Week 1 Notes (Variables, Data Type and Operator, Function)

Variables

Fungsi Variabel

Fungsi variabel adalah untuk menyimpan nilai

Mendeklarasikan Variabel

Cara mendeklarasikan variabel adalah dengan `nama_variabel = nilai`. Contohnya `a = 1`, a disini merupakan nama variabel dan 1 disini merupakan nilai.

Cara penamaan variabel

Cara penamaan variabel dalam programming merupakan kesepakatan universal. Bahasa pemrograman yang berbeda menggunakan sistem penamaan yang berbeda. Dalam bahasa pemrograman Python, kita akan menggunakan sistem "Snake Case". Berikut adalah cara penamaan variabel:

1. Nama variabel tidak boleh memiliki spasi. Jika ingin memisahkan kedua kata, gunakan `_`. Contoh: `file_number = 2`.
2. Nama variabel tidak boleh dimulai dengan angka. Tetapi boleh memiliki angka setelah huruf pertama. Contoh: `s123 = 13`.
3. Nama variabel tidak boleh memiliki simbol (`@`, `!`, `?`, dsb).

Comments (Komentar)

Comment di dalam programming sangat berguna karena bisa membantu orang untuk mengerti isi dari kode tersebut tanpa harus membaca kode yang ditulis. Biasanya komentar tidak akan dieksekusi di dalam program dan komentar dimulai dengan `#`.

Contoh:

In [1]:
```
num = [30, 50, 14, 85, 63, 22, 11, 8] # ini merupakan list <- komentar

# untuk mencari nilai maximum <- komentar
def find_max(lst):
    maximum = lst[0]
    for i in range(len(lst)):
        if lst[i] > maximum:
            maximum = lst[i]
    return maximum

print(find_max(num)) # output 85 <- komentar
```

```
85
```

Print

Cara untuk mengoutput nilai ke console / terminal adalah dengan menggunakan kata kunci `print()`.

Contoh:

In [2]:
```
a = 1
print(a) # 1
```

```
1
```

Data Types and Operators

Data Types

Tipe Data di Python pada umumnya ada 4:

1. `string`: biasa disingkat `str` berupa kata / kalimat biasanya dimulai dengan tanda kutip pembuka dan diakhiri dengan tanda kutip penutup. Contoh: `"hello world"`.
2. `integer`: biasa disingkat `int`. berupa angka bulat positif dan negatif.
3. `float`: berupa angka desimal.
4. `boolean`: True / False

Note: Tipe data yang berbeda tidak dapat digabungkan.

Untuk memeriksa tipe data dari suatu kode, kita bisa menggunakan kata kunci `type()`

Contoh:

In [3]:
```
print(type(2)) # integer
print(type(3.4)) # float
print(type("Hello World")) # string
print(type(True)) # boolean
```

```
<class 'int'>
<class 'float'>
<class 'str'>
<class 'bool'>
```

Operators

Operator terbagi menjadi 2 bagian:

1. Arithmetic (Aritmatika): `+`, `-`, `*`, `/`, `**` (pangkat), `%` (modulo / sisa bagi)
2. Comparison (Perbandingan): `<`, `>`, `<=`, `>=`, `==`

String Operator (+)

Tanda `+` di dalam pemrograman tidak hanya digunakan untuk menambahkan nilai, melainkan bisa digunakan untuk *menggabungkan kedua string* secara bersamaan.

Contoh:

In [4]:
```
str1 = "Hello"
str2 = "World"
space = " "

print(str1 + space + str2)
```

```
Hello World
```

Mengubah tipe data yang bukan String menjadi String

Terkadang kita perlu mengubah tipe data karena program hanya bisa menggabungkan tipe data yang sama. *Untuk mengubah tipe data yang lain menjadi string*, maka kita menggunakan kata kunci `str()`.

Contoh:

In [5]:
```
age = 10 # integer
print("Saya berusia " + str(age) + " tahun.")
```

```
Saya berusia 10 tahun.
```

Boolean Operator (AND, OR, NOT)

Tabel berikut adalah logika matematika untuk mendapatkan hasil boolean operator.

In [6]:
```
# Kode dibawah ini digunakan untuk menampilkan tabel seperti di output.
# Kita tidak akan mempelajari kode dibawah di kelas ini karena ini berkaitan dengan OOP (Object - Oriented Programming)
# Fokus saja pada output dari kode ini.
from prettytable import PrettyTable

table = PrettyTable()

table.field_names = ["A", "B", "NOT A", "NOT B", "A and B", "A or B"]
table.add_rows([["T", "T", "F", "F", "T", "T"],
                ["T", "F", "F", "T", "F", "T"],
                ["F", "T", "T", "F", "F", "T"],
                ["F", "F", "T", "T", "F", "F"]])

print(table)
```

```
+---+---+-------+-------+---------+--------+
| A | B | NOT A | NOT B | A and B | A or B |
+---+---+-------+-------+---------+--------+
| T | T |   F   |   F   |    T    |   T    |
| T | F |   F   |   T   |    F    |   T    |
| F | T |   T   |   F   |    F    |   T    |
| F | F |   T   |   T   |    F    |   F    |
+---+---+-------+-------+---------+--------+
```

Function

Kegunaan Fungsi

Kenapa kita harus mempelajari function? Function itu sifatnya fleksibel dan bisa digunakan berulang kali. Ini sangat berguna jika kita mengerjakan sebuah project yang begitu besar jumlah kodenya. Dengan menggunakan fungsi, maka kita bisa meminimalisir error yang akan terjadi dan kode kita terlihat lebih bersih.

Cara Mendeklarasikan Fungsi

Berikut adalah cara mendeklarasikan sebuah fungsi:

In [7]:
```
def function_name():
    # code here
    return None
```

- `def` menyatakan bahwa kita mulai mendeklarasikan fungsi
- `function_name` menyatakan nama fungsi
- `()` merupakan tempat kita menempatkan parameter
- `:` menyatakan akhir dari pendeklarasian fungsi
- `return` mengembalikan nilai dari fungsi

Cara Mengimplimentasikan Fungsi

Dalam matematika biasanya fungsi ditulis dengan `f(x) = x ** 2 + x + 1`. Jika kita mengimplimentasikannya ke dalam coding, maka dia akan menjadi seperti berikut.

In [8]:
```
def f(x): # x disini merupakan parameter karena dia menentukan apa hasil dari output
    return x ** 2 + x + 1

print(f(2))
```

```
7
```

Sama saja halnya dengan keliling lingkaran. Jika kita mengimplimentasi keliling lingkaran dengan menggunakan coding, maka dia akan menjadi seperti berikut

In [9]:
```
def k_lingkaran(radius): # radius disini merupakan parameter karena dia menentukan hasil dari output
    pi = 3.14
    k = 2 * pi * radius
    return k

print(k_lingkaran(10))
```

```
62.800000000000004
```

Bagaimana jika fungsi memiliki banyak parameter? Kita akan memisahkan fungsi dengan `,`. Salah satu contohnya disini adalah luas segitiga dengan parameter `alas` dan `tinggi`

In [10]:
```
def l_segitiga(alas, tinggi):
    l = 0.5 * alas * tinggi
    return l

print(l_segitiga(10, 50)) # jika terdapat 2 parameter, maka kita juga harus mengisi parameter dengan 2 nilai
```

```
250.0
```

Week1 Programming Assessment: https://github.com/winstencoellins/programming-fundamentals-week1