

modul modul pembelajaran
VedulPro

BELAJAR PYTHON 11



Fungsi Python

Fungsi adalah blok kode terorganisir dan dapat digunakan kembali yang digunakan untuk melakukan sebuah tindakan/action. Fungsi memberikan modularitas yang lebih baik untuk aplikasi Anda dan tingkat penggunaan kode yang tinggi.

Mendefinisikan Fungsi Python

Anda dapat menentukan fungsi untuk menyediakan fungsionalitas yang dibutuhkan. Berikut adalah aturan sederhana untuk mendefinisikan fungsi dengan Python.

- Fungsi blok dimulai dengan def kata kunci diikuti oleh nama fungsi dan tanda kurung ().
- Setiap parameter masukan atau argumen harus ditempatkan di dalam tanda kurung ini. Anda juga dapat menentukan parameter di dalam tanda kurung ini.
- Pernyataan pertama dari sebuah fungsi dapat berupa pernyataan opsional - string dokumentasi fungsi atau docstring.
- Blok kode dalam setiap fungsi dimulai dengan titik dua (:) dan indentasi.
- Pernyataan kembali [ekspresi] keluar dari sebuah fungsi, secara opsional menyampaikan kembali ekspresi ke pemanggil. Pernyataan pengembalian tanpa argumen sama dengan return None.

Contoh fungsi

```
def printme( str ) :  
    "This prints a passed string into this function"  
    print (str)  
    return
```

Modul Python

Modul memungkinkan Anda mengatur kode Python secara logis. Mengelompokkan kode terkait ke dalam modul membuat kode lebih mudah dipahami dan digunakan. Modul adalah objek Python dengan atribut yang diberi nama yang bisa Anda bind dan dijadikan referensi.

Secara sederhana modul adalah file yang terdiri dari kode Python. Modul dapat mendefinisikan fungsi, kelas dan variabel. Modul juga bisa menyertakan kode yang bisa dijalankan "runable".

Berikut adalah contoh modul sederhana pada Python :

```
def print_func( par ) :  
    print "Halo : ", par  
    return
```

Import Statement

Anda dapat menggunakan file sumber Python apapun sebagai modul dengan mengeksekusi pernyataan impor di file sumber Python lainnya. Impornya memiliki sintaks berikut.

Ketika interpreter menemukan sebuah pernyataan import, ia mengimpor modul jika modul tersebut ada di jalur pencarian. Jalur pencarian adalah daftar direktori yang ditafsirkan juru bahasa sebelum mengimpor modul. Misalnya, untuk mengimpor modul hello.py, Anda perlu meletakkan perintah berikut di bagian atas script.

```
# Import module support

import support

# Anda bisa memanggil fungsi defined sebagai berikut

support.print_func("Andy")
```

File I/O Python

Disini kita akan belajar semua fungsi dasar I/O yang tersedia pada Python 3. Jika Anda ingin mempelajari lebih detail, lihat dokumentasi standar Python.

Print

Cara termudah untuk menghasilkan output adalah dengan menggunakan pernyataan cetak di mana Anda bisa melewati nol atau lebih banyak ekspresi yang dipisahkan dengan koma. Fungsi ini mengubah ekspresi yang Anda berikan ke string dan menulis hasilnya ke output standar sebagai berikut :

```
print ("Python adalah bahasa pemrograman yang hebat")
```

Membaca Input Keyboard

Python 2 memiliki dua fungsi built-in untuk membaca data dari input standar, yang secara default berasal dari keyboard. Fungsi ini adalah input() dan raw_input()

Dengan Python 3, fungsi raw_input() tidak digunakan lagi. Selain itu, input() berfungsi membaca data dari keyboard sebagai string, terlepas dari apakah itu tertutup dengan tanda kutip (" atau ") atau tidak.

Fungsi Input Python

Fungsi input([prompt]) setara dengan raw_input, kecuali mengasumsikan bahwa input adalah ekspresi Python yang valid dan mengembalikan hasil yang dievaluasi ke Anda.

```
>>> x = input("something:")
```

```

>>> something:10

>>> x
>>> '10'

>>> x = input("something:")
>>> something:'10' #entered data treated as string with or without '

>>> x
>>> "'10'"

```

Exception Python

Python menyediakan dua fitur yang sangat penting untuk menangani kesalahan tak terduga dalam program Python Anda dan menambahkan kemampuan debugging di dalamnya.

- **Exception Handling**
- **Assertions** Exception adalah sebuah peristiwa, yang terjadi selama pelaksanaan program yang mengganggu aliran normal instruksi program. Secara umum, ketika skrip Python menemukan situasi yang tidak dapat diatasi, hal itu menimbulkan pengecualian. Exception adalah objek Python yang mewakili kesalahan.

Ketika skrip Python menimbulkan Exception, ia harus menangani Exception begitu saja sehingga berhenti dan berhenti.

Standard Exceptions

Nama	Penjelasan
<code>Exception</code>	Kelas dasar untuk semua pengecualian / exception
<code>StopIteration</code>	Dibesarkan ketika metode (iterator) berikutnya dari iterator tidak mengarah ke objek apa pun.
<code>SystemExit</code>	Dibesarkan oleh fungsi <code>sys.exit ()</code> .
<code>StandardError</code>	Kelas dasar untuk semua pengecualian built-in kecuali <code>StopIteration</code> dan <code>SystemExit</code> .
<code>ArithmeticError</code>	Kelas dasar untuk semua kesalahan yang terjadi untuk perhitungan numerik.
<code>OverflowError</code>	Dibesarkan saat perhitungan melebihi batas maksimum untuk tipe numerik.

<code>FloatingPointError</code>	Dibesarkan saat perhitungan floating point gagal.
<code>ZeroDivisonError</code>	Dibesarkan saat pembagian atau modulo nol dilakukan untuk semua tipe numerik.
<code>AssertionError</code>	Dibesarkan jika terjadi kegagalan pernyataan Assert.
<code>AttributeError</code>	Dibesarkan jika terjadi kegagalan referensi atribut atau penugasan.
<code>EOFError</code>	Dibesarkan bila tidak ada input dari fungsi <code>raw_input ()</code> atau <code>input ()</code> dan akhir file tercapai.
<code>ImportError</code>	Dibesarkan saat sebuah pernyataan impor gagal.
<code>KeyboardInterrupt</code>	Dibesarkan saat pengguna menyela eksekusi program, biasanya dengan menekan Ctrl + c.
<code>LookupError</code>	Kelas dasar untuk semua kesalahan pencarian.
<code>IndexError</code>	Dibesarkan saat sebuah indeks tidak ditemukan secara berurutan.
<code>KeyError</code>	Dibesarkan saat kunci yang ditentukan tidak ditemukan dalam kamus.
<code>NameError</code>	Dibesarkan saat pengenalan tidak ditemukan di namespace lokal atau global.
<code>UnboundLocalError</code>	Dibesarkan saat mencoba mengakses variabel lokal dalam suatu fungsi atau metode namun tidak ada nilai yang ditugaskan padanya.
<code>EnvironmentError</code>	Kelas dasar untuk semua pengecualian yang terjadi di luar lingkungan Python.
<code>IOError</code>	Dibesarkan saat operasi input / output gagal, seperti pernyataan cetak atau fungsi <code>open ()</code> saat mencoba membuka file yang tidak ada.
<code>OSError</code>	Dibangkitkan untuk kesalahan terkait sistem operasi.

<code>SyntaxError</code>	Dibesarkan saat ada kesalahan dengan sintaks Python.
<code>IndentationError</code>	Dibesarkan saat indentasi tidak ditentukan dengan benar.
<code>SystemError</code>	Dibesarkan saat penafsir menemukan masalah internal, namun bila kesalahan ini ditemui juru bahasa Python tidak keluar.
<code>SystemExit</code>	Dibesarkan saat juru bahasa Python berhenti dengan menggunakan fungsi <code>sys.exit ()</code> . Jika tidak ditangani dalam kode, menyebabkan penafsir untuk keluar.
<code>TypeError</code>	Dibesarkan saat operasi atau fungsi dicoba yang tidak valid untuk tipe data yang ditentukan.
<code>ValueError</code>	Dibesarkan ketika fungsi bawaan untuk tipe data memiliki jenis argumen yang valid, namun argumen tersebut memiliki nilai yang tidak valid yang ditentukan.
<code>RuntimeError</code>	Dibesarkan saat kesalahan yang dihasilkan tidak termasuk dalam kategori apa pun.
<code>NotImplementedError</code>	Dibesarkan ketika metode abstrak yang perlu diimplementasikan di kelas warisan sebenarnya tidak dilaksanakan.