Сообщить об ошибке.

Система импорта в Python

Оператор import является наиболее распространенным способом вызова определений модуля, но это не единственный способ. Такие функции, как importlib.import_module() и встроенная функция __import__() также могут быть использованы для вызова механизма импорта.

Оператор import объединяет две операции: он выполняет поиск именованного модуля, а затем привязывает результаты этого поиска к имени в области видимости вызывающей стороны. Операция поиска оператором import определяется как вызов функции __import__() с соответствующими аргументами. Возвращаемое значение __import__() используется для выполнения операции связывания имени оператором import.

Прямой вызов функции __import__() выполняет только поиск модуля и, если он найден, операцию инициализации модуля. Только оператор import выполняет операцию привязки имени к пространству имен вызывающей стороны.

При первом импорте модуля Python выполняет поиск модуля и, если он найден, создает объект модуля, инициализируя его. Если именованный модуль не может быть найден, то вызывается исключение ModuleNotFoundError.

importlib

Модуль importlib предоставляет богатый API для взаимодействия с системой импорта. Например, importlib.import_module() предоставляет рекомендуемый, более простой API, чем встроенный __import__(), для вызова механизма импорта.

Спецификация инструкции import

Базовая инструкция `import`, без оператора `from`, выполняется в два этапа. Поиск модуля, загрузка и инициализация его при необходимости. Определение имени или имён, в локальном пространстве имен, для области, в которой выполняется инструкция import.

Определение модуля и его импорт в Python

Модуль - это файл, содержащий определения и операторы Python. Имя файла - это имя модуля с добавленным суффиксом `.py`. Внутри модуля, имя модуля, в виде строки, доступно в качестве значения глобальной переменной `__name__`.

Конструкция импорта import modulle as name

Содержимое модуля можно импортировать из определения функции. В этом случае импорт не выполняется до тех пор, пока не будет вызвана функция

Конструкция импорта from modulle import names

Альтернативная форма инструкции `import` позволяет импортировать отдельные объекты из модуля непосредственно в область видимости вызывающего объекта (другого модуля или скрипта)

Конструкция импорта from modulle import name as alt_name

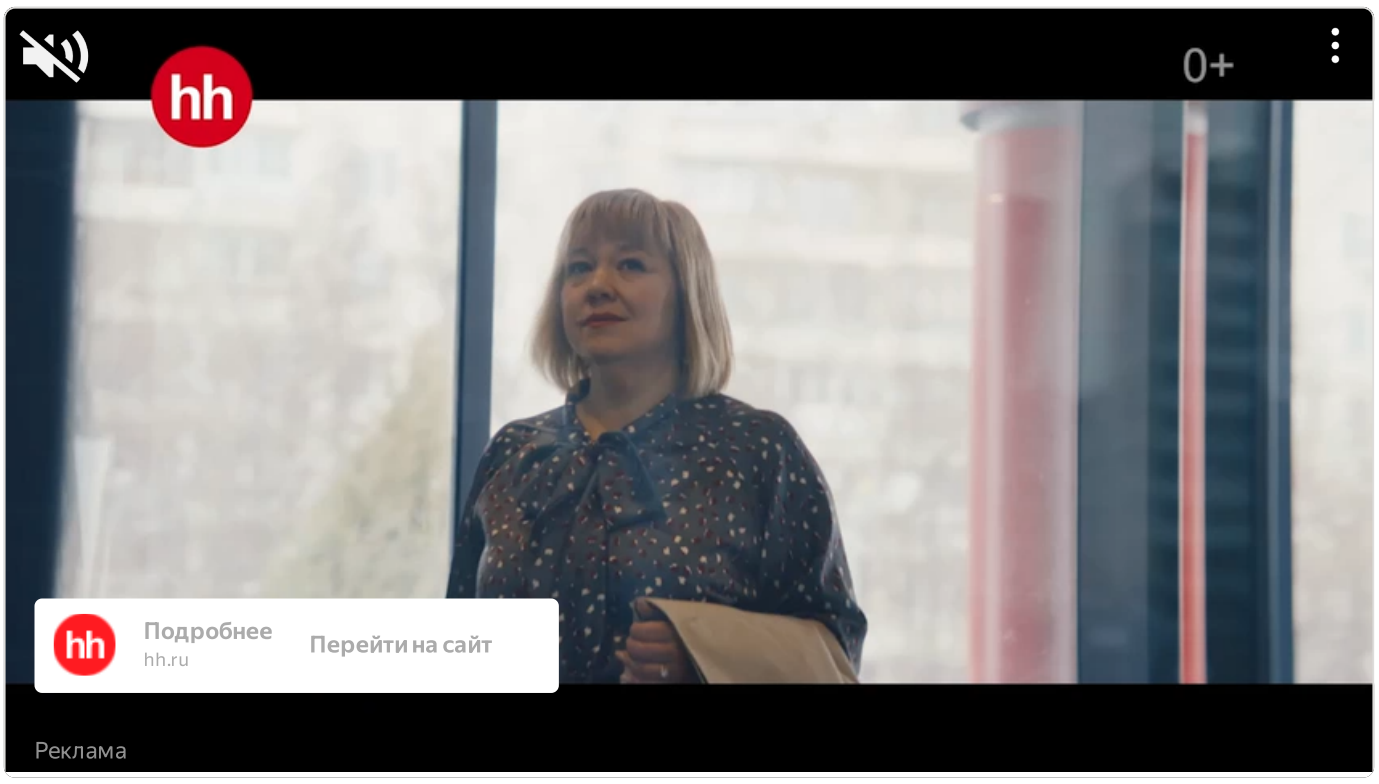
Можно импортировать отдельные объекты модуля, но ввести их в глобальную область видимости скрипта с альтернативными именами

Как Python ищет импортируемый модуль

Каталог, из которого был запущен скрипт. Список каталогов, содержащихся в переменной окружения `PYTHONPATH`. Список каталогов, настроенных во время установки Python.

Список имен, определенных в модуле Python

Встроенная функция `dir()` возвращает список определенных имен в пространстве имен. Если задан аргумент, который является именем модуля, `dir()` перечисляет имена, определенные в модуле.



Выполнение модуля как скрипта

Когда файл `.py` импортируется как модуль, Python устанавливает специальную переменную `__name__` в имя модуля. Однако, если файл выполняется как автономный сценарий, `__name__` устанавливается в строку `__main__`.

Перезагрузка модуля в Python

Если в модуль вносятся изменения то его нужно перезагружать, что бы изменения вступили в силу. Для этого нужно либо перезапустить интерпретатор, либо использовать функцию `importlib.reload()` из модуля `importlib`

Пакеты модулей в Python

Пакеты позволяют иерархически структурировать пространство имен модуля с использованием точечной нотации. Точно так же, как модули помогают избежать коллизий между именами глобальных переменных, пакеты помогают избежать коллизий между именами модулей

Файл пакета __init__.py

Если файл с именем __init__.py присутствует в каталоге пакета, то он вызывается при импорте пакета или модуля в пакете. Это может быть использовано для выполнения кода инициализации пакета, например инициализации данных уровня пакета.

Переменная __all__ в пакетах и модулях в Python

Если `__init__.py` файл в каталоге пакета содержит список с именем `__all__`, он считается списком модулей, которые должны быть импортированы при обнаружении инструкции `from <package> import *`.

Переменная пакета __path__ в Python

Атрибут пакета `__path__` используется при импорте его подпакетов. В механизме импорта он работает почти так же, как `sys.path` и предоставление списка местоположений для поиска модулей во время импорта.

Относительный импорт пакетов

Относительный импорт использует опережающие точки.

Вложенные подпакеты

Пакеты могут содержать вложенные подпакеты произвольной глубины. Для отделения имени пакета от имени вложенного пакета используется дополнительная точечная нотация

Пространства имен пакета

В настоящее время Python предоставляет pkgutil.extend_path для обозначения пространства имен пакета. `setuptools` предоставляет аналогичную функцию с именем `pkg_resources.declare_namespace`.

Настройка доступа к атрибутам модуля в Python

Специальные имена __getattr__ и __dir__ также могут использоваться для настройки доступа к атрибутам модуля. Функция __getattr__ на уровне модуля должна принимать один аргумент, который является именем атрибута, и возвращать вычисленное значение или вызывать AttributeError.