

# Outline

## Open Source Software

licensing software

Copyright and CopyLeft

## Python

More on Modules

Packages

## Numerical Python: NumPy

an open source project

## Summary + Assignments

MCS 260 Lecture 20  
Introduction to Computer Science  
Jan Vershelde, 12 October 2007

### Open Source Software

licensing software  
Copyright and CopyLeft

### Python

More on Modules  
Packages

### Numerical Python: NumPy

an open source project

### Summary + Assignments

# Outline

Open Source Software  
licensing software  
Copyright and CopyLeft

Python  
More on Modules  
Packages

Numerical Python: NumPy  
an open source project

Summary + Assignments

MCS 260 L-20

12 October 2007

Open Source  
Software

licensing software  
Copyright and CopyLeft

Python

More on Modules  
Packages

Numerical Python:  
NumPy

an open source project

Summary +  
Assignments

# History of Open Source

MCS 260 L-20

12 October 2007

In the 1960s, software came free with hardware.  
Users were programmers and software developers.

In the mid 1970s, software became a commercial product.  
Personal Computing meant that users did not program.

Unix, developed at Bell Labs and University of Berkeley (BSD) became a commercial product of AT&T in 1984.

GNU (not Unix) project started at MIT in 1984 by Richard Stallman. The Free Software Foundation published a first version of the General Public License (GPL) in 1988.

In 1991, a first version of what later became Linux is released. Linux = Linus+Unix, started by Linus Torvalds.

MIT started the OpenCourseWare project in 2000.

Open Source  
Software

licensing software  
Copyright and CopyLeft

Python

More on Modules  
Packages

Numerical Python:  
NumPy

an open source project

Summary +  
Assignments

# History of Open Source

MCS 260 L-20

12 October 2007

In the 1960s, software came free with hardware.  
Users were programmers and software developers.

In the mid 1970s, software became a commercial product.  
Personal Computing meant that users did not program.

Unix, developed at Bell Labs and University of Berkeley (BSD) became a commercial product of AT&T in 1984.

GNU (not Unix) project started at MIT in 1984 by Richard Stallman. The Free Software Foundation published a first version of the General Public License (GPL) in 1988.

In 1991, a first version of what later became Linux is released. Linux = Linus+Unix, started by Linus Torvalds.

MIT started the OpenCourseWare project in 2000.

Open Source  
Software

licensing software  
Copyright and CopyLeft

Python

More on Modules  
Packages

Numerical Python:  
NumPy

an open source project

Summary +  
Assignments

# History of Open Source

MCS 260 L-20

12 October 2007

In the 1960s, software came free with hardware.  
Users were programmers and software developers.

In the mid 1970s, software became a commercial product.  
Personal Computing meant that users did not program.

Unix, developed at Bell Labs and University of Berkeley  
(BSD) became a commercial product of AT&T in 1984.

GNU (not Unix) project started at MIT in 1984 by Richard  
Stallman. The Free Software Foundation published a first  
version of the General Public License (GPL) in 1988.

In 1991, a first version of what later became Linux is  
released. Linux = Linus+Unix, started by Linus Torvalds.

MIT started the OpenCourseWare project in 2000.

Open Source  
Software

licensing software  
Copyright and CopyLeft

Python

More on Modules  
Packages

Numerical Python:  
NumPy

an open source project

Summary +  
Assignments

# History of Open Source

MCS 260 L-20

12 October 2007

In the 1960s, software came free with hardware.  
Users were programmers and software developers.

In the mid 1970s, software became a commercial product.  
Personal Computing meant that users did not program.

Unix, developed at Bell Labs and University of Berkeley (BSD) became a commercial product of AT&T in 1984.

GNU (not Unix) project started at MIT in 1984 by Richard Stallman. The Free Software Foundation published a first version of the General Public License (GPL) in 1988.

In 1991, a first version of what later became Linux is released. Linux = Linus+Unix, started by Linus Torvalds.

MIT started the OpenCourseWare project in 2000.

Open Source  
Software

licensing software  
Copyright and CopyLeft

Python

More on Modules  
Packages

Numerical Python:  
NumPy

an open source project

Summary +  
Assignments

# History of Open Source

MCS 260 L-20

12 October 2007

In the 1960s, software came free with hardware.  
Users were programmers and software developers.

In the mid 1970s, software became a commercial product.  
Personal Computing meant that users did not program.

Unix, developed at Bell Labs and University of Berkeley (BSD) became a commercial product of AT&T in 1984.

GNU (not Unix) project started at MIT in 1984 by Richard Stallman. The Free Software Foundation published a first version of the General Public License (GPL) in 1988.

In 1991, a first version of what later became Linux is released. Linux = Linus+Unix, started by Linus Torvalds.

MIT started the OpenCourseWare project in 2000.

Open Source  
Software

licensing software  
Copyright and CopyLeft

Python

More on Modules  
Packages

Numerical Python:  
NumPy

an open source project

Summary +  
Assignments

# History of Open Source

MCS 260 L-20

12 October 2007

In the 1960s, software came free with hardware.  
Users were programmers and software developers.

In the mid 1970s, software became a commercial product.  
Personal Computing meant that users did not program.

Unix, developed at Bell Labs and University of Berkeley (BSD) became a commercial product of AT&T in 1984.

GNU (not Unix) project started at MIT in 1984 by Richard Stallman. The Free Software Foundation published a first version of the General Public License (GPL) in 1988.

In 1991, a first version of what later became Linux is released. Linux = Linus+Unix, started by Linus Torvalds.

MIT started the OpenCourseWare project in 2000.

Open Source  
Software

licensing software  
Copyright and CopyLeft

Python

More on Modules  
Packages

Numerical Python:  
NumPy

an open source project

Summary +  
Assignments

# Intellectual Property Rights

and software

Intellectual property rights (IPR) grant authors the possibility to control the copying, modification, and distribution of the products.

Four tools to protect IPR:

**trade secret** confidential information used to compete

**trademark** is word, name, symbol used to distinguish

**copyright** grants authors exclusive rights to copy, sell, license, distribute, modify, translate, etc.

**patent** gives creators exclusive rights to manufacture and sell invented goods.

A commercial program is typically delivered by giving only its object code, in contrast to open source.

Maple is somewhat hybrid, but the license explicitly forbids the reverse engineering of its code.

[Open Source Software](#)

licensing software  
Copyright and CopyLeft

[Python](#)

More on Modules  
Packages

[Numerical Python: NumPy](#)

an open source project

[Summary + Assignments](#)

# Intellectual Property Rights

and software

Intellectual property rights (IPR) grant authors the possibility to control the copying, modification, and distribution of the products.

Four tools to protect IPR:

**trade secret** confidential information used to compete

**trademark** is word, name, symbol used to distinguish

**copyright** grants authors exclusive rights to copy, sell, license, distribute, modify, translate, etc.

**patent** gives creators exclusive rights to manufacture and sell invented goods.

A commercial program is typically delivered by giving only its object code, in contrast to open source.

Maple is somewhat hybrid, but the license explicitly forbids the reverse engineering of its code.

[Open Source Software](#)

licensing software  
Copyright and CopyLeft

[Python](#)

More on Modules  
Packages

[Numerical Python:](#)

[NumPy](#)  
an open source project

[Summary + Assignments](#)

# Intellectual Property Rights

and software

Intellectual property rights (IPR) grant authors the possibility to control the copying, modification, and distribution of the products.

Four tools to protect IPR:

**trade secret** confidential information used to compete

**trademark** is word, name, symbol used to distinguish

**copyright** grants authors exclusive rights to copy, sell, license, distribute, modify, translate, etc.

**patent** gives creators exclusive rights to manufacture and sell invented goods.

A commercial program is typically delivered by giving only its object code, in contrast to open source.

Maple is somewhat hybrid, but the license explicitly forbids the reverse engineering of its code.

[Open Source Software](#)

licensing software  
Copyright and CopyLeft

[Python](#)

More on Modules  
Packages

[Numerical Python: NumPy](#)

an open source project

[Summary + Assignments](#)

# Intellectual Property Rights

and software

Intellectual property rights (IPR) grant authors the possibility to control the copying, modification, and distribution of the products.

Four tools to protect IPR:

**trade secret** confidential information used to compete

**trademark** is word, name, symbol used to distinguish

**copyright** grants authors exclusive rights to copy, sell, license, distribute, modify, translate, etc.

**patent** gives creators exclusive rights to manufacture and sell invented goods.

A commercial program is typically delivered by giving only its object code, in contrast to open source.

Maple is somewhat hybrid, but the license explicitly forbids the reverse engineering of its code.

[Open Source Software](#)

licensing software  
Copyright and CopyLeft

[Python](#)

More on Modules  
Packages

[Numerical Python: NumPy](#)

an open source project

[Summary + Assignments](#)

# Intellectual Property Rights

and software

Intellectual property rights (IPR) grant authors the possibility to control the copying, modification, and distribution of the products.

Four tools to protect IPR:

**trade secret** confidential information used to compete

**trademark** is word, name, symbol used to distinguish

**copyright** grants authors exclusive rights to copy, sell, license, distribute, modify, translate, etc.

**patent** gives creators exclusive rights to manufacture and sell invented goods.

A commercial program is typically delivered by giving only its object code, in contrast to open source.

Maple is somewhat hybrid, but the license explicitly forbids the reverse engineering of its code.

[Open Source Software](#)

licensing software  
Copyright and CopyLeft

[Python](#)

More on Modules  
Packages

[Numerical Python: NumPy](#)

an open source project

[Summary + Assignments](#)

# Intellectual Property Rights

and software

Intellectual property rights (IPR) grant authors the possibility to control the copying, modification, and distribution of the products.

Four tools to protect IPR:

**trade secret** confidential information used to compete

**trademark** is word, name, symbol used to distinguish

**copyright** grants authors exclusive rights to copy, sell, license, distribute, modify, translate, etc.

**patent** gives creators exclusive rights to manufacture and sell invented goods.

A commercial program is typically delivered by giving only its object code, in contrast to open source.

Maple is somewhat hybrid, but the license explicitly forbids the reverse engineering of its code.

[Open Source Software](#)

licensing software  
Copyright and CopyLeft

[Python](#)

More on Modules  
Packages

[Numerical Python: NumPy](#)

an open source project

[Summary + Assignments](#)

# Intellectual Property Rights

and software

Intellectual property rights (IPR) grant authors the possibility to control the copying, modification, and distribution of the products.

Four tools to protect IPR:

**trade secret** confidential information used to compete

**trademark** is word, name, symbol used to distinguish

**copyright** grants authors exclusive rights to copy, sell, license, distribute, modify, translate, etc.

**patent** gives creators exclusive rights to manufacture and sell invented goods.

A commercial program is typically delivered by giving only its object code, in contrast to open source.

Maple is somewhat hybrid, but the license explicitly forbids the reverse engineering of its code.

[Open Source Software](#)

[licensing software](#)  
[Copyright and CopyLeft](#)

[Python](#)

[More on Modules](#)  
[Packages](#)

[Numerical Python:](#)

[NumPy](#)  
an open source project

[Summary + Assignments](#)

# General Public License

MCS 260 L-20

12 October 2007

Motivation: needs and benefits of open source

1. mission critical and scientific applications place a very strong emphasis on correctness.
2. *given enough eyeballs, all bugs are shallow.*

Note: free as in free speech, not free beer.

Software is also a service.

Copyleft as reverse of *copyright*.

1. first state that the code is copyrighted
2. give everyone all rights on the code
3. *only if the distribution terms are unchanged*

The code and the freedoms become legally inseparable.

The license imposes a 'share alike' restriction.

Many open source licenses exist, 4 most popular: GNU GPL, GNU Library or Lesser GPL, Mozilla Public License, and Berkeley Software Distribution.

Open Source  
Software

licensing software  
Copyright and CopyLeft

Python

More on Modules  
Packages

Numerical Python:  
NumPy

an open source project

Summary +  
Assignments

# General Public License

MCS 260 L-20

12 October 2007

Motivation: needs and benefits of open source

1. mission critical and scientific applications place a very strong emphasis on correctness.
2. *given enough eyeballs, all bugs are shallow.*

Note: free as in free speech, not free beer.

Software is also a service.

Copyleft as reverse of *copyright*.

1. first state that the code is copyrighted
2. give everyone all rights on the code
3. *only if the distribution terms are unchanged*

The code and the freedoms become legally inseparable.

The license imposes a 'share alike' restriction.

Many open source licenses exist, 4 most popular: GNU GPL, GNU Library or Lesser GPL, Mozilla Public License, and Berkeley Software Distribution.

Open Source  
Software

licensing software  
Copyright and CopyLeft

Python

More on Modules  
Packages

Numerical Python:  
NumPy

an open source project

Summary +  
Assignments

# General Public License

MCS 260 L-20

12 October 2007

Motivation: needs and benefits of open source

1. mission critical and scientific applications place a very strong emphasis on correctness.
2. *given enough eyeballs, all bugs are shallow.*

Note: free as in free speech, not free beer.

Software is also a service.

Copyleft as reverse of *copyright*.

1. first state that the code is copyrighted
2. give everyone all rights on the code
3. *only if the distribution terms are unchanged*

The code and the freedoms become legally inseparable.

The license imposes a 'share alike' restriction.

Many open source licenses exist, 4 most popular: GNU GPL, GNU Library or Lesser GPL, Mozilla Public License, and Berkeley Software Distribution.

Open Source  
Software

licensing software  
Copyright and CopyLeft

Python

More on Modules  
Packages

Numerical Python:  
NumPy

an open source project

Summary +  
Assignments

# General Public License

MCS 260 L-20

12 October 2007

Motivation: needs and benefits of open source

1. mission critical and scientific applications place a very strong emphasis on correctness.
2. *given enough eyeballs, all bugs are shallow.*

Note: free as in free speech, not free beer.

Software is also a service.

Copyleft as reverse of *copyright*.

1. first state that the code is copyrighted
2. give everyone all rights on the code
3. *only if the distribution terms are unchanged*

The code and the freedoms become legally inseparable.

The license imposes a 'share alike' restriction.

Many open source licenses exist, 4 most popular: GNU GPL, GNU Library or Lesser GPL, Mozilla Public License, and Berkeley Software Distribution.

Open Source  
Software

licensing software  
Copyright and CopyLeft

Python

More on Modules  
Packages

Numerical Python:  
NumPy

an open source project

Summary +  
Assignments

# General Public License

MCS 260 L-20

12 October 2007

Motivation: needs and benefits of open source

1. mission critical and scientific applications place a very strong emphasis on correctness.
2. *given enough eyeballs, all bugs are shallow.*

Note: free as in free speech, not free beer.

Software is also a service.

Copyleft as reverse of *copyright*.

1. first state that the code is copyrighted
2. give everyone all rights on the code
3. *only if the distribution terms are unchanged*

The code and the freedoms become legally inseparable.

The license imposes a 'share alike' restriction.

Many open source licenses exist, 4 most popular: GNU GPL, GNU Library or Lesser GPL, Mozilla Public License, and Berkeley Software Distribution.

Open Source  
Software

licensing software  
Copyright and CopyLeft

Python

More on Modules  
Packages

Numerical Python:  
NumPy

an open source project

Summary +  
Assignments

# General Public License

MCS 260 L-20

12 October 2007

Motivation: needs and benefits of open source

1. mission critical and scientific applications place a very strong emphasis on correctness.
2. *given enough eyeballs, all bugs are shallow.*

Note: free as in free speech, not free beer.

Software is also a service.

Copyleft as reverse of *copyright*.

1. first state that the code is copyrighted
2. give everyone all rights on the code
3. *only if the distribution terms are unchanged*

The code and the freedoms become legally inseparable.

The license imposes a 'share alike' restriction.

Many open source licenses exist, 4 most popular: GNU GPL, GNU Library or Lesser GPL, Mozilla Public License, and Berkeley Software Distribution.

Open Source  
Software

licensing software  
Copyright and CopyLeft

Python

More on Modules  
Packages

Numerical Python:  
NumPy

an open source project

Summary +  
Assignments

# General Public License

Motivation: needs and benefits of open source

1. mission critical and scientific applications place a very strong emphasis on correctness.
2. *given enough eyeballs, all bugs are shallow.*

Note: free as in free speech, not free beer.

Software is also a service.

Copyleft as reverse of *copyright*.

1. first state that the code is copyrighted
2. give everyone all rights on the code
3. *only if the distribution terms are unchanged*

The code and the freedoms become legally inseparable.

The license imposes a 'share alike' restriction.

Many open source licenses exist, 4 most popular: GNU GPL, GNU Library or Lesser GPL, Mozilla Public License, and Berkeley Software Distribution.

Open Source  
Software

licensing software  
Copyright and CopyLeft

Python

More on Modules  
Packages

Numerical Python:  
NumPy

an open source project

Summary +  
Assignments

# Outline

## Open Source Software

licensing software

Copyright and Copyleft

## Python

More on Modules

Packages

## Numerical Python: NumPy

an open source project

## Summary + Assignments

MCS 260 L-20

12 October 2007

### Open Source Software

licensing software

Copyright and Copyleft

### Python

More on Modules

Packages

### Numerical Python: NumPy

an open source project

### Summary + Assignments

# Copyright applied to Software

MCS 260 L-20

12 October 2007

Open Source  
Software

licensing software  
Copyright and Copyleft

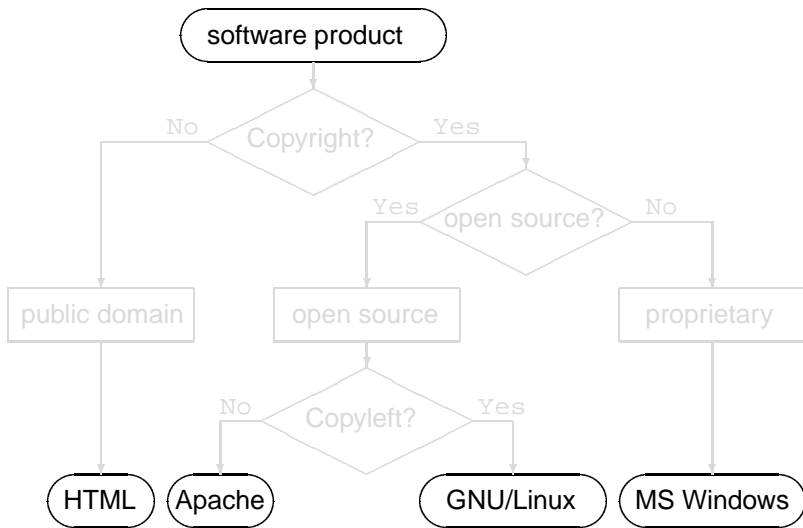
Python

More on Modules  
Packages

Numerical Python:  
NumPy

an open source project

Summary +  
Assignments



M. Muffatto: *Open Source. A Multidisciplinary Approach*, 2006.

# Copyright applied to Software

MCS 260 L-20

12 October 2007

Open Source  
Software

licensing software  
Copyright and Copyleft

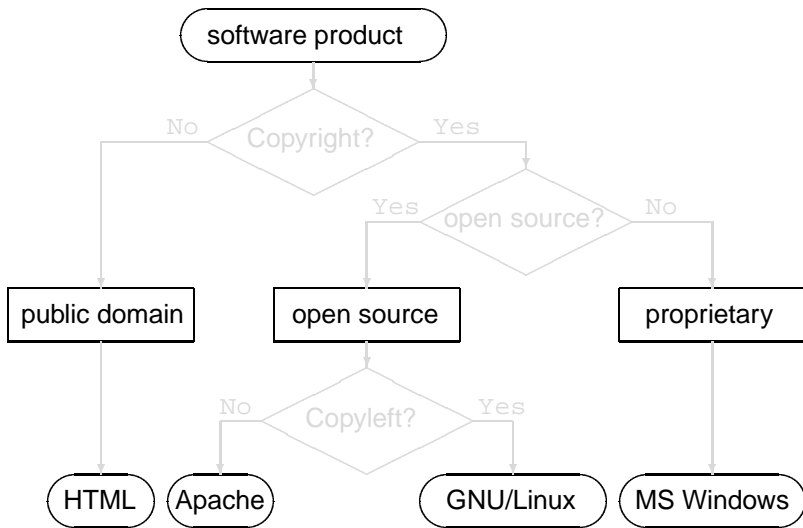
Python

More on Modules  
Packages

Numerical Python:  
NumPy

an open source project

Summary +  
Assignments



M. Muffatto: *Open Source. A Multidisciplinary Approach*, 2006.

# Copyright applied to Software

MCS 260 L-20

12 October 2007

Open Source  
Software

licensing software  
Copyright and Copyleft

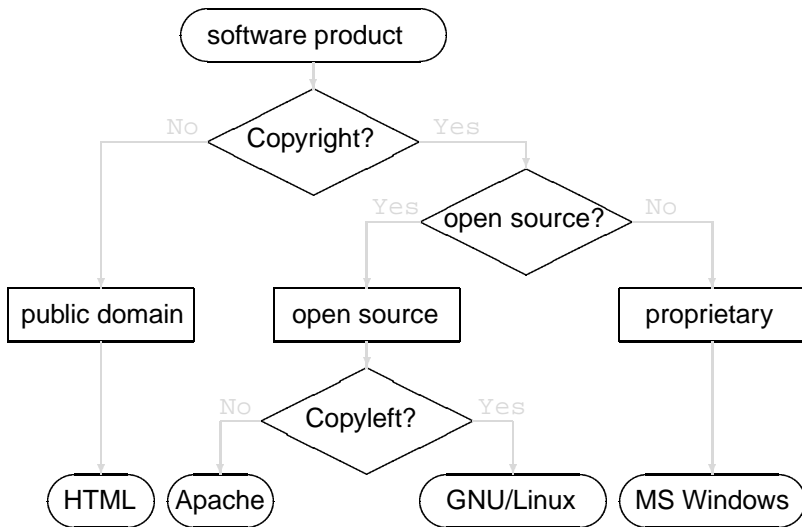
Python

More on Modules  
Packages

Numerical Python:  
NumPy

an open source project

Summary +  
Assignments



M. Muffatto: *Open Source. A Multidisciplinary Approach*, 2006.

# Copyright applied to Software

MCS 260 L-20

12 October 2007

[Open Source Software](#)

[licensing software](#)  
[Copyright and CopyLeft](#)

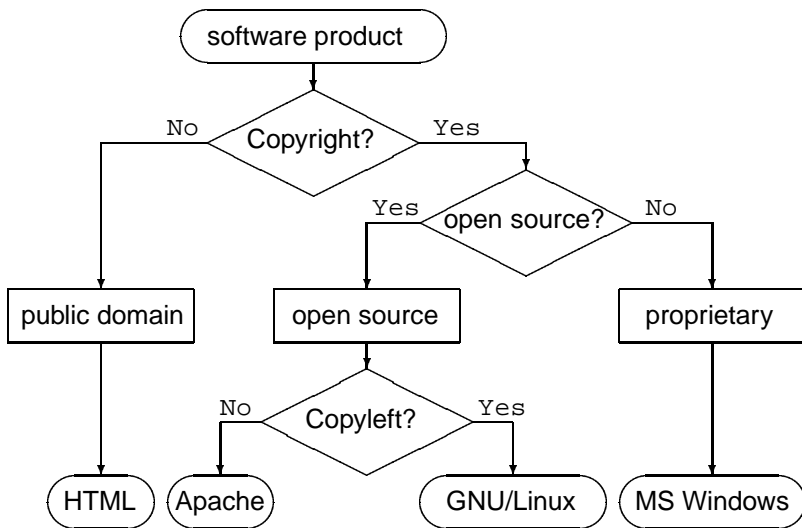
[Python](#)

[More on Modules](#)  
[Packages](#)

[Numerical Python: NumPy](#)

[an open source project](#)

[Summary + Assignments](#)



M. Muffatto: *Open Source. A Multidisciplinary Approach*, 2006.

# Outline

## Open Source Software

licensing software

Copyright and CopyLeft

## Python

**More on Modules**

Packages

## Numerical Python: NumPy

an open source project

## Summary + Assignments

MCS 260 L-20

12 October 2007

### Open Source Software

licensing software  
Copyright and CopyLeft

### Python

**More on Modules**  
Packages

### Numerical Python: NumPy

an open source project

### Summary + Assignments

# Module Search Path

MCS 260 L-20

12 October 2007

The module `sys` is built-in, after

```
>>> import sys
>>> help(sys)
```

The description starts with

```
This module provides access to some objects used
or maintained by the interpreter and to functions
that interact strongly with the interpreter.
```

Where does Python find the modules?

```
>>> sys.path
['', '/usr/local/lib/python25.zip', ...
```

The `sys.path` is a list where to look for modules, starting at the first element in the list, usually the current directory, to change this:

```
>>> sys.path.insert(0, '/tmp')
```

[Open Source Software](#)

licensing software  
Copyright and CopyLeft

[Python](#)

[More on Modules Packages](#)

[Numerical Python: NumPy](#)

an open source project

[Summary + Assignments](#)

[Open Source Software](#)licensing software  
Copyright and CopyLeft[Python](#)[More on Modules Packages](#)[Numerical Python: NumPy](#)

an open source project

[Summary + Assignments](#)

# Module Search Path

The module `sys` is built-in, after

```
>>> import sys
>>> help(sys)
```

The description starts with

This module provides access to some objects used or maintained by the interpreter and to functions that interact strongly with the interpreter.

Where does Python find the modules?

```
>>> sys.path
['', '/usr/local/lib/python25.zip', ...]
```

The `sys.path` is a list where to look for modules, starting at the first element in the list, usually the current directory, to change this:

```
>>> sys.path.insert(0, '/tmp')
```

# Module Search Path

MCS 260 L-20

12 October 2007

The module `sys` is built-in, after

```
>>> import sys
>>> help(sys)
```

The description starts with

This module provides access to some objects used or maintained by the interpreter and to functions that interact strongly with the interpreter.

Where does Python find the modules?

```
>>> sys.path
['', '/usr/local/lib/python25.zip', ...]
```

The `sys.path` is a list where to look for modules, starting at the first element in the list, usually the current directory, to change this:

```
>>> sys.path.insert(0, '/tmp')
```

[Open Source Software](#)

licensing software  
Copyright and CopyLeft

[Python](#)

More on Modules  
Packages

[Numerical Python: NumPy](#)

an open source project

[Summary + Assignments](#)

# Module Search Path

MCS 260 L-20

12 October 2007

The module `sys` is built-in, after

```
>>> import sys
>>> help(sys)
```

The description starts with

This module provides access to some objects used or maintained by the interpreter and to functions that interact strongly with the interpreter.

Where does Python find the modules?

```
>>> sys.path
['', '/usr/local/lib/python25.zip', ...]
```

The `sys.path` is a list where to look for modules, starting at the first element in the list, usually the current directory, to change this:

```
>>> sys.path.insert(0, '/tmp')
```

[Open Source Software](#)

licensing software  
Copyright and CopyLeft

[Python](#)

More on Modules  
Packages

[Numerical Python: NumPy](#)

an open source project

[Summary + Assignments](#)

# Module Built-in Functions

`dir()`

The `dir()` function has the specification:

**input:** the name of a module and

**output:** a sorted list of strings with the names defined by the module.

After `import math`, followed by `dir(math)` we see

1. the usual mathematical functions
2. mathematical constants `math.e` and `math.pi`
3. attributes `__doc__`, `__file__`, and `__name__`.

The three attributes in the module `math`:

`math.__doc__` a documentation string

`math.__file__` location for the static object

`math.__name__` name of the module (without path)

[Open Source Software](#)

licensing software  
Copyright and CopyLeft

[Python](#)

More on Modules  
Packages

[Numerical Python:](#)

[NumPy](#)

an open source project

[Summary + Assignments](#)

# Module Built-in Functions

`dir()`

The `dir()` function has the specification:

input: the name of a module and

output: a sorted list of strings with the names defined by the module.

After `import math`, followed by `dir(math)` we see

1. the usual mathematical functions
2. mathematical constants `math.e` and `math.pi`
3. attributes `__doc__`, `__file__`, and `__name__`.

The three attributes in the module `math`:

`math.__doc__` a documentation string

`math.__file__` location for the static object

`math.__name__` name of the module (without path)

[Open Source Software](#)

licensing software  
Copyright and CopyLeft

[Python](#)

More on Modules  
Packages

[Numerical Python: NumPy](#)

an open source project

[Summary + Assignments](#)

# Module Built-in Functions

`dir()`

The `dir()` function has the specification:

input: the name of a module and

output: a sorted list of strings with the names defined by the module.

After `import math`, followed by `dir(math)` we see

1. the usual mathematical functions
2. mathematical constants `math.e` and `math.pi`
3. attributes `__doc__`, `__file__`, and `__name__`.

The three attributes in the module `math`:

`math.__doc__` a documentation string

`math.__file__` location for the static object

`math.__name__` name of the module (without path)

[Open Source Software](#)

licensing software  
Copyright and CopyLeft

[Python](#)

More on Modules  
Packages

[Numerical Python: NumPy](#)

an open source project

[Summary + Assignments](#)

# Module Built-in Functions

`dir()`

The `dir()` function has the specification:

input: the name of a module and

output: a sorted list of strings with the names defined by the module.

After `import math`, followed by `dir(math)` we see

1. the usual mathematical functions
2. mathematical constants `math.e` and `math.pi`
3. attributes `__doc__`, `__file__`, and `__name__`.

The three attributes in the module `math`:

`math.__doc__` a documentation string

`math.__file__` location for the static object

`math.__name__` name of the module (without path)

[Open Source Software](#)

licensing software  
Copyright and CopyLeft

[Python](#)

More on Modules  
Packages

[Numerical Python: NumPy](#)

an open source project

[Summary + Assignments](#)

# Module Built-in Functions

`dir()`

The `dir()` function has the specification:

input: the name of a module and

output: a sorted list of strings with the names defined by the module.

After `import math`, followed by `dir(math)` we see

1. the usual mathematical functions
2. mathematical constants `math.e` and `math.pi`
3. attributes `__doc__`, `__file__`, and `__name__`.

The three attributes in the module `math`:

`math.__doc__` a documentation string

`math.__file__` location for the static object

`math.__name__` name of the module (without path)

[Open Source Software](#)

licensing software  
Copyright and CopyLeft

[Python](#)

More on Modules  
Packages

[Numerical Python: NumPy](#)

an open source project

[Summary + Assignments](#)

# Module Built-in Functions

`dir()`

The `dir()` function has the specification:

input: the name of a module and

output: a sorted list of strings with the names defined by the module.

After `import math`, followed by `dir(math)` we see

1. the usual mathematical functions
2. mathematical constants `math.e` and `math.pi`
3. attributes `__doc__`, `__file__`, and `__name__`.

The three attributes in the module `math`:

`math.__doc__` a documentation string

`math.__file__` location for the static object

`math.__name__` name of the module (without path)

[Open Source Software](#)

licensing software  
Copyright and CopyLeft

[Python](#)

More on Modules  
Packages

[Numerical Python: NumPy](#)

an open source project

[Summary + Assignments](#)

# Module Built-in Functions

`dir()`

The `dir()` function has the specification:

input: the name of a module and

output: a sorted list of strings with the names defined by the module.

After `import math`, followed by `dir(math)` we see

1. the usual mathematical functions
2. mathematical constants `math.e` and `math.pi`
3. attributes `__doc__`, `__file__`, and `__name__`.

The three attributes in the module `math`:

`math.__doc__` a documentation string

`math.__file__` location for the static object

`math.__name__` name of the module (without path)

[Open Source Software](#)

licensing software  
Copyright and CopyLeft

[Python](#)

More on Modules  
Packages

[Numerical Python:](#)

[NumPy](#)

an open source project

[Summary + Assignments](#)

# Module Built-in Functions

## locals() and globals()

The functions `locals()` and `globals()` take no argument. They return local and global names.

On return are dictionaries, the keys are strings.

Application: see if a variable is already defined:

```
>>> 'x' in globals()
False
>>> x = 2
>>> 'x' in globals()
True
```

The first we see in both is `__builtins__`:

- ▶ The module `__builtins__` is always loaded first when we launch Python.
- ▶ Type `help(__builtins__)` to see more.

[Open Source Software](#)

licensing software  
Copyright and CopyLeft

[Python](#)

[More on Modules Packages](#)

[Numerical Python:](#)

[NumPy](#)

an open source project

[Summary + Assignments](#)

# Module Built-in Functions

## locals() and globals()

The functions `locals()` and `globals()` take no argument. They return local and global names.

On return are dictionaries, the keys are strings.

Application: see if a variable is already defined:

```
>>> 'x' in globals()
False
>>> x = 2
>>> 'x' in globals()
True
```

The first we see in both is `__builtins__`:

- ▶ The module `__builtins__` is always loaded first when we launch Python.
- ▶ Type `help(__builtins__)` to see more.

[Open Source Software](#)

licensing software  
Copyright and CopyLeft

[Python](#)

More on Modules  
Packages

[Numerical Python: NumPy](#)

an open source project

[Summary + Assignments](#)

# Module Built-in Functions

## locals() and globals()

The functions `locals()` and `globals()` take no argument. They return local and global names. On return are dictionaries, the keys are strings. Application: see if a variable is already defined:

```
>>> 'x' in globals()
False
>>> x = 2
>>> 'x' in globals()
True
```

The first we see in both is `__builtins__`:

- ▶ The module `__builtins__` is always loaded first when we launch Python.
- ▶ Type `help(__builtins__)` to see more.

[Open Source Software](#)

licensing software  
Copyright and CopyLeft

[Python](#)

More on Modules  
Packages

[Numerical Python:](#)

[NumPy](#)  
an open source project

[Summary + Assignments](#)

# Module Built-in Functions

## locals() and globals()

The functions `locals()` and `globals()` take no argument. They return local and global names. On return are dictionaries, the keys are strings. Application: see if a variable is already defined:

```
>>> 'x' in globals()
False
>>> x = 2
>>> 'x' in globals()
True
```

The first we see in both is `__builtins__`:

- ▶ The module `__builtins__` is always loaded first when we launch Python.
- ▶ Type `help(__builtins__)` to see more.

[Open Source Software](#)

licensing software  
Copyright and CopyLeft

[Python](#)

More on Modules  
Packages

[Numerical Python: NumPy](#)

an open source project

[Summary + Assignments](#)

# Outline

## Open Source Software

licensing software

Copyright and CopyLeft

## Python

More on Modules

Packages

## Numerical Python: NumPy

an open source project

## Summary + Assignments

MCS 260 L-20

12 October 2007

### Open Source Software

licensing software  
Copyright and CopyLeft

### Python

More on Modules  
**Packages**

### Numerical Python: NumPy

an open source project

### Summary + Assignments

# Packages

hierarchical file directory structure

Modules in larger software systems are decomposed again into modules.

Python provides multi-modules packages, with naming as

```
< package name > . < module name >
```

NumPy is a multi-module package:

```
>>> import numpy
>>> import numpy.linalg
```

The `linalg` provides linear algebra operations.

[Open Source Software](#)

licensing software  
Copyright and CopyLeft

[Python](#)

More on Modules  
**Packages**

[Numerical Python: NumPy](#)

an open source project

[Summary + Assignments](#)

# Packages

hierarchical file directory structure

Modules in larger software systems are decomposed again into modules.

Python provides multi-modules packages, with naming as

```
< package name > . < module name >
```

NumPy is a multi-module package:

```
>>> import numpy
>>> import numpy.linalg
```

The `linalg` provides linear algebra operations.

[Open Source Software](#)

licensing software  
Copyright and CopyLeft

[Python](#)

More on Modules  
Packages

[Numerical Python: NumPy](#)

an open source project

[Summary + Assignments](#)

# Packages

hierarchical file directory structure

Modules in larger software systems are decomposed again into modules.

Python provides multi-modules packages, with naming as

```
< package name > . < module name >
```

NumPy is a multi-module package:

```
>>> import numpy
>>> import numpy.linalg
```

The `linalg` provides linear algebra operations.

[Open Source Software](#)

licensing software  
Copyright and CopyLeft

[Python](#)

More on Modules  
Packages

[Numerical Python: NumPy](#)

an open source project

[Summary + Assignments](#)

# Outline

## Open Source Software

licensing software

Copyright and Copyleft

## Python

More on Modules

Packages

## Numerical Python: NumPy

an open source project

## Summary + Assignments

MCS 260 L-20

12 October 2007

### Open Source Software

licensing software  
Copyright and Copyleft

### Python

More on Modules  
Packages

### Numerical Python: NumPy

an open source project

### Summary + Assignments

# NumPy

## Numerical Python

Available for download via

<http://numpy.scipy.org/>.

Python has no array types or methods to deal with matrices, which are essential for scientific computing.

```
>>> from numpy import array
>>> A = array([[1,2],[3,4]])
>>> A
array([[1, 2],
       [3, 4]])
>>> from numpy.linalg import det
>>> det(A)
-2.0
```

Computing just like in Maple, MATLAB, or Octave.

# NumPy

## Numerical Python

Available for download via  
<http://numpy.scipy.org/>.

Python has no array types or methods to deal with matrices, which are essential for scientific computing.

```
>>> from numpy import array
>>> A = array([[1,2],[3,4]])
>>> A
array([[1, 2],
       [3, 4]])
>>> from numpy.linalg import det
>>> det(A)
-2.0
```

Computing just like in Maple, MATLAB, or Octave.

# NumPy

## Numerical Python

Available for download via  
<http://numpy.scipy.org/>.

Python has no array types or methods to deal with matrices, which are essential for scientific computing.

```
>>> from numpy import array
>>> A = array([[1,2],[3,4]])
>>> A
array([[1, 2],
       [3, 4]])
>>> from numpy.linalg import det
>>> det(A)
-2.0
```

Computing just like in Maple, MATLAB, or Octave.

Available for download via  
<http://numpy.scipy.org/>.

Python has no array types or methods to deal with matrices, which are essential for scientific computing.

```
>>> from numpy import array
>>> A = array([[1,2],[3,4]])
>>> A
array([[1, 2],
       [3, 4]])
>>> from numpy.linalg import det
>>> det(A)
-2.0
```

Computing just like in Maple, MATLAB, or Octave.

[Open Source Software](#)

licensing software  
Copyright and CopyLeft

[Python](#)

More on Modules  
Packages

[Numerical Python: NumPy](#)

an open source project

[Summary + Assignments](#)

Suppose we want to solve

$$\begin{cases} x_1 + 2x_2 = 5 \\ 3x_1 + 4x_2 = 6 \end{cases}$$

First we define the arrays:

```
>>> from numpy import array
>>> A = array([[1,2],[3,4]])
>>> b = array([[5],[6]])
```

and then we apply the solve:

```
>>> from numpy.linalg import solve
>>> solve(A,b)
array([[ -4. ],
       [  4.5]])
```

[Open Source Software](#)

licensing software  
Copyright and CopyLeft

[Python](#)

More on Modules  
Packages

[Numerical Python: NumPy](#)

an open source project

[Summary + Assignments](#)

Suppose we want to solve

$$\begin{cases} x_1 + 2x_2 = 5 \\ 3x_1 + 4x_2 = 6 \end{cases}$$

First we define the arrays:

```
>>> from numpy import array
>>> A = array([[1,2],[3,4]])
>>> b = array([[5],[6]])
```

and then we apply the solve:

```
>>> from numpy.linalg import solve
>>> solve(A,b)
array([[ -4. ],
       [ 4.5]])
```

[Open Source Software](#)

licensing software  
Copyright and CopyLeft

[Python](#)

More on Modules  
Packages

[Numerical Python: NumPy](#)

an open source project

[Summary + Assignments](#)

Suppose we want to solve

$$\begin{cases} x_1 + 2x_2 = 5 \\ 3x_1 + 4x_2 = 6 \end{cases}$$

First we define the arrays:

```
>>> from numpy import array
>>> A = array([[1,2],[3,4]])
>>> b = array([[5],[6]])
```

and then we apply the solve:

```
>>> from numpy.linalg import solve
>>> solve(A,b)
array([[ -4. ],
       [  4.5]])
```

[Open Source Software](#)

licensing software  
Copyright and CopyLeft

[Python](#)

More on Modules  
Packages

[Numerical Python: NumPy](#)

an open source project

[Summary + Assignments](#)

# Summary + Assignments

MCS 260 L-20

12 October 2007

We covered more of chapter 6 in *Making Use of Python*; visit also [www.opensource.org](http://www.opensource.org) and [www.gnu.org](http://www.gnu.org).

Assignments:

1. Where can you find the Python license on your computer? Compare the Python license with GNU GPL and summarize the similarities and differences.
2. Give the Python commands to remove the first element of the module search path. Show also the output of the commands.
3. Download and install NumPy. Solve a linear system and verify its answer.

**Homework will be collected on Monday 15 October:** exercises 1 and 3 of Lecture 14; exercise 1 of Lecture 15; and exercises 1 and 2 of lecture 19.

Open Source  
Software

licensing software  
Copyright and CopyLeft

Python

More on Modules  
Packages

Numerical Python:  
NumPy

an open source project

Summary +  
Assignments