

1 ICS 104 - Introduction to Programming in Python and C

1.1 Functions

2 Lab Objectives

- To be able to implement functions
- To become familiar with the concept of parameter passing
- To develop strategies for decomposing complex tasks into simpler ones
- To be able to determine the scope of a variable

3 Worked Example

- **Problem Statement:** Many web sites and software packages require you to create passwords that contain at least one digit and one special character. Your task is to write a program that generates such a password of a given length. The characters should be chosen randomly.
- **Step 1:** Describe what the function should do.
 - The problem description asks you to write a program, not a function. We will write a password- generating function and call it from the program's main function.
 - Let us be more precise about the function. It will generate a password with a given number of characters.
 - We could include multiple digits and special characters, but for simplicity, we decide to include just one of each. We need to decide which special characters are valid.
 - For our solution, we will use the following set:

+ - * / ? ! @ # \$ % &

- The remaining characters of the password are letters. For simplicity, we will use only lowercase letters in the English alphabet.

- **Step 2:** Determine the function's "inputs".
- There is just one parameter: the length of the password.

- **Step 3:** Determine the types of the parameter variables and the return value.
- At this point, we have enough information to document and specify the function header:

```
## Generates a random password.  
# @param length an integer that specifies the length of the password  
# @return a string containing the password of the given length with one digit  
# and one special character  
#  
def makePassword(length) :
```

- **Step 4:** Write pseudocode for obtaining the desired result.

- One possible approach for making a password is as follow:

Make an empty string called password.
Randomly generate length – 2 letters and append them to password.
Randomly generate a digit and insert it at a random location in password.
Randomly generate a symbol and insert it at a random location in password.

- How do we generate a random letter, digit, or symbol? How do we insert a digit or symbol in a random location?
- We will delegate those tasks to helper functions.

- Each of those functions starts a new sequence of steps, which, for greater clarity, we will place after the steps for this function.

- **Step 5:** Implement the function body.

- We need to know the “black box” descriptions of the two helper functions described in Step 4 (which we will complete after this function). Here they are:

```
## Returns a string containing one character randomly chosen from a given string.
# @param characters the string from which to randomly choose a character
# @return a substring of length 1, taken at a random index
#
def randomCharacter(characters) :

## Inserts one string into another at a random position.
# @param string the string into which another string is inserted
# @param toInsert the string to be inserted
# @return the string that results from inserting toInsert into string
#
def insertAtRandom(string, toInsert) :
```

- Now we can translate the pseudocode in Step 4 into Python:

```
In [ ]: 1 def makePassword(length) :
2         password = ""
3
4         # Pick random Letters.
5         for i in range(length - 2) :
6             password = password + randomCharacter ("abcdefghijklmnopqrstuvwxyz")
7
8         # Insert a random digit and a random special character.
9         randomDigit = randomCharacter("0123456789")
10        password = insertAtRandom(password, randomDigit)
11
12        randomSymbol = randomCharacter("+-*/?!@#$$%&")
13        password = insertAtRandom(password, randomSymbol)
14
15        return password
16
```

- **Step 6:** Test your function.
- Here is a simple main function that calls the makePassword function:

In []:

```
1 def main() :  
2     result = makePassword(8)  
3     print(result)
```

In [7]:

```
1  ##
2  # This program generates a random password.
3  #
4
5  from random import randint
6
7  def main() :
8
9      result = makePassword(8)
10     print(result)
11
12     ## Generates a random password.
13     # @param length an integer that specifies the length of the password
14     # @return a string containing the password of the given length with one digit
15     # and one special character
16     def makePassword(length) :
17         password = ""
18
19         # Pick random Letters.
20         for i in range(length - 2) :
21             password = password + randomCharacter("abcdefghijklmnopqrstuvwxyz")
22
23         # Insert a random digit and a random special character.
24         randomDigit = randomCharacter("0123456789")
25         password = insertAtRandom(password, randomDigit)
26
27         randomSymbol = randomCharacter("+-*/?!@#%&")
28         password = insertAtRandom(password, randomSymbol)
29
30         return password
31
32     ## Returns a string containing one character randomly chosen from a given string.
33     # @param characters the string from which to randomly choose a character
34     # @return a substring of length 1, taken at a random index
35     #
36     def randomCharacter(characters) :
37         n = len(characters)
38         r = randint(0, n - 1)
39         return characters[r]
40
41     ## Inserts one string into another at a random position.
42     # @param string the string into which another string is inserted
```

```
43 # @param toInsert the string to be inserted
44 # @return the string that results from inserting toInsert into string
45 #
46 def insertAtRandom(string, toInsert) :
47     n = len(string)
48     r = randint(0, n)
49     result = ""
50
51     for i in range(r) :
52         result = result + string[i]
53     result = result + toInsert
54     for i in range(r, n) :
55         result = result + string[i]
56 # we can replace the above 2 loops by the statement below using slicing
57 # result=string[0:r]+toInsert+string[r:n]
58     return result
59
60 # Start the program.
61 main()
```

tod9kso&

4 Exercises

- **Exercise # 1** Write a function `def repeat(String, n, DELIM)` that returns the string `String` repeated `n` times, separated by the string `DELIM`. For example, `repeat("ho", 3, ", ")` returns "ho, ho, ho".
- The following is a sample run

```
Enter a string: Wow
How many time to repeat: 7
Enter a delimiter: **
Wow**Wow**Wow**Wow**Wow**Wow**Wow
```

```
In [1]: 1 # YOUR CODE HERE
2
3 def main():
4     repeat(input("Enter a string: "),int(input("How many time to repeat: ")),input("Enter a delimiter: "))
5
6
7 def repeat(STRING,n,DELIM):
8     for i in range(1,n+1):
9         print(STRING,end="")
10        if i < n:
11            print(DELIM,end="")
12
13
14 main()
```

```
Enter a string: Wow
How many time to repeat: 7
Enter a delimiter: **
Wow**Wow**Wow**Wow**Wow**Wow**Wow
```

- **Excercise # 2** Write a function `def isLeapYear(year)` that tests whether a year is a leap year, that is, a year with 366 days. Note that a year is a leap year if it is divisible by 4 but not by 100, or if it is divisible by 400.
- Following are sample runs:

- ```
enter a year number: 1600
1600 is a leap year
```
- ```
enter a year number: 2020
2020 is a leap year
```
- ```
enter a year number: 1000
1000 is not a leap year
```

In [2]:

```

1 # YOUR CODE HERE
2
3 def main():
4 isLeapYear(int(input("enter a year number: ")))
5
6
7
8 def isLeapYear(year):
9 if year % 4 == 0 and year % 100 != 0 or year % 400 == 0:
10 result = (str(year) + " is a leap year")
11 else:
12 result = (str(year) + " is not a leap year")
13 print(result)
14
15 main()

```

```

enter a year number: 1600
1600 is a leap year

```

**Exercise # 3** Write 3 functions to draw a box using a certain character. the function drawLine will receive 2 arguments; the character and how many times it is printed on one line. Then it will send the cursor to a new line. The function drawHollow will receive 2 arguments; the character and n. It will print the character, followed by n-2 blanks, followed by the character, then new line. The function drawBox will receive 3 arguments; the character, width and height. With the help of the previous 2 functions, it will display the box of a received character on the screen.

drawBox(8,6,'\$') will display the box shown below.

**4.0.0.1 Note:** You have to use 3 functions described above not one only to learn how functions call each other. Also, you are not allowed to use the \* operator to repeat a string. Use loops.

```

1 enter width of box: 8
2 Enter height of box: 6
3 Enter drawing symbol: $
4 $$$$$$$$
5 $ $
6 $ $
7 $ $
8 $ $
9 $$$$$$$$

```

In [4]:

```
1 # YOUR CODE HERE
2
3 def main():
4 drawBox(int(input("Enter width of box: ")),int(input("Enter height of box: ")),input("Enter drawing symbol: "))
5
6 def drawBox(width,length,character):
7 drawLine(width,character)
8 for i in range(length-2):
9 drawHollow(width,character)
10 drawLine(width,character)
11
12 def drawLine(n,character):
13 print(character*n)
14
15 def drawHollow(n,character):
16 print(character+" "*(n-2)+character)
17
18 main()
19
```

```
Enter width of box: 8
Enter height of box: 6
Enter drawing symbol: $
$$$$$$$
$ $
$ $
$ $
$ $
$$$$$$$
```