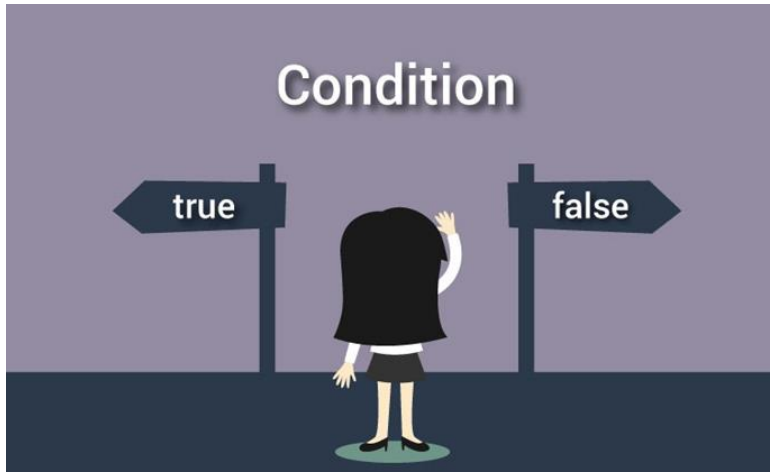
Python: Vol. 2

BPH – DASCO UNJ

Preliminary

CONDITIONS
LOOPING
EXCEPTION
FUNCTION

Conditions



BPH – DASCO UNJ

CONDITION

- Melakukan perintah (operasi) sesuai dengan kondisi yang diberikan

- Perintah di python:
  1. *if* : Kondisi awal
  2. *elif* (else if) : Kondisi berikutnya jika **Kondisi awal tidak terpenuhi**
  3. *else* : Negasi dari setiap kondisi yang telah di sebutkan sebelumnya

```
if KONDISI:
    # Block of Code
    pass
elif KONDISI_2:
    # Block of Code
    pass
else:
    # Block of Code
    pass
```

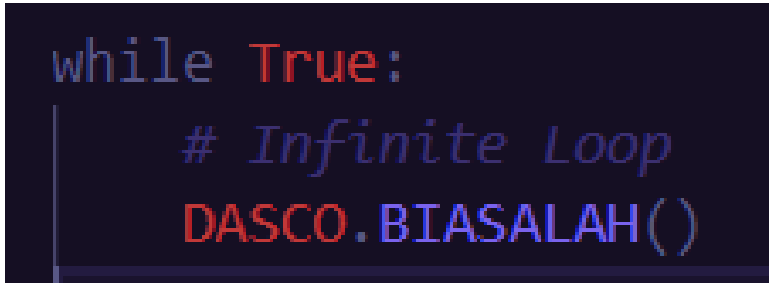
ONE LINE CONDITION

- Sangat berguna ketika assign nilai kedalam sebuah variable dengan sebuah kondisi

- Tidak di anjurkan untuk kondisi yang komplex

```
# One Line Condition
VARIABLE = "Nilai jika kondisi benar" if KONDISI else "Nilai jika kondisi Salah"
```

Looping



- Melakukan pengulangan sesuai dengan kondisi atau pada elemen - elemen tipe data sequence
- Perintah di Python:

    1. while : Perintah yang akan mengulang satu set code selama kondisi pengulangannya True
    2. for : Perintah yang akan mengulang satu set code sebanyak element yang ada pada sebuah data bertipe sequence

While Loop

```
i = 0
while i < 10:
    print(i)
    i += 1
```

- Melakukan pengulangan terhadap satu set kode selama kondisi pengulangannya masih True
- Jika kondisi pengulangannya tidak berubah maka akan terjadi Infinite loop

For Loop

```
for i in range(10):
    print(i)
```

- Melakukan pengulangan terhadap satu set kode sebanyak jumlah element yang ada di dalam sequence

Continue & Break

```
for i in range(10):
    if i < 5:
        continue
    print(i)
```

```
while True:
    print("BIASALAH")
    break
```

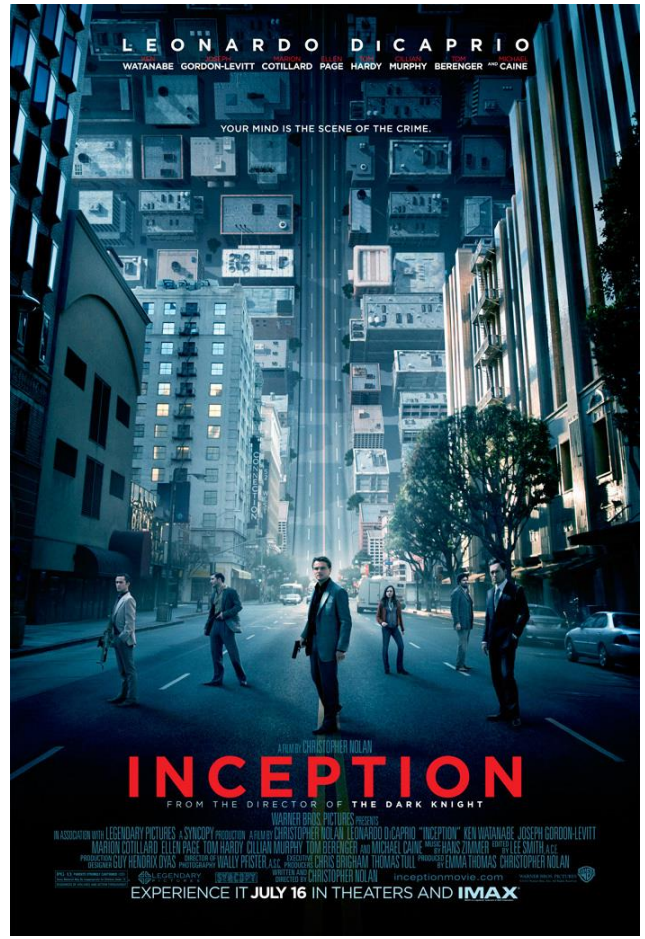
- Continue : Digunakan untuk melewati sebuah set kode yang berada di bawahnya pada saat looping
- Break : Digunakan untuk keluar dari loop secara paksa

Magic Python : List Comprehension

Digunakan untuk medifinisikan sebuah tipe data sequence dengan ketentuan tertentu dari data sequence lainnya.

```
# List bilangan pangkat 5 dari 1 - 100
VARIABLE = [x**5 for x in range(100)]
# List bilangan genap dari 1 - 100
VARIABLE_2 = [x for x in range(100) if x % 2 == 0]
```

Exception



BPH – DASCO UNJ

Exception Handling

- Mekanisme penanganan error saat *runtime* (Program berjalan)
- Perintah di Python:

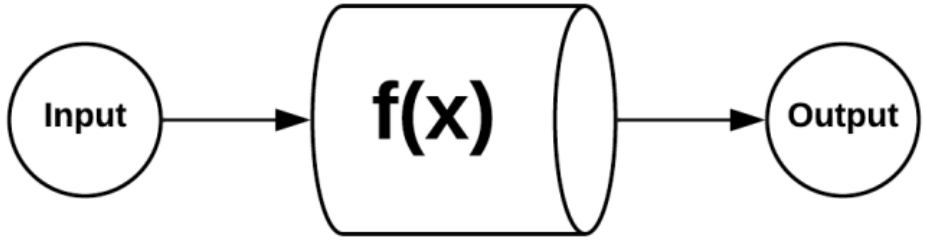
1. *try* : Perintah untuk mencoba satu set code
2. *except* : Perintah untuk menangani error jika terjadi pada satu set kode di dalam perintah *try*.

```
LIST = [1, 2, 3, 4, 'A']
for i in range(len(LIST)):
    try:
        LIST[i] += 1
    except TypeError:
        continue
```

Function

- Parameters Packing
- Rekursi
- Lambda

- Sebuah blok kode yang hanya akan tereksekusi jika di panggil
- Perintah di python:
    1. *def* : Untuk mendefinisikan fungsi
    2. *return* : Untuk mengembalikan hasil dari suatu fungsi




```
def FUNGSI_BARU(PARAMETERNYA):
    """Docstring"""
    # Block of Code
    return SESUATU
```

Parameters Packing

- Pack ke Tuple

```python
def FUNGSI_BARU(*PARAMETER2NYA):
    # Pack to Tuple
    return PARAMETER2NYA

PANGGIL_FUNGSI = FUNGSI_BARU(1, 2, 3, 4)
```

- Pack ke Dictionary

```python
def FUNGSI_BARU(**KEYWORD_ARGS):
    # Pack to Tuple
    return KEYWORD_ARGS

PANGGIL_FUNGSI = FUNGSI_BARU(SATU = 1, DUA = 2, TIGA = 3)
```

BPH – DASCO UNJ

Rekursi

```python
# Rekursi
def FACTORIAL(NUM):
    if NUM <= 1:
        return 1
    return NUM * FACTORIAL(NUM - 1)
```

- Rekursi adalah ketika sebuah fungsi yang di definisikan memanggil dirinya sendiri
- Jika sebuah fungsi rekursi didefinisikan tanpa exit condition maka akan terjadi Recursion Error

Lambda

```python
VARIABLE = lambda ARGS : RETURN_ARGS
```

- Cara lain membuat fungsi yang lebih sederhana
- Perintah di Python:
    1. *lambda* : perintah untuk medeklarasinkan fungsi sederhana

Terima Kasih

BPH – DASCO UNJ