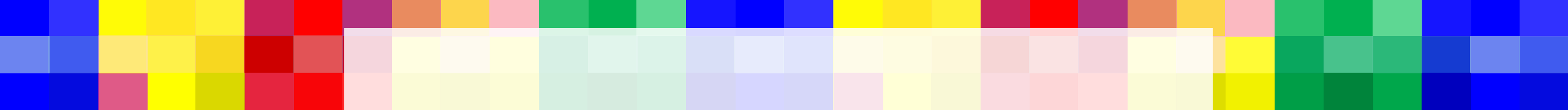




Ошибки, исключения



- 
- По умолчанию: остановка выполнения программы в тот момент, когда обнаружена ошибка
 - Примеры ошибок:
 - Синтаксические
 - Исключения

Исключения: Механизм информирования о произошедшем в коде



Исключения

- Встроенные
- Пользовательские

```
x = 15
if x > 10:
    raise Exception("x should not exceed 10")
```

- Оператор raise
 - Аргумент: экземпляр стандартного класса исключения/
пользовательского наследника от Exception

assert

```
x = "Python"
# если условие возвращает True, то ничего не происходит
assert x == "Python"

# если условие возвращает False, возникает ошибка AssertionError:
assert x == "Julia"
```

Применяется при отладке

```
x = "Python"

# если условие возвращает False, возникает ошибка AssertionError
assert x == "Julia", "x should be 'Python' "
```

Try-except

- Перехват и обработка исключений
- Без перехвата и обработки = сбой программы



A decorative border composed of a grid of small squares in various colors including blue, yellow, red, green, and purple, framing the top and bottom of the slide.

Try...except

Try...except

1. Исполнение блока try
2. Если нет исключения, except пропускается
3. Если возникло исключение, прекращается выполнение блока кода в try, выполняется блок кода, соответствующий типу исключения except, продолжение выполнения кода после конструкции try...except
4. Если в перечне except нет нужного типа исключений, выполнение передается во внешний блок try ... Если перехвачено не было – остановка хода выполнения скрипта

```
while True:
    try:
        x = int(input("Please enter a number: "))
        break
    except ValueError:
        print("Oops! That was no valid number. Try again...")
```

```
except (RuntimeError, TypeError, NameError):
```

Try...except...else

- Else
 - Указывается код, который будет выполнен в случае отсутствия исключений



Finally:

- Возникло или нет исключение,
- Этот блок будет выполнен
 - Повторный вызов исключения, если не было обработано в соответствующем блоке
 - Если исключение было вызвано блоком except – выполнение finally
 - Перед выполнением break, continue, return (вернется значение return -finally)



Summary

- Raise – вызывает исключение в любой момент
- Assert – генерирует исключение при невыполнении условия
- Try – последовательное выполнение, пока не обнаружено исключение
- Except – блок обработки исключения
- Else – выполнение кода, если в try не было исключений
- Finally – выполняется всегда, было исключение или нет

- 
- [8. Errors and Exceptions — Python 3.13.0 documentation](#)