replacing import with accio

muggle python

```
>>> import random

>>> random.random()
0.18660693012691754

>>> accio random
  File "<stdin>", line 1
    accio random
                ^
SyntaxError: invalid syntax
```

```
>>> import random

>>> random.random()
0.18660693012691754

>>> accio random
  File "<stdin>", line 1
    accio random
          ^
SyntaxError: invalid syntax
```

```
>>> import random

>>> random.random()
0.186606930012691754

>>> accio random
  File "<stdin>", line 1
    accio random
               ^
SyntaxError: invalid syntax
```

```
>>> import random

>>> random.random()
0.18660693012691754


>>> accio random
  File "<stdin>", line 1
    accio random
                ^
SyntaxError: invalid syntax
```

```
>>> import random

>>> random.random()
0.18660693012691754


>>> accio random
  File "<stdin>", line 1
    accio random
                ^
SyntaxError: invalid syntax
```

nagini

```
>>> accio random     # no error

>>> random.random()
0.18660693012691754


>>> import random
  File "<stdin>", line 1
    import random
                 ^
SyntaxError: invalid syntax
```

```
>>> accio random      # no error

>>> random.random()
0.18660693012691754

>>> import random
  File "<stdin>", line 1
    import random
                 ^
SyntaxError: invalid syntax
```

```
>>> accio random      # no error

>>> random.random()
0.186606930126917540

>>> import random
  File "<stdin>", line 1
    import random
                 ^
SyntaxError: invalid syntax
```

```
>>> accio random      # no error

>>> random.random()
0.18660693012691754

>>> import random
  File "<stdin>", line 1
    import random
                ^
SyntaxError: invalid syntax
```

```
>>> accio random     # no error

>>> random.random()
0.18660693012691754


>>> import random
  File "<stdin>", line 1
    import random
                 ^
SyntaxError: invalid syntax
```

```
__builtins__
import
Grammar
bootstrapping
```

__builtins__

import

Grammar

bootstrapping

```
>>> __builtins__
<module '__builtin__' (built-in)>

>>> dir(__builtins__)
['ArithmeticError',…,'__import__',
…,'float',…,'zip']
```

```
>>> __builtins__
<module '__builtin__' (built-in)>


>>> dir(__builtins__)
['ArithmeticError',…,'__import__',
…,'float',…,'zip']
```

```
>>> __builtins__
<module '__builtin__' (built-in)>


>>> dir(__builtins__)
['ArithmeticError',…,'__import__',
…,'float',…,'zip']
```

```
>>> __builtins__
<module '__builtin__' (built-in)>


>>> dir(__builtins__)
['ArithmeticError',…,'__import__',
…,'float',…,'zip']
```

we can shadow builtins!

```
>>> float(3)

3.0


>>> wingardium_leviosa =
__builtins__.float


>>> wingardium_leviosa(3)

3.0
```

```
>>> float(3)

3.0


>>> wingardium_leviosa =
__builtins__.float


>>> wingardium_leviosa(3)

3.0
```

```
>>> float(3)

3.0


>>> wingardium_leviosa =
__builtins__.float


>>> wingardium_leviosa(3)

3.0
```

```
>>> float(3)
3.0


>>> wingardium_leviosa =
__builtins__.float


>>> wingardium_leviosa(3)
3.0
```

```
>>> float(3)
3.0


>>> wingardium_leviosa =
__builtins__.float


>>> wingardium_leviosa(3)
3.0
```

we can delete
builtins!

```
>>> del __builtins__.float

>>> float(3)
Traceback (most recent call last):
  File "<stdin>", line 1, in <module>
NameError: name 'float' is not
defined


>>> wingardium_leviosa(3)
3.0
```

```
>>> del __builtins__.float

>>> float(3)
Traceback (most recent call last):
  File "<stdin>", line 1, in <module>
NameError: name 'float' is not
defined

>>> wingardium_leviosa(3)
3.0
```

```
>>> del __builtins__.float

>>> float(3)
Traceback (most recent call last):
  File "<stdin>", line 1, in <module>
NameError: name 'float' is not
defined

>>> wingardium_leviosa(3)
3.0
```

```
>>> del __builtins__.float

>>> float(3)
Traceback (most recent call last):
  File "<stdin>", line 1, in <module>
NameError: name 'float' is not
defined

>>> wingardium_leviosa(3)
3.0
```

```
>>> del __builtins__.float

>>> float(3)
Traceback (most recent call last):
  File "<stdin>", line 1, in <module>
NameError: name 'float' is not
defined


>>> wingardium_leviosa(3)
3.0
```

__builtins__

import

Grammar

bootstrapping

can we shadow import?

```
>>> accio = import
  File "<stdin>", line 1
    accio = import
                  ^
SyntaxError: invalid syntax
```

```
>>> accio = import
  File "<stdin>", line 1
    accio = import
                  ^
SyntaxError: invalid syntax
```

:(

can we delete import?

```
>>> del import
  File "<stdin>", line 1
    del = import
          ^
SyntaxError: invalid syntax
```

```
>>> del import
  File "<stdin>", line 1
    del = import
                ^
SyntaxError: invalid syntax
```

:(

__import__

```
>>> dir(__builtins__)
['ArithmeticError',…,'__import__',
…,'float',…,'zip']


>>> __builtins__.__import__
<built-in function __import__>
```

```
>>> dir(__builtins__)

['ArithmeticError',…,'__import__',
…,'float',…,'zip']


>>> __builtins__.__import__
<built-in function __import__>
```

```
>>> dir(__builtins__)

['ArithmeticError',…,'__import__',
…,'float',…,'zip']


>>> __builtins__.__import__
<built-in function __import__>
```

```
>>> dir(__builtins__)
['ArithmeticError',…,'__import__',
…,'float',…,'zip']


>>> __builtins__.__import__
<built-in function __import__>
```

```
>>> __builtins__.__import__(random)
Traceback (most recent call last):
  File "<stdin>", line 1, in <module>
NameError: name 'random' is not
defined


>>> __builtins__.__import__('random')
<module 'random' from '/usr/local/…/
random.pyc'>
```

```
>>> __builtins__.__import__(random)
Traceback (most recent call last):
  File "<stdin>", line 1, in <module>
NameError: name 'random' is not
defined


>>> __builtins__.__import__('random')
<module 'random' from '/usr/local/…/
random.pyc'>
```

```
>>> __builtins__.__import__(random)
Traceback (most recent call last):
  File "<stdin>", line 1, in <module>
NameError: name 'random' is not
defined


>>> __builtins__.__import__('random')
<module 'random' from '/usr/local/…/
random.pyc'>
```

```
>>> __builtins__.__import__(random)
Traceback (most recent call last):
  File "<stdin>", line 1, in <module>
NameError: name 'random' is not
defined


>>> __builtins__.__import__('random')
<module 'random' from '/usr/local/…/
random.pyc'>
```

```
>>> random =
__builtins__.__import__('random')

>>> random.random()
0.05857571707711129
```

```
>>> random =
__builtins__.__import__('random')


>>> random.random()
0.05857571707711129
```

```
>>> random =
__builtins__.__import__('random')

>>> random.random()
0.05857571707711129
```

can we shadow
__import__?

```
>>> accio = __builtins__.__import__

>>> random = accio('random')

>>> random.random()
0.873593464437295
```

```
>>> accio = __builtins__.__import__

>>> random = accio('random')

>>> random.random()
0.873593464377295
```

```
>>> accio = __builtins__.__import__

>>> random = accio('random')

>>> random.random()
0.873593464437295
```

```
>>> accio = __builtins__.__import__

>>> random = accio('random')

>>> random.random()
0.873593464437295
```

can we delete
__import__?

```
>>> del __builtins__.__import__
```

```
>>> random = accio('random')
Traceback (most recent call last):
  File "<stdin>", line 1, in <module>
  File "/usr/local/…/random.py", line
42, in <module>
    from __future__ import division
ImportError: __import__ not found
```

```
>>> del __builtins__.__import__

>>> random = accio('random')
Traceback (most recent call last):
  File "<stdin>", line 1, in <module>
  File "/usr/local/…/random.py", line
42, in <module>
    from __future__ import division
ImportError: __import__ not found
```

```
>>> del __builtins__.__import__

>>> random = accio('random')
Traceback (most recent call last):
  File "<stdin>", line 1, in <module>
  File "/usr/local/…/random.py", line
42, in <module>
    from __future__ import division
ImportError: __import__ not found
```

`import x` is a statement

`__builtins__`

`import`

Grammar

`bootstrapping`

clone cpython!

```
$ ls -d cpython/*/
Doc/            Objects/
Grammar/        PC/
Include/        PCbuild/
Lib/            Parser/
Mac/            Python/
Misc/           Tools/
Modules/
```

```
$ ls -d cpython/*/
Doc/            Objects/

Grammar/        PC/

Include/        PCbuild/

Lib/            Parser/

Mac/            Python/

Misc/           Tools/

Modules/
```

```
$ ls -d cpython/*/

Doc/              Objects/

Grammar/          PC/

Include/          PCbuild/

Lib/              Parser/

Mac/              Python/

Misc/             Tools/

Modules/
```

```
import_stmt: import_name |
   import_from
import_name: 'import' dotted_as_names
import_from: ('from' ('.'*
   dotted_name | '.'+)'import' ('*'
   | '(' import_as_names ')' |
   import_as_names))
```

```
import_stmt: import_name |
    import_from

import_name: 'import' dotted_as_names

import_from: ('from' ('.'*
    dotted_name | '.'+)'import' ('*'
    | '(' import_as_names ')' |
    import_as_names))
```

import_from: ('from' ('.'*

dotted_name | '.'+)'import' ('*'

| '(' import_as_names ')' |

import_as_names))

```
import_name:

    import matplotlib.pyplot as plt


import_from:

    from matplotlib import pyplot as plt
```

```
import_name:
    import matplotlib.pyplot as plt


import_from:
    from matplotlib import pyplot as plt
```

```
import_stmt: import_name |
    import_from

import_name: 'import' dotted_as_names

import_from: ('from' ('.'*
    dotted_name | '.'+) 'import' ('*'
    | '(' import_as_names ')' |
    import_as_names))
```

```
import_stmt: import_name |
    import_from
import_name: 'accio' dotted_as_names
import_from: ('from' ('.'*
    dotted_name | '.'+) 'accio' ('*'
    | '(' import_as_names ')' |
    import_as_names))
```

```
$ make

Traceback (most recent call last):

  …

  File "/…/cpython/Lib/sysconfig.py",
line 4

    import sys

         ^

SyntaxError: invalid syntax
```

```
$ make

Traceback (most recent call last):

 …

 File "/…/cpython/Lib/sysconfig.py",
line 4

     import sys

          ^

SyntaxError: invalid syntax
```

half of cpython is
written in python!

```
$ (script that replaces `import` with
`accio` in all .py files)


$ make

Traceback (most recent call last):

 …

    accio sys

       ^

SyntaxError: invalid syntax
```

```
$ (script that replaces `import` with
`accio` in all .py cpython files)


$ make
Traceback (most recent call last):

  …

    accio sys


       ^

SyntaxError: invalid syntax
```

```
$ (script that replaces `import` with
`accio` in all .py cpython files)


$ make

Traceback (most recent call last):

 …

    accio sys

         ^

SyntaxError: invalid syntax
```

wat!

`__builtins__`

`import Grammar`

bootstrapping

```
import_stmt: import_name |
    import_from

import_name: 'import' dotted_as_names

import_from: ('from' ('.'*
    dotted_name | '.'+)'import' ('*'
    | '(' import_as_names ')' |
    import_as_names))
```

```
import_stmt: import_name |
    import_from
import_name: 'import' dotted_as_names
import_from: ('from' ('.'*
    dotted_name | '.'+)'import' ('*'
    | '(' import_as_names ')' |
    import_as_names))
```

```
import_name: 'import' dotted_as_names
           | 'accio' dotted_as_names
```

```
import_name: 'import' dotted_as_names
           | 'accio' dotted_as_names
```

```
$ make


$ ./python


>>> accio random     # no errors!


>>> import sys       # no errors :(
```

```
$ make

$ ./python

>>> accio random       # no errors!

>>> import sys         # no errors :(
```

```
$ make


$ ./python


>>> accio random     # no errors!


>>> import sys       # no errors :(
```

```
$ make


$ ./python


>>> accio random     # no errors!


>>> import sys       # no errors :(
```

yo dawg I heard
you like pythons

fin