

## Исключения

- Исключение – это событие, после которого дальнейшее продолжение работы в данной точке бессмысленно. По итогу такого события генерируется объект-исключение, и исполнение передается обработчику ошибок этого типа
- Пример – деление на 0, выбрасывается ошибка `ZeroDivisionError`
- Исключения помогают упростить код, убрав из него множество проверок и значительно облегчить восстановление программы после сбоя
- Исключения упрощают доставку информации о ошибке от той точки, в которой она возникла к той точке где она может быть обработанна

## Исключения

```
1
2     try :
3         block1
4     except tp2 as var2:
5         block2
6     except (tp3 , tp4 ) as var3:
7         block3
8     else :
9         block5
10    finally :
11        block4
```

## Исключения

```
1  try :  
2      raise tp2("xxx")  # <<<<  
3  except tp2 as var2:  
4      block2            # <<<<  
5  except (tp3, tp4) as var3:  
6      block3  
7  else:  
8      block5  
9  finally:  
10     block4            # <<<<
```

## Исключения

```
1  try :  
2      pass                # <<<  
3  except tp2 as var2 :  
4      block2  
5  except (tp3 , tp4 ) as var3 :  
6      block3  
7  else :  
8      block5              # <<<  
9  finally :  
10     block4              # <<<
```

## Исключения

```
1  def f1(t, d, x, y):
2      if t - d == 0:
3          return None
4      else:
5          t1 = ((x + y) / (t - d))
6          if t1 == 0:
7              return None
8          else:
9              return 1 / ((x + y) / (t - d))
10
11 def f2(t, d, x, y):
12     try:
13         return 1 / ((x + y) / (t - d))
14     except ZeroDivisionError:
15         return None
```

## Исключения. raise

- `raise` `ExceptionType("Some_message")` порождает исключение
- `ExceptionType` должно наследовать `Exception`
- `raise` без параметров разрешено только в блоке `except`. Оно повторно выбрасывает исключение, которое сейчас обрабатывается

```
1     try :  
2         func()  
3     except Exception :  
4         print "func_cause_exception"  
5         raise
```

## Исключения. traceback

- В обработчике исключения `sys.exc_info()` возвращает тройку (Тип исключения, Объект исключения, Состояние Стекa)

```
1     try:
2         raise ValueError("ddd")
3     except Exception as x:
4         tb = sys.exc_info()[2]
5
6     print tb.tb_frame # <frame at 0x....>
7     print tb.tb_frame.f_lineno # 4
8     print tb.tb_frame.f_code.co_name # '<module>'
9     print tb.tb_frame.f_code.co_filename
10        # '<ipython-input-7-492d537cf800>'
11    print tb.tb_next # <frame at 0x....> or None
12    del tb
```