IMPORT VARIANTS

AND SOME MISCONCEPTIONS

Import variants

```
# module1.py
import math
```

is `math` in `sys.modules`?

if not, load it and insert ref

`sys.modules`

`math`	*<module object>*

add symbol `math` to `module1`'s global namespace referencing the same object

`module1.globals()`

`math`	*<module object>*

`math` symbol in namespace

(if `math` symbol already exists in `module1`'s namespace, replace reference)

Import variants

```
# module1.py
import math as r_math
```

sys.modules

math	<module object>

is `math` in `sys.modules`?

if not, load it and insert ref

add symbol `r_math` to `module1`'s global namespace referencing the same object

`module1.globals()`

r_math	<module object>

`r_math` symbol in namespace
`math` symbol not in namespace

(if `r_math` symbol already exists in `module1`'s namespace, replace reference)

Import variants

```
# module1.py
from math import sqrt
```

is `math` in `sys.modules`?

if not, load it and insert ref

`sys.modules`

```
math      <module object>
```

add symbol `sqrt` to `module1`'s global namespace referencing `math.sqrt`

`module1.globals()`

```
sqrt   <math.sqrt object>
```

`math` symbol not in namespace

(if `sqrt` symbol already exists in `module1`'s namespace, replace reference)

Import variants

```
# module1.py
from math import sqrt as r_sqrt
```

is `math` in `sys.modules`?

if not, load it and insert ref

`sys.modules`

```
math      <module object>
```

add symbol `r_sqrt` to `module1`'s global namespace referencing `math.sqrt`

`module1.globals()`

```
r_sqrt <math.sqrt object>
```

`math` symbol not in namespace

(if `r_sqrt` symbol already exists in `module1`'s namespace, replace reference)

Import variants

```
# module1.py
from math import *
```

is `math` in `sys.modules`?

if not, load it and insert ref

`sys.modules`

```
math      <module object>
```

add "all" symbols defined in math to `module1`'s global namespace

what "all" means can be defined by the module being imported

`module1.globals()`

```
pi        <math.pi object>
sin       <math.sin object>
and many more...
```

`math` symbol not in namespace

(if any symbols already exists in `module1`'s namespace, replace their reference)

Commonality

In **every** case the `math` module was loaded into memory and referenced in `sys.modules`

Running `from math import sqrt`

did not "partially" load math

it only affected what symbols were placed in `module1`'s namespace!

Things may be different with packages, but for simple modules this is the behavior

Why `from <module> import *` can lead to bugs

```
# module1.py
from cmath import *
```

```
module1.globals()
  sqrt   <cmath.sqrt>
  ...
```

```
from math import *
```

```
module1.globals()
  sqrt   <math.sqrt>
  ...
```

Efficiency

What's more efficient?

```
import math
```

or

```
from math import sqrt
```

importing → same amount of work

calling `math.sqrt(2)`

`sqrt(2)`

This first needs to find the `sqrt` symbol in `math`'s namespace

`dict` lookup → super fast!

Code