

```
if __name__ == __main__
```

Was bedeutet ?

```
if __name__ == __main__
```

if `__name__` == `__main__`

- Aus der Dokumentation zu Python 3.7.1:

`__main__` — Top-level script environment

'`__main__`' is the name of the scope in which top-level code executes. A module's `__name__` is set equal to '`__main__`' when read from standard input, a script, or from an interactive prompt.

A module can discover whether or not it is running in the main scope by checking its own `__name__`, which allows a common idiom for conditionally executing code in a module when it is run as a script or with `python -m` but not when it is imported:

```
if __name__ == "__main__":  
    # execute only if run as a script  
    main()
```

For a package, the same effect can be achieved by including a `__main__.py` module, the contents of which will be executed when the module is run with `-m`.

if `__name__ == __main__`

- `'__main__'` ist der Name der obersten Auswertungs-Umgebung.
- Der Name eines Moduls (einer `.py` -Datei) wird auf `'__main__'` gesetzt, wenn es von der Standardeingabe, einem Script oder einer interaktiven Eingabe aufgerufen wird.
- Standardeingabe: Python-Shell
- Der Name wird allerdings nicht so gesetzt, wenn das Modul importiert wird.
- Dadurch können wir ihn in der Zeile mit dem `if` abfragen, um zu entscheiden, ob der nachfolgende Code ausgeführt werden soll.