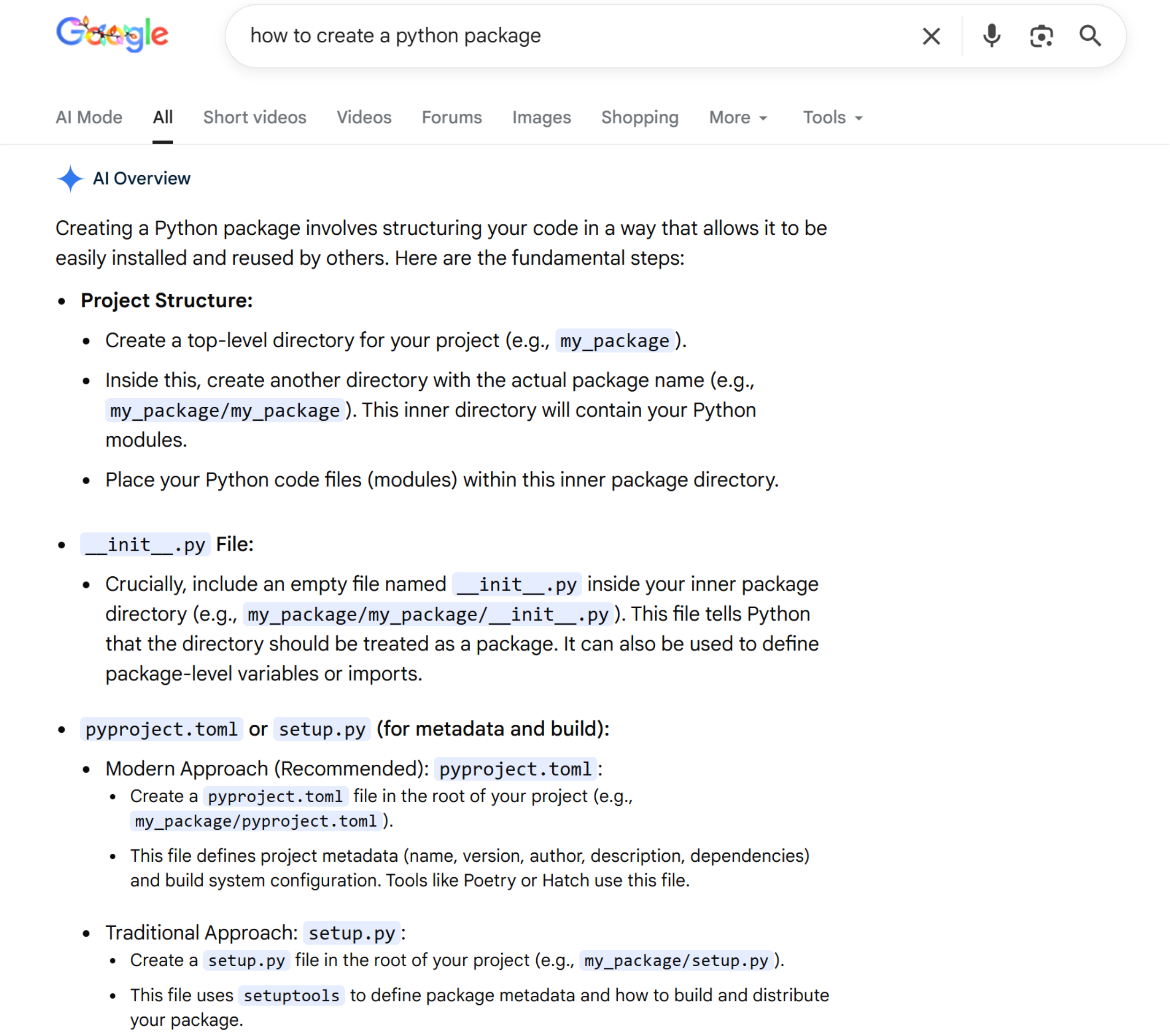

how to create a python package

AI Overview

Creating a Python package involves structuring your code in a way that allows it to be easily installed and reused by others. Here are the fundamental steps:

- **Project Structure:**
  - Create a top-level directory for your project (e.g., `my_package`).
  - Inside this, create another directory with the actual package name (e.g., `my_package/my_package`). This inner directory will contain your Python modules.
  - Place your Python code files (modules) within this inner package directory.
- **`__init__.py` File:**
  - Crucially, include an empty file named `__init__.py` inside your inner package directory (e.g., `my_package/my_package/__init__.py`). This file tells Python that the directory should be treated as a package. It can also be used to define package-level variables or imports.
- **`pyproject.toml` or `setup.py` (for metadata and build):**
  - Modern Approach (Recommended): `pyproject.toml`:
    - Create a `pyproject.toml` file in the root of your project (e.g., `my_package/pyproject.toml`).
    - This file defines project metadata (name, version, author, description, dependencies) and build system configuration. Tools like Poetry or Hatch use this file.
  - Traditional Approach: `setup.py`:
    - Create a `setup.py` file in the root of your project (e.g., `my_package/setup.py`).
    - This file uses `setuptools` to define package metadata and how to build and distribute your package.

- `README.md` and `LICENSE`:
  - Include a `README.md` file in the root of your project to describe your package, its features, installation instructions, and usage examples.
  - Include a `LICENSE` file (e.g., `LICENSE.txt`) specifying the terms under which your package can be used.
- **Build and Distribute (Optional, for PyPI):**
  - If you plan to publish your package to PyPI, you will need to build distribution archives (source distribution and wheel).
  - Using `build` (recommended):

Code

```
pip install build
python -m build
```

- This will create `dist/` directory containing `.tar.gz` and `.whl` files.
- **Installation and Usage:**
  - Once packaged, you can install it locally in a virtual environment using `pip install .` from the root directory of your project.
  - Users can install it from PyPI using `pip install my_package`.
  - Then, you can import and use your modules:

Python

```python
from my_package import my_module
my_module.my_function()
```