

Plagiarism Scan Report

Summary

Report Generated Date	27 Aug, 2017
Plagiarism Status	100% Unique
Total Words	564
Total Characters	4174
Any Ignore Url Used	

Content Checked For Plagiarism:

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

Menangani pengecualian

Jika Anda memiliki beberapa kode yang mencurigakan yang mungkin menimbulkan pengecualian, Anda dapat mempertahankan program Anda dengan menempatkan kode yang mencurigakan di coba: blokir. Setelah dicoba: blokir, sertakan sebuah pernyataan kecuali:, diikuti oleh blok kode yang menangani masalah ini seaman mungkin.

Sintaksis

Berikut adalah sintaks sederhana coba kecuali ... blok lain -

```
try:
    You do your operations here;
    .....
except ExceptionI:
    If there is ExceptionI, then execute this block.
except ExceptionII:
    If there is ExceptionII, then execute this block.
    .....
else:
    If there is no exception then execute this block
```

Berikut adalah beberapa poin penting tentang sintaks yang disebutkan di atas -

Pernyataan percobaan tunggal dapat memiliki banyak kecuali pernyataan. Ini berguna saat blok coba berisi pernyataan yang mungkin membuang berbagai jenis pengecualian.

Anda juga bisa memberikan klausa umum kecuali klausul, yang menangani pengecualian apapun.

Setelah klausa kecuali, Anda bisa memasukkan klausul lain. Kode di blok yang lain dijalankan jika kode di coba: blok tidak menimbulkan pengecualian.

Blok yang lain adalah tempat yang baik untuk kode yang tidak perlu dicoba: perlindungan blokir.

Contoh

Contoh ini membuka file, menulis konten di file, dan keluar dengan anggun karena tidak ada masalah sama sekali -

```
#!/usr/bin/python
```

```
try:
h = open("testfile", "w")
h.write("This is my test file for exception handling!!")
except IOError:
print "Error: can't find file or read data"
else:
print "Written content in the file successfully"
h.close()
```

Ini menghasilkan hasil sebagai berikut -

Written content in the file successfully

Klausul kecuali tanpa pengecualian

Anda juga dapat menggunakan pernyataan kecuali tanpa pengecualian yang didefinisikan sebagai berikut -

```
try:
You do your operations here;
.....
except:
If there is any exception, then execute this block.
.....
else:
If there is no exception then execute this block.
```

Pernyataan try-except semacam ini menangkap semua pengecualian yang terjadi. Dengan menggunakan jenis try-except statement ini tidak dianggap sebagai praktik pemrograman yang bagus, karena menangkap semua pengecualian namun tidak membuat programmer mengenali akar permasalahan yang mungkin terjadi.

Klausul Kecuali dengan Beberapa Pengecualian

Anda juga dapat menggunakan pernyataan kecuali yang sama untuk menangani beberapa pengecualian sebagai berikut -

```
try:
You do your operations here;
.....
except(Exception1[, Exception2[,...ExceptionN]]):
```

If there is any exception from the given exception list, then execute this block.

.....

else:

If there is no exception then execute this block.

```
#!/usr/bin/python
```

```
try:
```

```
h = open("testfile", "w")
```

```
h.write("This is my test file for exception handling!!")
```

```
finally:
```

```
print "Error: can't find file or read data"
```

```
#!/usr/bin/python
```

```
try:
```

```
h = open("testfile", "w")
```

```
try:
```

```
h.write("This is my test file for exception handling!!")
```

```
finally:
```

```
print "Going to close the file"
```

```
h.close()
```

```
except IOError:
```

```
print "Error: can't find file or read data"
```

Bila dikecualikan dilempar di blok coba, eksekusi langsung lolos ke blok akhirnya. Setelah semua pernyataan di blok akhirnya dieksekusi, pengecualian dinaikkan lagi dan ditangani dalam pernyataan kecuali jika ada di lapisan yang lebih tinggi dari pernyataan try-except. Argumen Eksepsi

Pengecualian dapat memiliki argumen, yang merupakan nilai yang memberi informasi tambahan tentang masalah tersebut. Isi argumen bervariasi menurut pengecualian. Anda menangkap argumen pengecualian dengan menyediakan sebuah variabel dalam klausa kecuali sebagai berikut -

```
try:
```

```
You do your operations here;
```

```
.....
```

```
except ExceptionType, Argument:
```

```
You can print value of Argument here...
```

Jika Anda menulis kode untuk menangani satu pengecualian, Anda dapat memiliki variabel mengikuti nama pengecualian dalam pernyataan kecuali. Jika Anda menjebak beberapa pengecualian, Anda dapat memiliki vari