packages

package is also called collection of module

the purpose of packages id that to provide code re-usability across the folder/drivers.

a.create a package b.re-use the packages

a.i)create a folder.

```
   ii)place empty file __init__.py

   iii)write module name
```

b.i)by using sys module

syntax= sys.path.append('absolute path')

```
   ii)by using python environment var.]+
```

Exception Handling

runtime error of every program called exception.

To convert tech error messages into user friendly called exception handling.

the purpose of exception handling is that to build roburst (strong) application.

types of error

1.complie time error: occur due to not following syntax.

2.logical error: occur due to invalid logic.

3.runtime error: occur due to invalid input.

keyword for exception handling

1.try: exception monitoring block.

2.except: convert tech error into user friendly.

3.else:

4.finally:

5.raise:

various form of exception block

1.syntax 1

try:

block of code

except exception class name 1:

except exception class name n:

2.syntax 2

try:

3.syntax 3

try:

block of code

except exception class as alias name 1:

print(alias name 1)

except exception class as alia name n:

print(n alias name)

4.syntax 4

try:

block of statements

except: or exception as a

print(excetpion or ,a)

block of code

except (exception class1,exception class n):

user friendly messages

3.syntax 3

try:

block of code

except exception class as alias name 1:

print(alias name 1)

except exception class as alia name n:

print(n alias name)

4.syntax 4

try:

block of statements

except: or exception as a

print(excetpion or ,a)

In [ ]:

```
#hyd.py
class HyderabadDivisonError(Exception):pass
```

In [ ]:

```
#division.py

from hyd import HyderabadDivisonError
def devop(a,b):
       if (b==0):
              raise HyderabadDivisonError
       else:
              return (a/b)
```

In [ ]:

```
#divisiondemo.py

from division import devop
from hyd import HyderabadDivisonError
a=int(input('enterr your first value'))
b=int(input('enter your second value'))
try:
    res=devop(a,b)
except HyderabadDivisonError:
    print('do not enter zero for dem')
else:
    print('div ={}'.format(res))
```

In [ ]:

```
divex1.py

#program for accepting two values and find div.
#divex1.py
print('program execution started')
try:
    s1=input('enter your first value')
    s2=input('enter your second value')
    a=int(s1)
    b=int(s2)
    c=a/b
except ZeroDivisionError:
    print('don not enter zero decimal')
except ValueError:
    print('do not enter string ,alnums and symbols')
else:
    print('value of a ={}'.format(a))
    print('value of b ={}'.format(b))
    print('div =',c)
print('execution time completed')
```

In [ ]:

```
#divex2.py

print('program execution started')
try:
    s1=input('enter your first value')
    s2=input('enter your second value')
    a=int(s1)
    b=int(s2)
    c=a/b
except (ZeroDivisionError,ValueError) :
    print('do not enter zero for dem')
    print('do not enter alum,symbols,str')
else:
    print('value of a ={}, and b ={}'.format(a,b))
    print('division ={}'.format(c))
finally:
    print('i am from final block')
```

Loading [MathJax]/jax/output/CommonHTML/fonts/TeX/fontdata.js