



# Kondisi pada Python

**Selamat Datang!** Notebook ini akan mengajarkan Anda tentang condition (kondisi) pada bahasa pemrograman Python. Akhir dari notebook ini, Anda akan tahu bagaimana cara menggunakan pernyataan kondisi di Python, termasuk operator dan percabangan

## Daftar Isi

- [Pernyataan Kondisi](#)
  - [Operator Perbandingan \(comp\)](#)
  - [Percabangan \(branch\)](#)
  - [Operator Logika \(logic\)](#)
- [Kuis Pernyataan Condition](#)

Prakiraan waktu yang dibutuhkan: **20 menit**

## Pernyataan Kondisi

### Operator Perbandingan

Operator perbandingan membandingkan beberapa nilai atau operator dan, berdasarkan pada kondisi, akan menghasilkan nilai Boolean. Ketika membandingkan dua nilai, Anda dapat menggunakan operator berikut:

- sama dengan: `==`
- tidak sama dengan: `!=`
- lebih besar dari: `>`
- kurang dari: `<`
- lebih besar atau sama dengan: `>=`
- kurang dari atau sama dengan: `<=`

Tentukan `a` dengan nilai 5. Gunakan operator sama dengan, dengan dua tanda sama dengan (`==`) untuk menentukan jika dua nilai tersebut adalah sama. Kasus berikut membandingkan variabel `a` dengan 6

In [1]:

```
# Condition Equal  
  
a = 5  
a == 6
```

Out[1]:

False

Hasilnya adalah **False**, karena 5 tidak sama dengan 6.

Perhatikan operator pembandingan sama dengan `i > 5`. jika nilai dari sisi kiri operator, pada kasus ini adalah variabel `i`, adalah lebih besar dari sisi kanan operator, pada kasus ini adalah 5, kemudian pernyataannya adalah **True**. Jika tidak, pernyataannya adalah **False**. Jika `i` sama dengan 6, karena 6 lebih besar dari 5, hasilnya adalah **True**.

In [2]:

```
# Lebih besar dari  
  
i = 6  
i > 5
```

Out[2]:

True

Buat `i = 2`. Pernyataannya adalah False, karena 2 tidak lebih besar dari 5:

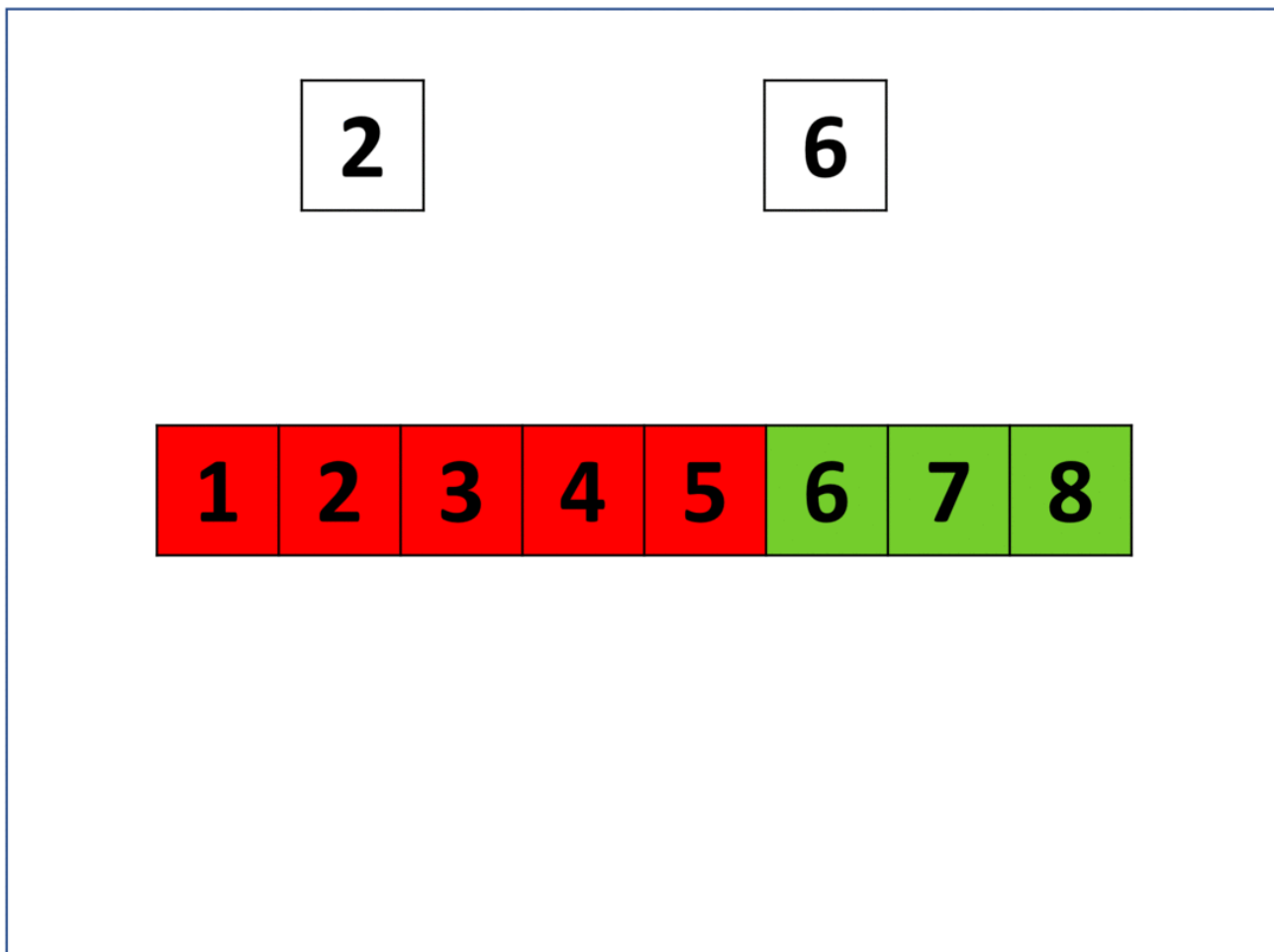
In [3]:

```
# Lebih besar dari  
  
i = 2  
i > 5
```

Out[3]:

False

Mari tampilkan beberapa nilai untuk `i` pada gambar. Atur nilai lebih besar dari 5 pada hijau dan lainnya pada merah. Area hijau merepresentasikan dimana kondisi adalah **True**, sisi merah dimana pernyataannya adalah **False**. Jika nilai dari `i` adalah 2, kita dapati **False** karena 2 berada pada area merah. Jika nilai dari `i` adalah 6, kita dapati **True** karena kondisinya berada pada area hijau.



Tes ketidaksamaan menggunakan tanda seru diawal tanda sama dengan, jika dua operan tidak sama dengan kemudian kondisi menjadi **True**. Sebagai contohm kondisi berikut akan menghasilkan **True** karena nilai dari `i` tidak sama dengan 6:

In [4]:

```
# Tanda tidak sama dengan  
  
i = 2  
i != 6
```

Out[4]:

True

Ketika `i` sama dengan 6, ekspresi tidak sama dengan menghasilkan **False**

In [5]:

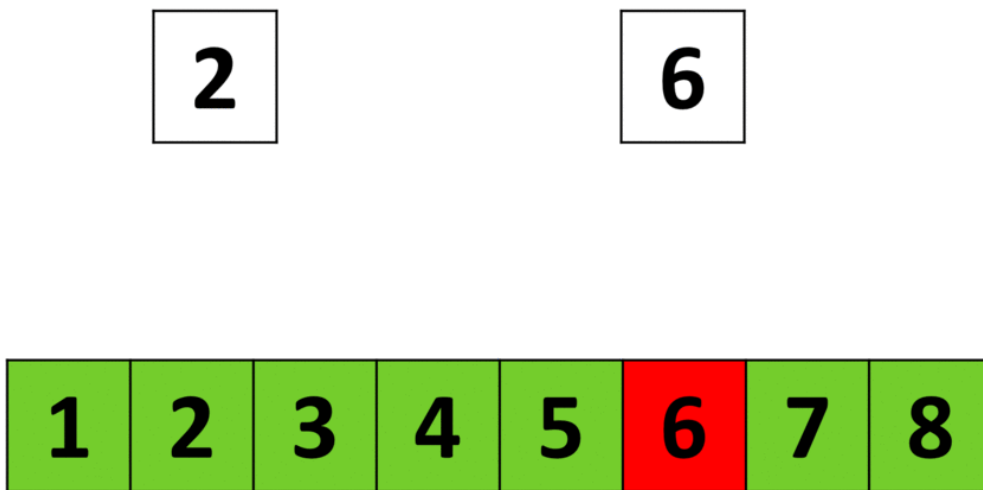
```
# Tanda tidak sama dengan
```

```
i = 6  
i != 6
```

Out[5]:

False

Lihat baris angka dibawah. Ketika kondisinya adalah **True** sesuai dengan angka ditandai dengan hijau dan dimana kondisinya adalah **False** sesuai dengan angka yang ditandai dengan merah. Jika kita atur `i` sama dengan 2, hasilnya adalah true pada area hijau. Jika kita atur `i` sama dengan 6, kita dapatkan **False** karena kondisinya pada area merah.



Kita dapat menerapkan metode yang sama pada string. Contohnya, gunakan operator sama dengan pada string yang berbeda. Karena string tidak sama, kita dapatkan **False**

In [6]:

```
# Gunakan tanda sama dengan untuk membandingkan string
```

```
"ACDC" == "Michael Jackson"
```

Out[6]:

False

Jika kita gunakan operator tidak sama dengan, hasilnya akan menjadi **True** karena kedua string tidak sama.

In [7]:

```
# Gunakan tanda tidak sama dengan untuk membandingkan string  
"ACDC" != "Michael Jackson"
```

Out[7]:

True

Operator ketidaksamaan juga digunakan untuk membandingkan kalimat/kata/symbol berdasarkan ASCII. Nilai desimal ditunjukkan pada tabel berikut yang merepresentasikan urutan dari karakter.

Sebagai contoh, kode ASCII untuk **!** adalah 21, kode ASCII untuk **+** adalah 43. Dengan demikian **+** lebih besar dari **!** karena 43 lebih besar dari 21.

Demikian pula, nilai ASCII untuk **A** adalah 101, dan nilai untuk **B** adalah 102 kemudian :

In [8]:

```
# Membandingkan karakter  
'B' > 'A'
```

Out[8]:

True

Ketika ada kata yang terdiri dari lebih dari 1 huruf, huruf pertama yang akan diperhitungkan terlebih dahulu:

In [9]:

```
# Membandingkan karakter  
'BA' > 'AB'
```

Out[9]:

True

**Note:** Kata dengan huruf besar memiliki perbedaan kode ASCII dibandingkan dengan kata dengan huruf kecil, yang berarti perbandingan antara kata pada python adalah case-sensitive.

## Branching (Percabangan)

Percabangan memperbolehkan kita untuk menjalankan perintah yang berbeda untuk input yang berbeda. Ini sangat membantu dalam berpikir **if statement** (pernyataan if) sebagai ruang yang terkunci, jika pernyataan **True** kita dapat masuk ke dalam ruangan dan program Anda akan menjalankan tugas yang telah didefinisikan, tapi jika pernyataannya adalah **False** program akan menolak tugas Anda.

Sebagai contoh, perhatikan persegi biru merepresentasikan sebuah konser ACDC. Jika individu-individu berumur lebih dari 18, mereka dapat memasuki konser ACDC. Jika mereka 18 atau lebih muda dari itu, mereka tidak dapat memasuki konser.

Gunakan pernyataan kondisi yang telah dipelajari sebelumnya sebagai kondisi yang perlu di cek dalam **if statement**. Sintaknya sederhana: *if condition statement* : , dimana mengandung kata **if** , sembarang pernyataan kondisi, dan tanda titik dua di akhir. Tulislah kode Anda pada baris baru di bawah titik dua dan dimulai dengan indentasi (menjorok maju dengan tab). Baris kode setelah titik dua dan dengan indentasi hanya akan dieksekusi ketika **if statement** adalah **True**. Eksekusi berakhir jika baris kode tidak mengandung indentasi.

Pada kasus dibawah, kode `print("you can enter")` akan dijalankan jika variabel `age` lebih besar dari 18 adalah **True** karena baris kode memiliki inden. Namun, eksekusi dari `print("move on")` tidak akan dipengaruhi oleh pernyataan **if**.

In [10]:

```
# Contoh pernyataan if

age = 19
#age = 18

# Ekspresi yang bisa jadi true atau false
if age > 18:

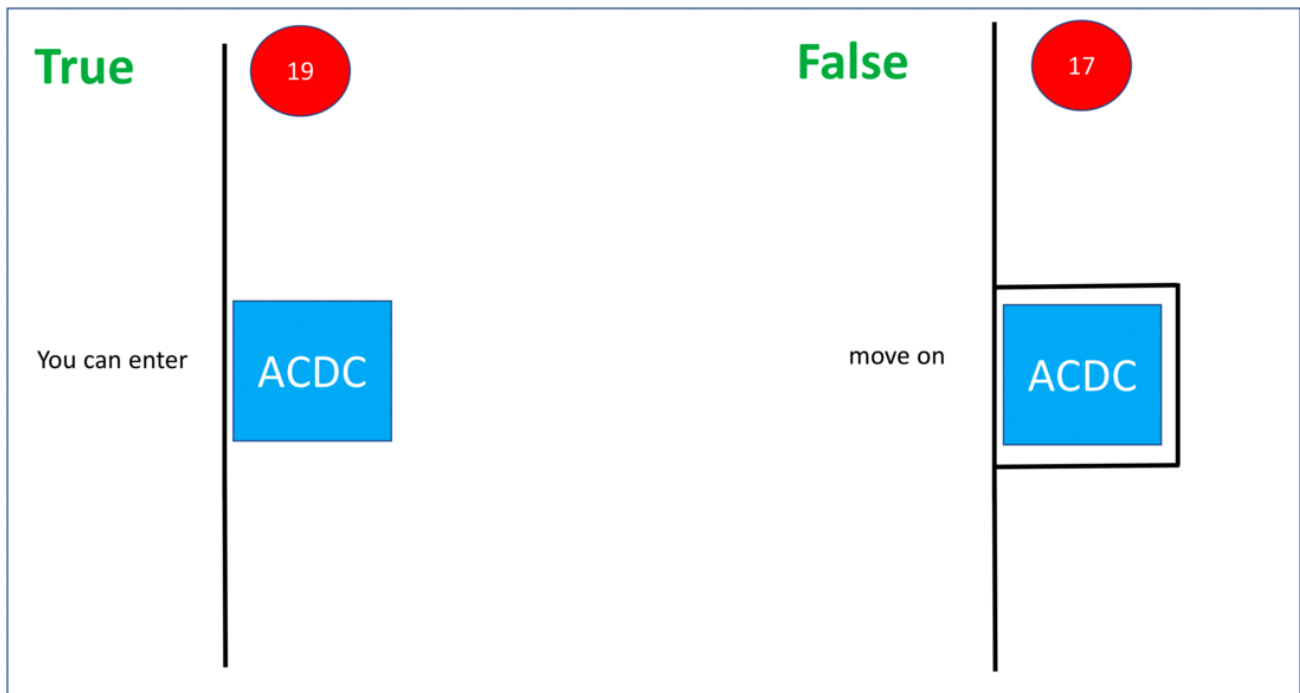
    # Dengan sebuah indentasi, kita memiliki ekspresi yang dijalankan jika kondisinya a
    # dalah true
    print("you can enter" )

# Pernyataan setelah if statemen akan dijalankan bagaimanapun juga jika kondisi adalah
# ture atau false
print("move on")
```

```
you can enter
move on
```

*Coba hapus komentar pada variabel age*

Ilustrasi dibawah sangat membantu untuk memahami prosesnya. Pada sisi kiri, kita lihat apa yang terjadi ketika kondisinya adalah **True**. Pengunjung memasuki konser ACDC direpresentasikan dengan kode dalam indentasi yang dieksekusi; mereka kemudian berjalan. Pada sisi kanan kita lihat apa yang terjadi ketika kondisinya adalah **False**; pengunjung tidak diberikan akses, dan pengunjung berjalan. Pada kasus ini, segmen kode dalam inden tidak dijalankan, tetapi kode di akhir pernyataan yang akan dijalankan.



Blok kode pada `else` akan dijalankan jika tidak ada kondisi **True** sebelum pernyataan `else` yang terpenuhi. Mari kita gunakan analogi ACDC lagi. Jika pengguna berusia 17, mereka tidak dapat pergi ke konser ACDC, tapi mereka bisa pergi ke konser Meatloaf. Sintaks dari pernyataan `else` mirip dengan sintaks dari pernyataan `if`, sebagai `else :`. Perhatikan, tidak ada pernyataan kondisi untuk `else`. Coba ubah nilai dari `age` untuk melihat apa yang terjadi:

In [11]:

```
# Contoh pernyataan else

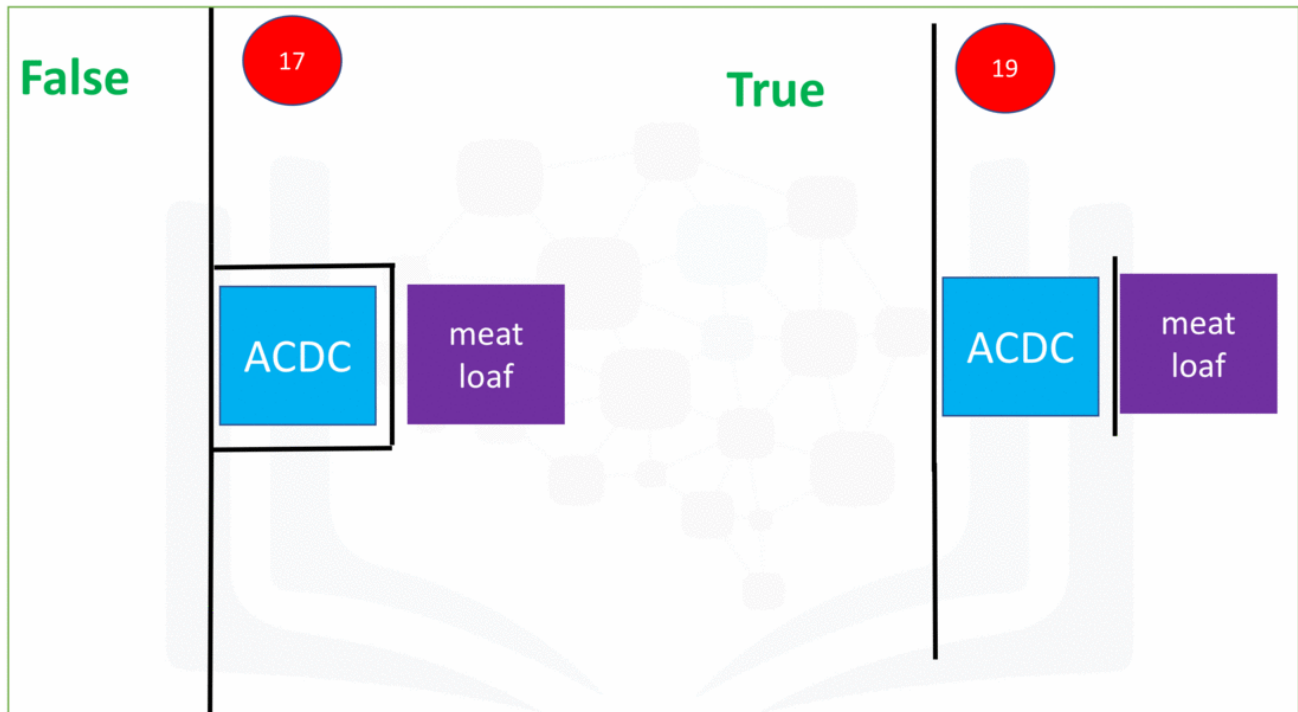
age = 18
# age = 19

if age > 18:
    print("you can enter" )
else:
    print("go see Meat Loaf" )

print("move on")
```

```
go see Meat Loaf
move on
```

Prosesnya didemonstrasikan dibawah ini, dimana tiap kemungkinan di ilustrasikan pada tiap sisi dari gambar. Pada sisi kiri adalah kasus dimana usia adalah 17, kita buat variabel `age = 17`, dan ini sesuai dengan kehadiran pengunjung pada konser Meatloaf. Porsi sebelah kanan menunjukkan apa yang terjadi ketika pengunjung berusia lebih dari 18, pada kasus ini adalah 19, dan pengunjung diberikan akses ke konser.



Pernyataan `elif`, kependekan dari `else if`, memperbolehkan kita untuk mengecek kondisi tambahan jika pernyataan kondisi adalah **False**. Jika kondisi dari pernyataan `elif` adalah **True**, ekspresi alternatifnya akan dijalankan. Perhatikan contoh pada konser, dimana jika pengunjung berusia 18 mereka akan pergi ke konser Pink Floyd daripada menghadiri konser ACDC atau Meatloaf. Orang dengan usia 18 tahun memasuki area, dan karena tidak lebih dari 18 tahun, mereka tidak dapat melihat ACDC, terkecuali mereka 18 tahun, mereka menghadiri konser Pink Floyd. Setelah melihat Pink Floyd, mereka berjalan. Sintaks dari pernyataan `elif` mirip dengan `if` hanya berubah sedikit dari `if` menjadi `elif`.

In [12]:

```
# Contoh pernyataan elif

age = 18

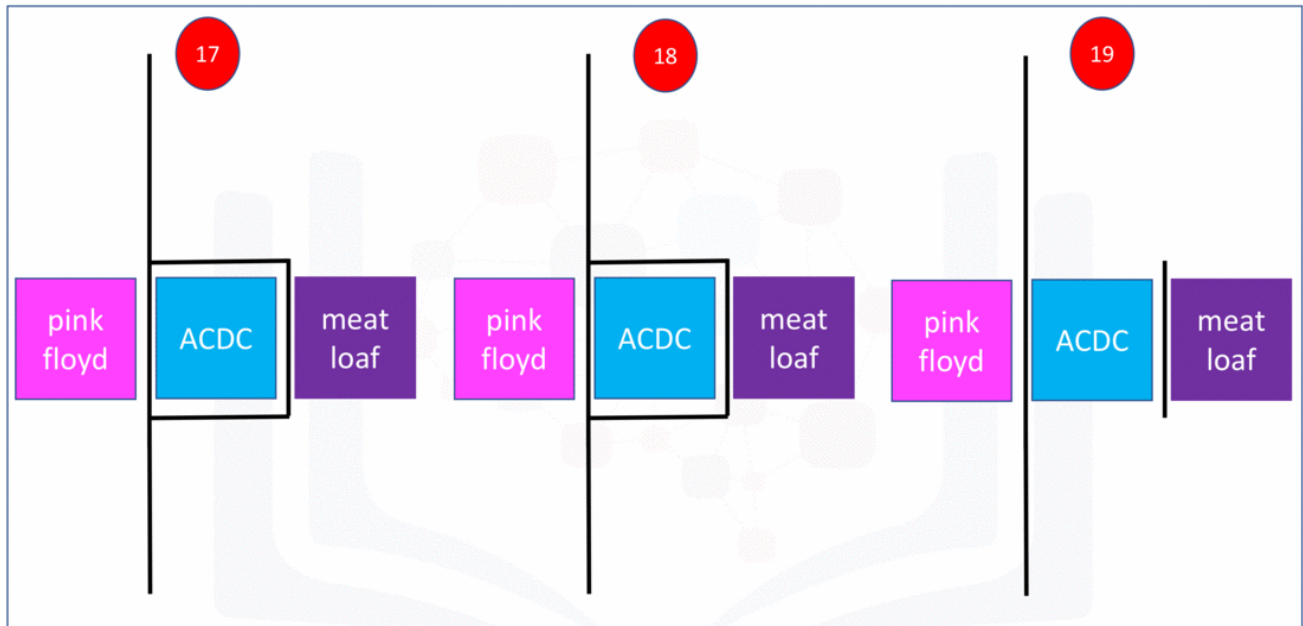
if age > 18:
    print("you can enter" )
elif age == 18:
    print("go see Pink Floyd")
else:
    print("go see Meat Loaf" )

print("move on")

go see Pink Floyd
move on
```



Kombinasi dari ketiganya ditunjukkan pada gambar dibawah. Sisi paling kiri menunjukkan apa yang terjadi ketika pengunjung kurang dari 18 tahun. Sisi tengah menunjukkan ketika pengunjung tepat berusia 18. Sisi ujung kanan menunjukkan ketika pengunjung lebih dari 18 tahun.



Lihat baris kode berikut :

In [13]:

```
# Contoh pernyataan kondisi

album_year = 1983
album_year = 1970

if album_year > 1980:
    print("Album year is greater than 1980")

print('do something..')
```

do something..

Ubahlah nilai dari `album_year` menjadi nilai lainnya -- Anda akan lihat perubahan hasilnya

Perhatikan kode yang berada di blok indentasi akan hanya di eksekusi jika hasilnya adalah **True**

Seperti sebelumnya, kita bisa menambahkan pernyataan `else` ke dalam pernyataan `if`. Kode dalam `else` akan hanya di eksekusi jika hasilnya adalah **False**

**Sintaks:**

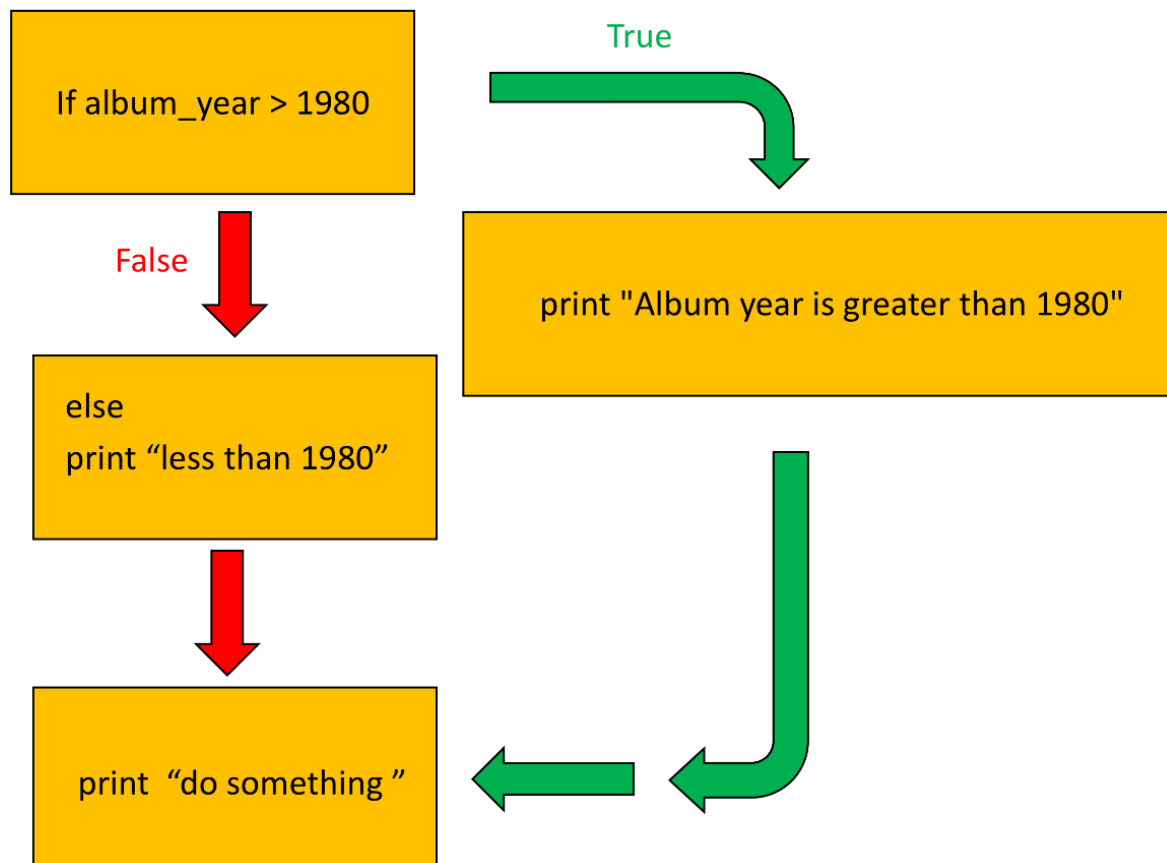
`if (condition):`

```
# do something
```

`else:`

```
# do something else
```

Jika kondisi dalam pernyataan `if` adalah **False**, pernyataan setelah `else` akan di eksekusi. Lihat gambar dibawah:



In [14]:

```
# Contoh pernyataan kondisi

album_year = 1983
#album_year = 1970

if album_year > 1980:
    print("Album year is greater than 1980")
else:
    print("less than 1980")

print('do something..')
```

```
Album year is greater than 1980
do something..
```

Ubahlah nilai dari `else` menjadi nilai lainnya -- Anda akan lihat perubahan hasilnya!

## Logical operators

Terkadang Anda ingin untuk mengecek lebih dari satu kondisi. Sebagai contoh, mungkin Anda ingin untuk cek jika satu kondisi dan kondisi lainnya adalah **True**. Operator logika membolehkan Anda untuk menggabungkan atau memodifikasi kondisi ini.

- `and`
- `or`
- `not`

Hasil dari operasi logika tersebut disajikan dalam *truth table* berikut

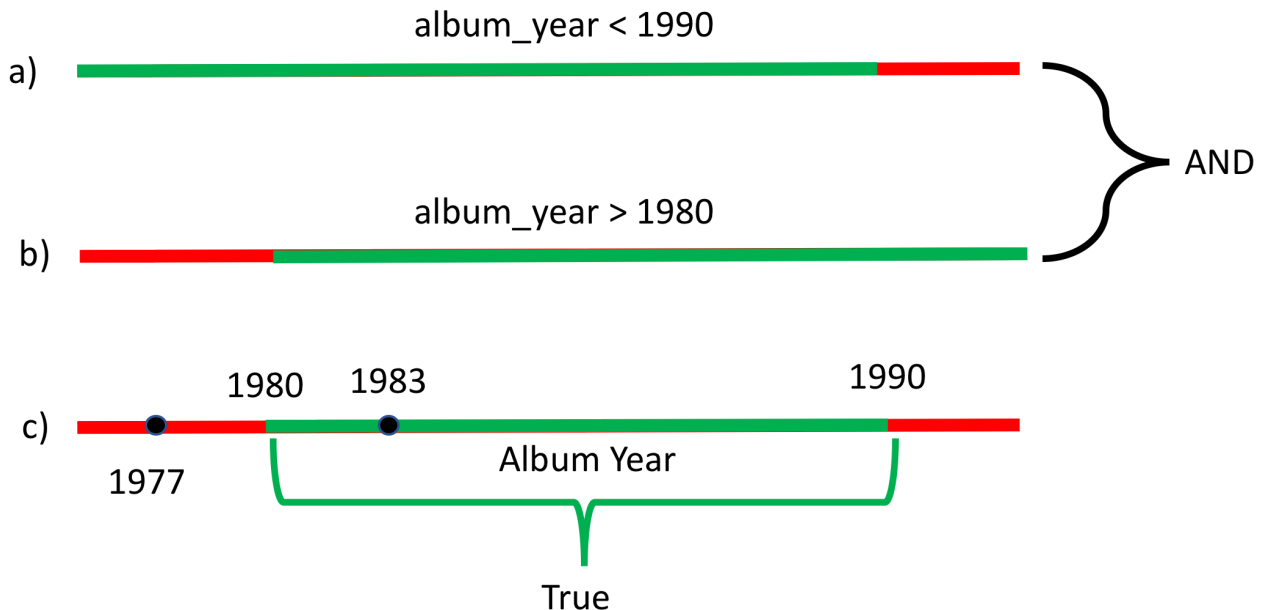
A	B	A & B
False	False	False
False	True	False
True	False	False
True	True	True

A	B	A or B
False	False	False
False	True	True
True	False	True
True	True	True

A	A!
False	True
True	False

Operasi `and` hanya bernilai **True** jika kedua kondisi bernilai benar. Operasi `or` bernilai benar jika salah satu kondisi bernilai **True**. Pernyataan `not` menghasilkan negasi dari kondisi sebenarnya.

Mari kita coba untuk menentukan apakah suatu album dirilis setelah 1979 (tahun 1979 tidak termasuk) dan sebelum 1990 (tahun 1990) tidak termasuk. Periode antara 1980 dan 1989 akan memenuhi kondisi tersebut, dan ditampilkan pada gambar di bawah ini. Warna hijau pada garis **a** dan **b** menggambarkan periode yang menghasilkan pernyataan bernilai **True**. Warna hijau pada garis **c** menggambarkan daerah dimana kedua kondisi bernilai **True**, yang merupakan perpotongan daerah **a** dan **b** yang bernilai **True**.



Kode program untuk melakukan pengecekan adalah sebagai berikut:

In [15]:

```
# Contoh kondisi
album_year = 1980

if(album_year > 1979) and (album_year < 1990):
    print ("Album year was in between 1980 and 1989")

print("")
print("Do Stuff..")
```

Album year was in between 1980 and 1989

Do Stuff..

Ilustrasi di bawah ini menunjukkan kondisi apakah sebuah album dirilis sebelum 1980 atau setelah 1989. Warna hijau pada garis **a** menunjukkan periode sebelum 1980 sedangkan warna hijau pada garis **b** menunjukkan periode setelah 1989. Garis berwarna hijau pada **c** menunjukkan kondisi jika salah satu dari **a** atau **b** bernilai **True**.



Kode untuk melakukan pengecekan adalah sebagai berikut:

In [16]:

```
# Contoh seleksi kondisi

album_year = 1990

if(album_year < 1980) or (album_year > 1989):
    print ("Album was not made in the 1980's")
else:
    print("The Album was made in the 1980's ")
```

Album was not made in the 1980's

Pernyataan not digunakan untuk mengecek apakah suatu pernyataan bernilai salah

In [17]:

```
# Contoh seleksi kondisi

album_year = 1983

if not (album_year == '1984'):
    print ("Album year is not 1984")
```

Album year is not 1984

## Kuis mengenai Kondisi

Tulislah sebuah pernyataan untuk menentukan apakah sebuah album memiliki rating lebih dari 8. Ujilah menggunakan album **Back in Black** yang memiliki rating sebesar 8,5. Jika pernyataan bernilai benar, tampilkan kalimat "This album is Amazing!"

In [5]:

```
# Tulis kode anda dibawah ini dan tekan Shift+Enter untuk melakukan eksekusi
back_in_black = 8.5

if back_in_black > 8:
    print ("This Album is Amazing")

else:
    print ("This Album is Not Amazing")

print ("Do Something")
```

This Album is Amazing  
Do Something

Double-click **disini** untuk melihat jawabannya.

---

Tulislah pernyataan if-else yang melakukan operasi berikut: Jika rating lebih dari 8 tampilkan "this album is amazing", jika rating kurang dari atau sama dengan 8 tampilkan "this album is ok"

In [6]:

```
# Tulis kode anda dibawah ini dan tekan Shift+Enter untuk melakukan eksekusi
album = 10

if album > 8:
    print("This Album is Amazing")
elif album <= 8:
    print("This Album is Good")
```

This Album is Amazing

Double-click **disini** untuk melihat jawabannya.

---

Write an if statement to determine if an album came out before 1980 or in the years: 1991 or 1993. If the condition is true print out the year the album came out.

Tulislah sebuah pernyataan `if` untuk menentukan apakah sebuah album dirilis sebelum tahun 1980 atau pada salah satu tahun berikut : 1991 atau 1993. Jika kondisi tersebut terpenuhi, tampilkan tahun rilis dari album tersebut

In [7]:

```
# Tulis kode anda dibawah ini dan tekan Shift+Enter untuk melakukan eksekusi
album_year = 1980

if(album_year > 1980 ) or (album_year < 1993):
    print ("Album was not made in the 1970")
else:
    print ("Album was made in the 1970 ")
```

Album was not made in the 1970

Double-click **disini** untuk melihat jawabannya.

## About the Authors:

[Joseph Santarcangelo](https://www.linkedin.com/in/joseph-s-50398b136/) (<https://www.linkedin.com/in/joseph-s-50398b136/>) is a Data Scientist at IBM, and holds a PhD in Electrical Engineering. His research focused on using Machine Learning, Signal Processing, and Computer Vision to determine how videos impact human cognition. Joseph has been working for IBM since he completed his PhD.

Other contributors: [Mavis Zhou](https://www.linkedin.com/in/jiahui-mavis-zhou-a4537814a/) ([www.linkedin.com/in/jiahui-mavis-zhou-a4537814a](https://www.linkedin.com/in/jiahui-mavis-zhou-a4537814a/))

---

Copyright © 2018 IBM Developer Skills Network. This notebook and its source code are released under the terms of the [MIT License](https://cognitiveclass.ai/mit-license/) (<https://cognitiveclass.ai/mit-license/>).