
NAMESPACES

*We encounter this problem in everyday life.
(human names)*

NAMESPACES

- Naming system for making names **unique**.
- Give your code a *name* and use it *somewhere else*.
 - The **import** statement
 - example:
 - **import** owner.module

NAMESPACES

```
# helpers.py
```

```
def count_words(word):  
    print(len(word))
```

```
PI = 3.14
```

NAMESPACES

```
# helpers.py
```

```
def count_words(word):  
    print(len(word))
```

```
PI = 3.14
```

```
# worker1.py
```

```
import helpers
```

```
print(helpers.PI)
```

```
helpers.count_words('eggs')
```

NAMESPACES

```
# helpers.py
```

```
def count_words(word):  
    print(len(word))
```

```
PI = 3.14
```

```
# worker2.py
```

```
import helpers as h  
print(h.PI)
```

```
h.count_words('eggs')
```

NAMESPACES

```
# helpers.py
```

```
def count_words(word):  
    print(len(word))
```

```
PI = 3.14
```

```
# worker3.py
```

```
from helpers import*  
print(PI)
```

```
count_words('eggs')
```

NAMESPACES

```
# helpers.py
```

```
def count_words(word):  
    print(len(word))
```

```
PI = 3.14
```

```
# worker4.py
```

```
from helpers import PI as p,  
count_words as foo
```

```
print(p)
```

```
foo('eggs')
```

NAMESPACES – ORGANISE THE HIERARCHICALLY

```
import matplotlib.pyplot as plt
```

NAMESPACES – ORGANISE THE HIERARCHICALLY

```
import matplotlib.pyplot as plt
```

```
plt.show()
```

NAMESPACES – ORGANISE THE HIERARCHICALLY

```
import matplotlib.pyplot as plt
```

```
plt.show()
```

```
matplotlib.pyplot.show()
```

USING MULTIPLE PACKAGES

```
try:
    from fastlib import xyz as foo
except ImportError:
    from slowlib import abc as foo

foo('something',3,4)
```

USING MULTIPLE PACKAGES

```
try:
    from fastlib import xyz as foo
except ImportError:
    from slowlib import abc as foo

foo('something',3,4)
```

```
try:
    from fastlib import xyz as foo
except ImportError:
    from slowlib import abc as _abc
    def foo(x,y,z): return _abc(z,x,y)

foo('something',3,4)
```