={ }` will generate a _____.
- What's the return value of function `list("xyz")`?
- Suppose `list1=[1,2,3,0,4]`, then what is `list1[:-1]`?
- Suppose `list1=[1,2,3,0,4]`, then what is `list1[::-1]`?
- What's the output of the following code?

```
x='hello'  
y=2  
print(x+y)
```
- In Python which of the following is a valid variable name?
 - A. 2you B. my-name C. _item D. while



Program comprehension

- Write a program to reverse a string.

```
def string_reverse(str1):  
    rstr1 = ''  
    index = len(str1)  
    while index > 0:  
        _____(1)_____  
        index = index - 1  
    return rstr1  
  
print(string_reverse('1234abcd'))
```



Program comprehension

- Write a Python function that checks whether a passed string is palindrome or not. Note: A palindrome is a word, phrase or sequence that reads the same backward as forward.

```
def isPalindrome(s):  
    i = 0  
    j = len(s) - 1  
    while j > i:  
        if not s[i] == s[j]:  
            return False  
        i += 1  
        j -= 1  
    return True
```

```
print(isPalindrome('aza'))
```



Read the program and write the displaying results.

- Write the displaying result when the input is 5.
- Write the displaying result when the input is 10.

```
n = int(input())
s = 0
for i in range(1,n):
    s = s+i
    if i == 5:
        break
print(s)
```



Read the program and write the displaying results.

```
def find_domain(s):  
    words = s.split()  
    word = words[1]  
    print(word)  
    pos=word.find('@')  
    print(pos)  
    return (word[pos+1:])  
  
s = 'From alex@zstu.edu.cn Sat Jan'  
print(find_domain(s))
```




Programs

- Input an integer N , and calculate: $S=1+1/2+1/4+\dots+1/2^N$. If the input is not a positive integer, print “Invalid input!”, otherwise, print the value of S and keep two digits after the decimal point.



String formatting

- String formatting is the process of infusing things in the string dynamically and presenting the string.



String formatting

● Formatting string using % operator

- It is the oldest method of string formatting. Here we use the modulo % operator. The modulo % is also known as the “string-formatting operator”.

```
print("The mangy, scrawny stray dog %s gobbled down" +  
      "the grain-free, organic dog food." % 'hurriedly')
```

- We can also inject multiple strings at a time and can also use variables to insert objects in the string.

```
x = 'looked'
```

```
print("Misha %s and %s around"%('walked',x))
```



String formatting

- Formatting string using % operator
 - %s is used to inject strings similarly %d for integers, %f for floating-point values, %b for binary format. For all formats, conversion methods visit the official documentation.

```
print('Joe stood up and %s to the crowd.' % 'spoke')
```

```
print('There are %d dogs.' %4)
```



Float precision with the placeholder method

- Floating-point numbers use the format `%a.bf`. Here, **a** would be the minimum number of digits to be present in the string; these might be padded with white space if the whole number doesn't have this many digits. Close to this, **bf** represents how many digits are to be displayed after the decimal point.



Float precision with the placeholder method

```
print('The value of pi is: %5.4f' %(3.141592))
```

The value of pi is: 3.1416



Float precision with the placeholder method

```
print('Floating point numbers: %1.0f' %(13.144))
```



Float precision with the placeholder method

- You can use multiple format conversion types in a single print statement

```
variable = 12
```

```
string = "Variable as integer = %d \n\  
Variable as float = %f" %(variable, variable)
```

```
print (string)
```

```
Variable as integer = 12  
Variable as float = 12.000000
```