30thsep23_pythonexceptionhandling00

April 16, 2024

[1]:
```
a=10
b=0
```

[2]:
```
a*b
```

[2]:
```
0
```

[3]:
```
a/b
```

```
---------------------------------------------------------------------------
ZeroDivisionError                         Traceback (most recent call last)
Cell In[3], line 1
----> 1 a/b

ZeroDivisionError: division by zero
```

[ ]:
```
a=int(input("enter a value for a"))
b=int(input("enter a value for b"))
```

[ ]:
```
a*b
```

[ ]:
```
a/b
```

[ ]:
```
a+b
```

[ ]:
```
a/b
print("print somthing")
```

[ ]:
```
a+b
```

[ ]:
```
l=[4,5,6,7,8,9,0]
l1=[]
def even_parser(l):
    for i in l:
        if i%2 == 0:
            l1.append(i)
    return l1
```

[ ]:

```
even_parser(l)
```

[ ]:

```
even_parser("sha")
```

[ ]:

```
even_parser(6)
print("print somthing")
```

[ ]:

```
even_parser(l)
print("print somthing")
```

Exception handling, we use try block

[ ]:

```
try:
    a=int(input("enter a vlue for a "))
    b=int(input("enter a value for b "))
    c=a/b
    print(c)
except:
    pass
```

[ ]:

```
try:
    a=int(input("enter a vlue for a "))
    b=int(input("enter a value for b "))
    c=a/b
    print(c)
except:
    pass
print("print something")
```

[ ]:

```
a=int(input("enter a value for a "))
```

[ ]:

```
try :
    c = 6/0
except Exception as e :
    print(e)
```

[ ]:

```
c=6/0
```

[ ]:

```
try :
    a=int(input())
    b=int(input())
    c=a/b
except Exception as e :
    print(e)
```

[ ]:

```
try :
    a=int(input())
    b=int(input())
    c=a/b
except TypeError  :
    print(t)
except ZeroDivisionError  :
    print(z)
```

[ ]:

```
try :
    a=int(input())
    b=int(input())
    c=a/b
except TypeError as t :
    print(t)
except ZeroDivisionError as z :
    print(z)
except ValueError as v :
    print(v)
```

[ ]:

```
try:
    x=5/0
except ZeroDivisionError as z:
    print(z)
    try :
        y=int("sha")
    except ValueError as v :
        print (v)
```

[ ]:

```
try:
    x=5/0
except ZeroDivisionError as z:
    print(z)
else:
    print("this will execute itself once try will execute without and error")
```

[ ]:

```
try:
    x=5/2
except ZeroDivisionError as z:
    print(z)
else:
    print("this will execute itself once try will execute without and error")
```

[ ]:

```
try:
    x=5/2
except ZeroDivisionError as z:
    print(z)
else:
    print("this will execute itself once try will execute without and error")
```

```
finally:
    print("this will be executed always")
```

[ ]:

```
try:
    x=5/0
except ZeroDivisionError as z:
    print(z)
else:
    print("this will execute itself once try will execute without and error")
finally:
    print("this will be executed always")
```

[ ]:

```
try:
    d={"name" : "sha", "mobile_no" :234523,"email_id":"sha@gmail.com"}
    d["course"]
except Exception :
    print("this is key error")
```

[ ]:

```
d
```

[ ]:

```
l=[1,2,3,44,5,5]
l[10]
```

[ ]:

```
try:
    l=[1,2,3,44,5,5]
    l[10]
except Exception as e:
    print(e)
```

[ ]:

```
a= "sha"
int(a)
```

[ ]:

```
try:
    a= "sha"
    int(a)
except Exception as e:
    print(e)
```

[ ]:

```
file_name="text.txt"
open(file_name)
f.read()
```

[ ]:

```
file_name="sample.txt"
f=open(file_name)
f.read()
```

[ ]:

```
ls -l
```

[ ]:
```
pwd#present working directory
```

[ ]:
```
file_name="/home/jovyan/work/sample.txt"
f=open(file_name)
f.read()
```

[ ]:
```
try :
    file_name ="D':/DataScience/Python Coding/Class/sample.txt"
    f=open(file_name)
    f.read()
except Exception as e:
    print(e)
```

[ ]:
```
try :
    file_name ="D':\\DataScience\\Python Coding\\Class\\sample.txt"
    f=open(file_name)
    f.read()
except Exception as e:
    print(e)
```

[ ]:
```
open('/home/jovyan/work/sample.txt')#just an example
```

[ ]:
```
cd pro
```

[ ]:
```
pwd
```

[ ]:
```
ls
```

[ ]:
```
cd..
```

[ ]:
```
cd ..
```

[ ]:
```
ls
```

[ ]:
```
n=int(input("enter a number"))
div=342/n
print(div)
```

[ ]:
```
try:
    n=int(input("enter a number"))
    div=342/n
    print(div)
except ValueError:
    print("user has entered invalid input")
except ZeroDivisionError:
    print("user has entered zero")
```

[ ]:

```
try :
    n = int(input("enter a number"))
    div = 342/n
    print(div)
except (ValueError,ZeroDivisionError):
    print("user has entere invalid input or may be zero")
```

[ ]:

```
try :
    try:
        n = int(input("enter a number"))
        div = 342/n
        print(div)
    except ZeroDivisionError:
        print("user has entered zero")
except ValueError:
    print("user has entered invalid input")
```

[ ]:

```
try :
    n = int(input("enter a number"))
    div = 342/n
    print(div)
except Exception :
    print("user has entere invalid input or may be zero")
```

[ ]:

```
try:
    n=int(input("entera number"))
    div =342/n
    print(div)
except Exception:
    print("user has entered invalid input or may be zzero")
else :
    print(div)
```

[ ]:

```
f=open("/home/jovyan/work/sample.txt")
f.read()
```

[ ]:

```
f
```

[4]:

```
import logging as log
```

[5]:

```
log.basicConfig(filename = 'sample.log')
```

[6]:

```
pwd
```

[6]:

```
'/home/jovyan/work'
```

[7]:

```
log.info("store a msg")
```

[8]:

```
log.basicConfig(filename = 'sample.log', level = log.INFO)
```

[9]:

```
log.basicConfig(filename = 'sample.log', level = log.INFO)
```

[10]:

```
log.debug("this is a debug msg")
```

[11]:

```
log.warning("this is my warning")
```

[12]:

```
log.error("this is a error msg")
```

[13]:

```
log.critical("this is a critical msg")
```

[15]:

```
import logging
logging.basicConfig(filename='trytest.log',level =logging.DEBUG)
try:
    logging.info("this is a start of my prog")
    logging.info("asking user to enter age")
    age = int(input("enter your age"))
    logging.info("user has entered the age")
    if age < 0 :
        logging.info("i am checking negative condition")
        raise valueError("user has entered a negative agewhich is not valid")
    elif age<18:
        logging.info("i am trying to check underage condition")
        raise Exception("user is underage")
    else :
        print("valid user")
except Exception as e :
    logging.error(e)
    print(e)
```

```
enter your age -852
```

```
name 'valueError' is not defined
```

[2]:

```
import logging
logging.basicConfig(filename='trytest.log',level =logging.DEBUG, format =␣
 ↪'%(asctime)s')
try:
    logging.info("this is a start of my prog")
    logging.info("asking user to enter age")
    age = int(input("enter your age"))
    logging.info("user has entered the age=%d", age)
    if age < 0 :
        logging.info("i am checking negative condition")
        raise valueError("user has entered a negative agewhich is not valid")
    elif age<18:
        logging.info("i am trying to check underage condition")
        raise Exception("user is underage")
    else :
        print("valid user")
except Exception as e :
    logging.error(e)
    print(e)
```

```
enter your age 89
```

```
valid user
```

[3]:

```
import logging
logging.basicConfig(filename='even_parse.log',level =logging.DEBUG, format =␣
 ↪'%(asctime)s- %(levelname)s- %(message) ')
l=[4,5,6,7,8,9,0]
logging.info(l)
l1=[]
def even_parser(l):
    try:
        logging.info("inside try block")
        for i in l:
            logging.info("inside for loop i=%d",i)
            if i%2 == 0:
                logging.info("checking the value of i")
                l1.append(i)
                logging.info("appending the value i = %d",i)
        return l1
    except Exception as e :
        logging.error(e)
```

[4]:

```
even_parser(l)
```

[4]:

```
[4, 6, 8, 0]
```

[ ]: