

# Виключення

Обробка виключень - це механізм в програмуванні, який дозволяє вам зробити гнучке управління помилками та невластивими станами програми.

У Python обробка виключень використовує блоки `try`, `except`, `else` та `finally`. Ось загальна структура:

```
try:
    # Код, який може викликати виключення
except ExceptionType as e:
    # Обробка виключення типу ExceptionType
    # Об'єкт виключення доступний в змінній 'e'
else:
    # Викликається, якщо блок 'try' завершується без виключення
finally:
    # Викликається незалежно від того, чи виникло виключення чи ні
```

Використання:

1. **try**: В цьому блоку розміщується код, який може викликати виключення.
2. **except**: Якщо виникає виключення, програма шукає підходящий блок `except`. Якщо тип виключення відповідає типу, зазначеному в `except`, виконується код у цьому блоку. Об'єкт виключення доступний в змінній, яка зазвичай називається `e`.
3. **else**: Цей блок викликається, якщо жодного виключення не виникло у блоку `try`.
4. **finally**: Цей блок викликається завжди, незалежно від того, чи виникло виключення чи ні. Часто використовується для відключення ресурсів (наприклад, файлових дескрипторів), щоб гарантувати, що вони будуть відключені навіть при виникненні виключення.

Приклад:

```
try:
    x = int(input("Введіть число: "))
    result = 10 / x
except ZeroDivisionError as e:
```

```
print("Помилка: Ділення на нуль!")
result = None
else:
    print("Результат:", result)
finally:
    print("Операція завершена.")
```

У цьому прикладі, якщо користувач введе 0, виникне `ZeroDivisionError`. Код у блоку `except` буде виконано, а потім програма перейде до блоку `finally`.

Ця конструкція дозволяє уникнути "збоїв" програми при виникненні помилок та надає можливість коректно обробляти непередбачувані ситуації.

У Python існують різні типи виключень, які виникають у різних ситуаціях. Ось кілька основних типів виключень:

#### 1. **SyntaxError:**

- Виникає, коли виникає синтаксична помилка в програмному коді. Це означає, що код неможливо виконати через неправильний синтаксис.

#### 2. **IndentationError:**

- Специфічний тип `SyntaxError`, що виникає, коли неправильно використовуються пробіли або табуляція для відступів.

#### 3. **NameError:**

- Виникає, коли звертаються до змінної чи функції, яка не була визначена.

#### 4. **TypeError:**

- Виникає, коли виконується операція над об'єктом неправильного типу.

#### 5. **ValueError:**

- Виникає, коли функція отримує аргумент правильного типу, але невірне значення.

#### 6. **ZeroDivisionError:**

- Виникає, коли ви намагаєтесь поділити на нуль.

#### 7. **FileNotFoundError:**

- Виникає, коли намагаєтесь відкрити файл, який не існує.

#### 8. **IndexError:**

- Виникає, коли намагаєтесь звернутися до елементу списку за межами його діапазону.

#### 9. **KeyError:**

- Виникає, коли намагаєтесь отримати доступ до ключа словника, який не існує.

#### 10. **AttributeError:**

- Виникає, коли намагаєтесь отримати доступ до атрибуту об'єкта, який не існує.

#### 11. **ImportError:**

- Виникає, коли відбувається помилка при імпортуванні модуля.

#### 12. **KeyboardInterrupt:**

- Виникає, коли користувач натискає комбінацію клавіш для примусового завершення виконання програми (зазвичай `Ctrl+C`).

Це лише декілька прикладів типів виключень. У Python існує багато інших вбудованих та користувацьких типів виключень, які можуть виникати у ваших програмах. Корисно вміти правильно обробляти ці виключення для запобігання виходу програми зі строю через непередбачувані ситуації.