

[Kelas B+D] Lab 3 Dasar-Dasar Pemrograman 1

String

DEADLINE: 22 September 2017, pukul 15.30 waktu SCell

Selamat datang di Lab/Tutorial Dasar-Dasar Pemrograman 1!

Pertama-tama kami ucapkan selamat kepada kalian semua para peTARUNG yang sudah dianugerahi kesempatan untuk mengejar sarjana di Fasilkom UI. Percayalah, bahwa langkah pertama meraih S.Kom *pasti* bermula dari Dasar-Dasar Pemrograman. Maka dari itu DDP sebaiknya diikuti dengan seksama dan jangan kendor semangatnya! :)

String Representation

Dalam Python, sebuah string dinyatakan dengan tanda petik satu (' ') atau petik dua (" "). Untuk menyatakan sebuah kalimat yang mengandung tanda petik satu sebagai satu kesatuan dalam kalimat tersebut (seperti dalam kalimat berbahasa Inggris), kita dapat menggunakan *escape sequence* \ ' atau \"

Contoh:

```
>>> excuse = 'I am sick'
>>> excuse = "I am sick"
>>> excuse = 'I'm "sick"'
SyntaxError: invalid syntax
>>> excuse = 'I\'m "sick"'
>>> print(excuse)
I'm "sick"
```

String Indexing

Dalam Python, terdapat sebuah method untuk melakukan indexing. Untuk melakukan indexing pada suatu variable kita dapat menggunakan syntax `variable_name[x]`, dengan x merupakan index dari sesuatu di variabel `variable_name`. Indexing juga berguna dalam melakukan slicing pada string. Untuk melakukan slicing, kita dapat menggunakan syntax `variable_name[a:b:c]`, dengan a sebagai index awal (bersifat inklusif), b sebagai index akhir (bersifat eksklusif), dan c adalah index lompatan dari satu karakter ke karakter yang lain. Index lompatan tidak bersifat wajib dan dapat digunakan ketika kita membutuhkannya. Jika a dikosongkan, maka slicing akan dimulai dari index awal. Jika b dikosongkan, maka slicing akan dilakukan hingga akhir kata/kalimat.

Contoh:

```
>>> kata = "Fasilkom"
>>> kata[2]
's'
>>> kata[:2] #melakukan slicing dari index ke-0 sampai ke-1
'Fa'
>>> kata[3:] #melakukan slicing dari index ke-3 sampai akhir kalimat
'Ilkom'
>>> kata[::-1] #melakukan reversing pada kalimat
'moklisaF'
```

String Method

Dalam Python, terdapat beberapa method yang sering digunakan untuk memanipulasi string. Namun, perlu ditekankan kembali, String bersifat immutable, sehingga tidak ada satupun string method yang memodifikasi sebuah string yang tersimpan dalam suatu variabel. Beberapa contoh method dari string adalah:

```
>>> s = 'http://www.main.com/smith/index.html'
>>> s.upper() #membuat string menjadi huruf besar
'HTTP://WWW.MAIN.COM/SMITH/INDEX.HTML'
>> s.find('smith')
20
>>> s.capitalize()
'http://www.main.com/smith/index.html'
>>> s.replace('smith', 'ferreira')
'http://www.main.com/ferreira/index.html'
>>> s
'http://www.main.com/smith/index.html'
>>> s.split('/')
['http:', '', 'www.main.com', 'smith', 'index.html']
>>> len(s)
36
```

Lalu, kalau string bersifat immutable, gimana caranya agar kita dapat mengubah secara permanen sebuah string yang disimpan dalam suatu variabel? Caranya adalah, kita dapat menyimpan hasil manipulasi string ke dalam sebuah variabel yang baru.

```
>>> contoh = "FASILKOM"
>>> contoh.lower()
'Fasilkom'
>>> contoh
'FASILKOM'
>>> contoh_baru = contoh.lower()
>>> contoh_baru
'fasilkom'
```

Untuk mengetahui method-method lainnya dari tipe data string, kalian bisa melihatnya di:

<https://docs.python.org/3/library/string.html>

SOAL TUTORIAL

Classic Cryptography

Benny dan Ibnu merupakan dua orang sahabat karib. Mereka suka bertukar pesan satu sama lain. Namun, mereka tidak ingin pesan mereka dapat diketahui oleh orang lain, apalagi jika mereka sedang membicarakan wanita idaman mereka, sehingga mereka menggunakan enkripsi pada pesan mereka. Suatu hari, Anda tidak sengaja mengetahui teknik enkripsi mereka, karena Anda menguping pembicaraan mereka. Mereka menggunakan teknik “Caesar cipher” dalam melakukan enkripsi pesan.

(https://en.wikipedia.org/wiki/Caesar_cipher).

Teknik enkripsi mereka adalah untuk setiap huruf pada pesan, dilakukan operasi: Diketahui K , maka lakukan pergeseran ke arah kanan sebesar K (Misal $K = 1$, “a” menjadi “b”, “j” menjadi “k”, dan “z” menjadi “a”).

Anda sebagai orang yang ingin mengetahui pesan apa yang dikirimkan oleh Benny dan Ibnu mencoba untuk membuat program agar tidak perlu melakukan dekripsi secara manual.

Format Masukan

Masukan berupa satu baris yang menunjukkan berapa banyak pergeseran ke arah kanan ($0 \leq K < 26$) dan pesan yang telah dienkripsi (pesan yang telah dienkripsi dipastikan huruf kecil semua)

Format masukan adalah sebagai berikut: **K encrypted_text**

Format Keluaran

Keluaran berupa satu baris yang menunjukkan pesan awal sebelum dienkripsi.

Hint #1

Kamu bisa menggunakan import string untuk mendapatkan seluruh alfabet huruf kecil tanpa harus mengetikannya secara manual.

Contoh Input/Output (bagian input yang diketikkan oleh user **ditebalkan**)

```
Masukkan operasi: 1 plf
Kalimat aslinya adalah: oke
```

Hint #2

Kamu dapat menggunakan sebuah method untuk memisahkan antara K dan kalimat yang sudah dienkripsi. Method apakah itu?

```
Masukkan operasi: 12 mwg nuem ppb
Kalimat aslinya adalah: aku bisa ddp

Masukkan operasi: 24 kclhybg kyfyqqguy dyqgjmk kcpsnyiyl qsyrs iczyleeeyl
Kalimat aslinya adalah: menjadi mahasiswa fasilkom merupakan suatu
kebanggaan

Masukkan operasi: 15 sse tphn etphn atbdc hfjtton
Kalimat aslinya adalah: ddp easy peasy lemon squeezy
```

Soal Bonus (Yey... ada bonus!)

Selain menggunakan teknik yang standar, kamu juga bisa memanfaatkan `chr()` dan `ord()` dalam program-mu. Cobalah buat teknik dekripsi Caesar cipher dengan memanfaatkan `chr()` dan `ord()`. Ketikkan program bonus-mu di dalam file yang sama.

Penilaian

- 80%** Kebenaran (Program berjalan dan berhasil menyelesaikan uji kasus),
- 10%** Kerapihan (Penamaan variabel jelas, mudah dibaca),
- 05%** Dokumentasi (Dokumentasi jelas, comment di bagian-bagian yang penting),
- 05%** Kesesuaian penamaan file submisi, kesesuaian output sesuai permintaan soal.

Penamaan File

Lab03_[Kelas]_[Kode Asdos]_[Nama]_[NPM].py

Contoh Penamaan File: Lab03_B_BI_Benny Ibnu_1606889911.py

**)kamu tidak perlu menuliskan kurung siku dalam penamaan file kamu.*

SELAMAT MENGERJAKAN DAN HAPPY CODING 🐍