



# Python Enthusiast

```
bootcamp = ["ParselMouth", "PythonEnthusiast"]
>> for i in bootcamp:
>>     if i:
>>         print("Hello {}".format(i))
...
>> print("let's learn Python Characteristic")
```

## Daftar Isi

<b>1. Environment .....</b>	<b>2</b>
<b>2. Python Characteristic.....</b>	<b>2</b>
01. Easy .....	3
a. Easy to Code.....	3
b. Easy to Read.....	3
02. Expressive .....	3
03. Free and Open Source .....	3
04. High Level .....	4
05. Portable .....	4
06. Object Oriented .....	4
07. Extensible .....	4
08. Embeddable .....	4
09. Interpreted .....	4
10. Large Standard Library .....	5
11. GUI (General User Interface) .....	5
12. Dynamically Typed .....	5
<b>3. The key differences between Python 2.7.x and Python 3.x .....</b>	<b>5</b>

## 1. *Environment*

Sebelum memulai untuk menciptakan sebuah program, seorang *programmer* harus mempersiapkan beberapa kelengkapan demi mendukung kelancaran dalam penulisan dan penyusunan kode. Kelengkapan tersebut diantaranya bisa berupa *editor*, *compiler* maupun *library* eksternal lain yang berhubungan langsung dengan kebutuhan program, Biasanya kelengkapan tersebut disebut sebagai *environment*.

Sementara itu, di dalam bahasa pemrograman Python sudah tersedia IDE (*Integrated Development Environment*) *powerful* demi terciptanya *Environment* yang sesuai, IDE tersebut adalah Anaconda, didalam Anaconda kita bisa memanfaatkan *library built-in* (*library* yang sudah terinstall) sehingga untuk keperluan kebutuhan standar (secara umum) kita tidak diharuskan untuk melakukan installasi *library* tersebut. Kebutuhan standar yang perlu dipersiapkan untuk bisa memulai pemrograman Python diantaranya adalah sebagai berikut:

- a. *Virtual Environment*
- b. *Python Interpreter*
- c. *Editor*

## 2. *Python Characteristic*

Sebelum memulai penulisan kode Bahasa Pemrograman Python, Seorang Python *Entusiast* harus memahami beberapa karakter dasar bahasa pemrograman Python, dikarenakan Python adalah bahasa pemrograman yang *clear and powerful* pada bahasa pemrograman dengan orientasi objek. Diantaranya seperti bahasa pemrograman Perl, Ruby, Scheme, ataupun Java. Detail mengenai fitur dan karakteristik bahasa pemrograman Python dikutip dari data-flair.training adalah sebagai berikut:



## 01. *Easy*

Python merupakan bahasa pemrograman mudah untuk ditulis dan dibaca.

### a. *Easy to Code*

Bahasa pemrograman Python merupakan bahasa pemrograman yang mudah untuk ditulis dibandingkan dengan bahasa pemrograman C++ ataupun Java. Seorang *programmer* Python dapat menuliskan kode program dengan waktu yang lebih sedikit dikarenakan penulisan kode Python yang terlalu singkat, sehingga tidak banyak menghabiskan baris kode, seperti pada penggunaan *begin* dan *end* (mulai dan selesainya block baris kode), di dalam bahasa pemrograman Python cukup menggunakan indentasi 4 kali ketukan spasi atau satu kali tabulasi, akan tetapi harus konsisten. Disamping penulisan kode yang singkat, *programmer* Python harus bisa memahami cara penggunaan library, karena dalam bahasa pemrograman Python tersedia banyak paket dan modul.

### b. *Easy to Read*

Meskipun bahasa pemrograman Python merupakan bahasa pemrograman *High Level*, Bahasa Pemrograman Python tersebut mencoba untuk beradaptasi seperti penulisan bahasa alami manusia, yaitu dengan mengakomodasi indentasi pada penulisan teks secara abjad sehingga membentuk suatu kalimat/paragraf. Sehingga dalam pembacaan kode lebih dimudahkan dengan melakukan identifikasi pada indentansi tersebut.

## 02. *Expressive*

*Expressiveness* adalah kemudahan *programmer* untuk berekspresi ke dalam *block* kode, seperti halnya *programmer* dimudahkan dalam menggunakan *dictionary* dan *list* secara bersamaan di dalam Python dengan hanya menggunakan beberapa *line code*. Sedangkan apabila menggunakan bahasa pemrograman C++, *programmer* tersebut akan sedikit kesulitan dengan diperlukannya banyak *line code* yang melebihi *line code* implementasi dari bahasa Python sebelumnya.

## 03. *Free and Open Source*

Sudah jelas, bahwa bahasa pemrograman Python merupakan bahasa pemrograman yang bersifat *open source* dan banyak komunitas yang mendukung terkait pengembangan.

#### 04. High Level

Bahasa Pemrograman Python merupakan bahasa tingkat tinggi, yang dimana lebih dekat dengan bahasa alami manusia. Sehingga untuk dapat menuliskan program secara umum *programmer* tidak dituntut untuk menyediakan *memory*, menyediakan arsitektur system yang langsung berkaitan dengan perangkat keras.

#### 05. Portable

Bahasa pemrograman Python dapat dijalankan secara *cross platform* (sistem operasi yang bervariasi). Apabila menuliskan kode di sistem operasi mac atau windows, kode tersebut tentu masih bisa dijalankan pada sistem operasi linux.

#### 06. Object Oriented

Sudah jelas, pada bahasa pemrograman Python mengakomodasi paradigma pemrograman dengan Orientasi Objek, dengan keuntungan penambahan paradigma pemrograman secara *functional programming* yaitu dengan disediakannya *lambda function*.

#### 07. Extensible

Bahasa pemrograman Python dapat dikembangkan ke dalam bahasa pemrograman yang lain, seperti halnya kedalam bahasa pemrograman C++. *Programmer* dapat melakukan *import module* didalamnya berisi kode dengan bahasa pemrograman Python ke dalam kode dengan penulisan bahasa pemrograman C++.

#### 08. Embeddable

Seperti poin 07, akan tetapi secara kebalikannya. Bahasa pemrograman Python dapat melakukan *importing module* yang ditulis dari bahasa pemrograman lain, contohnya adalah bahasa pemrograman C++.

#### 09. Interpreted

Berbeda dengan bahasa pemrograman C++ dan Java. Bahasa Pemrograman Python merupakan bahasa pemrograman dengan eksekusi tanpa melakukan *compiling* terlebih dahulu yang biasa dilakukan oleh C++ dengan gcc dan Java dengan javac. Pada bahasa pemrograman Python, kode Python akan dieksekusi dari baris ke baris, sehingga memudahkan *programmer* untuk melakukan *error tracing* pada kode yang telah ditulis.

### 10. *Large Standard Library*

Python memiliki support banyak *library*, untuk *data science* seperti *image processing*, *web programming*, *IOT*, *regular expressions*, *threading*, *CGI*, *big data* dan masih banyak lagi

### 11. *GUI (General User Interface)*

Telah disediakan beberapa *library* standar untuk dapat menggunakan GUI tersebut, mulai dari skala primitive hingga enterprise. Contohnya adalah Tk, Qt dan masih banyak lagi.

### 12. *Dynamically Typed*

Merupakan salah satu karakteristik bahasa pemrograman Python yang paling disukai. Dengan mengakomodasi *dynamically typed*, programmer menjadi lebih efisien dalam menuliskan kode. Karena *programmer* tidak perlu mendeklarasikan tipe data untuk bisa menggunakan variable tersebut. Kembali ke poin 09, Python merupakan bahasa pemrograman *Interpreted*, sehingga pengecekan tipe nilai dilakukan saat *running time* sehingga (tidak diperlukan compiling).

## 3. *The key differences between Python 2.7.x and Python 3.x*

Beberapa perubahan dan penambahan yang diterapkan pada Python 3.x menyebabkan kode program yang ditulis menggunakan Python 2.x tidak bisa dijalankan pada Python 3.x secara langsung. Karena Python merupakan bahasa pemrograman *open source* dan memiliki anggota komunitas yang besar, sebenarnya perubahan dan penambahan mengenai hal yang berkaitan dengan bahasa pemrograman Python dapat dilihat secara langsung pada <https://www.python.org/dev/peps/>. Di dalam web tersebut dijelaskan beberapa perubahan dan penambahan pada pembahasan PEPs (*Python Enhancement Proposals*). Secara garis besar perubahan yang ada Python 3.x adalah sebagai berikut:

feature	optional in	mandatory in	effect
nested_scopes	2.1.0b1	02.02	<a href="#">PEP 227: Statically Nested Scopes</a>
generators	2.2.0a1	02.03	<a href="#">PEP 255: Simple Generators</a>
division	2.2.0a2	03.00	<a href="#">PEP 238: Changing the Division Operator</a>

absolute_import	2.5.0a1	03.00	<a href="#">PEP 328: Imports: Multi-Line and Absolute/Relative</a>
with_statement	2.5.0a1	02.06	<a href="#">PEP 343: The “with” Statement</a>
print_function	2.6.0a2	03.00	<a href="#">PEP 3105: Make print a function</a>
unicode_literals	2.6.0a2	03.00	<a href="#">PEP 3112: Bytes literals in Python 3000</a>

Dilihat dari tabel PEPs diatas yang diambil dari [blog](#) terlihat 4 *mandatory* (wajib) penerapan pada Python 3.x diantaranya adalah :

- Perubahan pada operator pembagian
- Penyesuaian pada *Import module*
- Penyesuaian pada penggunaan *print*
- Penyesuaian *Bytes Literal* pada Python