In [14]:

```
x=-1
if x<0:
    raise Exception("sorry, no nmber  below  zero")
```

```
---------------------------------------------------------------------------
Exception                                 Traceback (most recent call last)
Input In [14], in <cell line: 2>()
      1 x=-1
      2 if x<0:
----> 3     raise Exception("sorry, no nmber  below  zero")

Exception: sorry, no nmber  below  zero
```

In [4]:

```
try:
    print(y)
except:
    print("something went wrong")
else:
    print("Nothing error")
finally:
    print("The 'try except' is finished")
print("shiva")
```

```
something went wrong
The 'try except' is finished
shiva
```

In [5]:

```
try:
    print(x)
except:
    print("something went wrong")
else:
    print("Nothing error")
finally:
    print("The 'try except' is finished")
print("shiva")
```

```
1
Nothing error
The 'try except' is finished
shiva
```

In [7]:

```
try:
    print(y)
except Exception as ob:
    print(ob)
else:
    print("Nothing error")
finally:
    print("The 'try except' is finished")
print("shiva")
```

```
name 'y' is not defined
The 'try except' is finished
shiva
```

In [10]:

```
try:
    f=open("shiv.txt")
    try:
        f.write("shiva  is very good boy")
    except:
        print("something went wrong when writng to the file")
    finally:
        f.close()
except:
    print("Something went wrong when opening the file")
```

```
something went wrong when writng to the file
```

In [11]:

```
try:
    f=open("shiv.txt",w)
    try:
        f.write("shiva  is very good boy")
    except:
        print("something went wrong when writng to the file")
    finally:
        f.close()
except:
    print("Something went wrong when opening the file")
```

```
Something went wrong when opening the file
```

In [13]:

```
try:
    f=open("shiv.txt",'w')
    try:
        f.write("shiva  is very good boy")
    except:
        print("something went wrong when writng to the file")
    finally:
        f.close()
        print("operation executed succesfully")
except:
    print("Something went wrong when opening the file")
```

```
operation executed succesfully
```

In [15]:

```
try:
    print(1 / 0)
except ZeroDivisionError as e:
    print(e)
    print(type(e))
```

```
division by zero
<class 'ZeroDivisionError'>
```

In [16]:

```
try:
    print(1 / 0)
except ZeroDivisionError:
    print('Error')
```

```
Error
```

You can also specify a base class. For example, ArithmeticError is the base class for ZeroDivisionError. The variable stores the exception object of the derived class that actually occurred.

In [17]:

```
print(issubclass(ZeroDivisionError, ArithmeticError))
# True

try:
    print(1 / 0)
except ArithmeticError as e:
    print(e)
    print(type(e))
```

```
True
division by zero
<class 'ZeroDivisionError'>
```

In [20]:

```
# Apply the same operation to multiple exceptions
def divide_same(a, b):
    try:
        print(a / b)
    except (ZeroDivisionError, TypeError) as e:
        print(e)
divide_same('a', 'b')
```

```
unsupported operand type(s) for /: 'str' and 'str'
```

In [22]:

```
# ignore exception
def divide_pass(a, b):
    try:
        print(a / b)
    except ZeroDivisionError:
        pass
```

In [ ]: