try_exception_functions

June 28, 2021

1 Try_Exception

[1]:
```
12 / 0
```

```
---------------------------------------------------------------------------
ZeroDivisionError                         Traceback (most recent call last)
<ipython-input-1-df982fec2812> in <module>
----> 1 12 / 0

ZeroDivisionError: division by zero
```

[6]:
```
try:
    12 / 4
    print("          ")
except ZeroDivisionError:
    print("            ")

print("try-exception-             ")
```

```
try-exception-
```

[7]:
```
try:
    12 / 0
    print("          ")
except ZeroDivisionError:
    print("            ")

print("try-exception-             ")
```

```
try-exception-
```

[8]:
```
try:
    12 / 0
    print("          ")
except ZeroDivisionError as var1:
```

```
    print("            ")
    print(var1)

print("try-exception-             ")
```

```
division by zero
try-exception-
```

[9]:
```
a = input("       : ")
```

[10]:
```
type(a)
```

[10]:
```
str
```

[12]:
```
a = input("       : ")
print(type(a))
```

```
<class 'str'>
```

[13]:
```
b = int(32)
```

[14]:
```
type(b)
```

[14]:
```
int
```

[15]:
```
b = int('32')
type(b)
```

[15]:
```
int
```

[16]:
```
b = int(32.9)
```

[17]:
```
b
```

[17]:
```
32
```

[18]:
```
type(b)
```

[18]:
```
int
```

[19]:
```
b = int('fg')
```

```
---------------------------------------------------------------------------
ValueError                                Traceback (most recent call last)
<ipython-input-19-5d0acbb0e372> in <module>
----> 1 b = int('fg')
```

```
ValueError: invalid literal for int() with base 10: 'fg'
```

[20]:
```
try:
    a = input("         : ")
    a = int(a)
    12 / a
    print("          ")
except ZeroDivisionError as var1:
    print("             ")
    print(var1)
except ValueError:
    print("               ")

print("try-exception-             ")
```

```
try-exception-
```

[21]:
```
try:
    a = input("         : ")
    a = int(a)
    12 / a
    print("          ")
except ZeroDivisionError as var1:
    print("             ")
    print(var1)
except ValueError:
    print("               ")

print("try-exception-             ")
```

```
division by zero
try-exception-
```

[22]:
```
try:
    a = input("         : ")
    a = int(a)
    12 / a
    print("          ")
except ZeroDivisionError as var1:
    print("             ")
    print(var1)
except ValueError:
    print("               ")
```

```
print("try-exception-             ")
```

```
try-exception-
```

[23]:
```
try:
    a = input("         : ")
    a = int(a)
    12 / a
    print("           ")
except ZeroDivisionError as var1:
    print("             ")
    print(var1)
except ValueError:
    print("               ")
finally:
    print("                         ")

print("try-exception-             ")
```

```
division by zero

try-exception-
```

[24]:
```
try:
    a = input("         : ")
    a = int(a)
    12 / a
    print("           ")
except ZeroDivisionError as var1:
    print("             ")
    print(var1)
except ValueError:
    print("               ")
finally:
    print("                         ")

print("try-exception-             ")
```

```
try-exception-
```

[25]:
```
try:
    a = input("         : ")
    a = int(a)
```

```
    12 / a
    print("          ")
except ZeroDivisionError as var1:
    print("            ")
    print(var1)
except ValueError:
    print("               ")
finally:
    print("                         ")

print("try-exception-             ")
```

```
try-exception-
```

[26]:
```
try:
    a = int(input("         : "))

    12 / a
    print("          ")
except ZeroDivisionError as var1:
    print("            ")
    print(var1)
except ValueError:
    print("               ")
finally:
    print("                         ")

print("try-exception-             ")
```

```
try-exception-
```

[29]:
```
sum = 0
while True:
    try:
        a = input("                      n           :")

        if a == 'n':
            break
        else:
            a = int(a)
            sum += a
    except ValueError:
        print("            .")
```

```
        print("         ")
        continue

print("           ")
print(f"sum = {sum}")
```

.

.

```
sum = 170
```

2

[30]:
```
def test():
    print("Test            ")
```

[31]:
```
test()
```

```
Test
```

[32]:
```
test
```

[32]:
```
<function __main__.test()>
```

[33]:
```
type(test)
```

[33]:
```
function
```

[34]:
```
f = test
```

[35]:
```
f
```

[35]:
```
<function __main__.test()>
```

[36]:
```
f()
```

```
Test
```

[37]:
```
test()
```

```
Test
```

[38]:
```
l = [2, 3, 4, test, "spam"]
```

[39]:
```
l[3]
```

[39]:
```
<function __main__.test()>
```

[40]:
```
l[3]()
```

```
Test
```

[42]:
```
def double():

    x = 12
    x += 2

x = 90
double()
print(x)
```

```
90
```

[43]:
```
def double():
    global x
    x = 12
    x += 2

x = 90
double()
print(x)
```

```
14
```

[44]:
```
def double(x):
    x *= 2
    return x
```

[45]:
```
double(3)
```

[45]:
```
6
```

[46]:
```
a = double(4)
```

[47]:
```
a
```

[47]:
```
8
```

[48]:
```
def test3(a, b, c):
    return (a,b,c)
```

[49]:
```
test3(2,3,4)
```

[49]:
```
(2, 3, 4)
```

[51]:
```
lambda x : x*2
```

[51]:
```
<function __main__.<lambda>(x)>
```

[52]:
```
f = lambda x : x*2
```

[53]:
```
f(2)
```

[53]:
```
4
```

[ ]:
```

```