

```
In [ ]: ▶ # print(a)                :NameError

s= 'ali'
# print(s + 2)                : TypeError

# s.append('reza')            :AttributeError

lst = [15, 20, 17]
# print(Lst[3])                : IndexError

# print(Lst + 4)                : TypeError
# Lst.add(5)                    : AttributeError

d = {'a' : 5 , 'b' : 6}
# print(d['f'])                : KeyError

# x = 8/0                        : ZeroDivisionError
```

```
In [ ]: ▶ try:
        print(k)
except NameError:
    print('error')            # error
```

```
In [ ]: ▶ try:
        print(k)
except NameError as ne:
    print(ne)                # name 'k' is not defined
```

```
In [ ]: ▶ s = 'ali'

try:
    print(s + 2)
except TypeError as te:
    print(te)                # can only concatenate str (not "int") to str
```

```
In [ ]: ▶ try:
        x = 8 / 2
except ZeroDivisionError as ze:
    print(ze)
else:
    print(x)                # 4
```

```
In [ ]: ▶ def divide(x, y):
        try:
            r = x / y
        except ZeroDivisionError:
            print('error')
        else:
            print(r)
        finally:
            print('by')

divide(2, 1)      # 2.0    by
divide(4, 0)     # error  by
```

```
In [ ]: ▶ s = '23'
        try:
            i = int(s)
        except ValueError:
            i = -1
        print(i)          # 23
```

```
In [ ]: ▶ s = 'a'
        try:
            i = int(s)
        except ValueError:
            i = -1
        print(i)          # -1
```

```
In [ ]: ▶ def f(n):
        try:
            if n == 13:
                raise ValueError('unlucky number')
            return 20 / n
        except (ZeroDivisionError, TypeError):
            return 'Enter s number other than 0'

print(f(5))        # 4
print(f(0))        # Enter s number other than 0
print(f('a'))     # Enter s number other than 0
#print(f(13))     # unlucky number
```

```
In [ ]: ▶ # Nested try_except Blocks

try:
    print(5 / 0)
    try:
        print(n)
    except NameError as ne:
        print(ne)
except ZeroDivisionError as ze:
    print(ze)                                # division by zero
```

```
In [ ]: ▶ try:
    print(5 / 2)
    try:
        print(n)
    except NameError as ne:
        print(ne)
except ZeroDivisionError as ze:
    print(ze)
...
2.5
name 'n' is not defined
...
```

```
In [ ]: ▶ try :
    n = int(input('enter:'))
    assert n % 2 == 0
except:
    print('Not even')
else:
    print( n * 2)
```

دانشگاه شهید مدنی آذربایجان  
برنامه نویسی مقدماتی با پایتون  
امین گلزاری اسکویی  
۱۴۰۰-۱۴۰۱

[Codes and Projects \(click here\) \(https://github.com/Amin-Golzari-Oskouei/Python-Programming-Course-Basic-2021\)](https://github.com/Amin-Golzari-Oskouei/Python-Programming-Course-Basic-2021) slides and videos (click here) (https://drive.google.com/drive/folders/1ZsQjBJJ4UAAp9zrGxm3c4qrhmvGBUYHw)