

Internet  
Basics

evolution of the web  
IP addresses and  
URLs  
client/server and  
HTTP

Markup  
Languages

HTML, XML, MathML  
MathML generated  
by Maple

Retrieving  
Data

the weather forecast

CGI  
Programming

showing current date  
and time in browser

Summary +  
Assignments

- 1 Internet Basics
  - evolution of the web
  - IP addresses and URLs
  - client/server and HTTP
- 2 Markup Languages
  - HTML, XML, MathML
  - MathML generated by Maple
- 3 Retrieving Data
  - the weather forecast
- 4 CGI Programming
  - showing current date and time in browser
- 5 Summary + Assignments

MCS 260 Lecture 18  
Introduction to Computer Science  
Jan Vershelde, 22 February 2010

# networking and the internet markup languages

## Internet Basics

### evolution of the web

IP addresses and  
URLs

client/server and  
HTTP

## Markup Languages

HTML, XML, MathML

MathML generated  
by Maple

## Retrieving Data

the weather forecast

## CGI Programming

showing current date  
and time in browser

## Summary + Assignments

### 1 Internet Basics

evolution of the web

IP addresses and URLs

client/server and HTTP

### 2 Markup Languages

HTML, XML, MathML

MathML generated by Maple

### 3 Retrieving Data

the weather forecast

### 4 CGI Programming

showing current date and time in browser

### 5 Summary + Assignments

# The World Wide Web

## WWW: historical development

- **1970s: development of TCP/IP = Transmission Control Protocol/Internet Protocol.**  
**Main benefit: capability of electronic mail (*email*).**
- mid 1980s: connections between computer facilities.  
In 1989, Tim Berners-Lee of CERN developed HTML, HyperText Markup Language.
- Mosaic was the first web browser developed at NCSA, released in 1993, leading to Netscape.  
Search engines originated at the end of the nineties.
- Web 2.0: publishing → participation.  
*Wiki* is server software that allows users to freely create and edit Web pages using any Web browser.

# The World Wide Web

## WWW: historical development

- 1970s: development of TCP/IP = Transmission Control Protocol/Internet Protocol.  
Main benefit: capability of electronic mail (*email*).
- mid 1980s: connections between computer facilities.  
In 1989, Tim Berners-Lee of CERN developed HTML, HyperText Markup Language.
- Mosaic was the first web browser developed at NCSA, released in 1993, leading to Netscape.  
Search engines originated at the end of the nineties.
- Web 2.0: publishing → participation.  
*Wiki* is server software that allows users to freely create and edit Web pages using any Web browser.

# The World Wide Web

## WWW: historical development

- 1970s: development of TCP/IP = Transmission Control Protocol/Internet Protocol.  
Main benefit: capability of electronic mail (*email*).
- mid 1980s: connections between computer facilities.  
In 1989, Tim Berners-Lee of CERN developed HTML, HyperText Markup Language.
- Mosaic was the first web browser developed at NCSA, released in 1993, leading to Netscape.  
Search engines originated at the end of the nineties.
- Web 2.0: publishing → participation.  
*Wiki* is server software that allows users to freely create and edit Web pages using any Web browser.

# The World Wide Web

## WWW: historical development

- 1970s: development of TCP/IP = Transmission Control Protocol/Internet Protocol.  
Main benefit: capability of electronic mail (*email*).
- mid 1980s: connections between computer facilities.  
In 1989, Tim Berners-Lee of CERN developed HTML, HyperText Markup Language.
- Mosaic was the first web browser developed at NCSA, released in 1993, leading to Netscape.  
Search engines originated at the end of the nineties.
- Web 2.0: publishing → participation.  
*Wiki* is server software that allows users to freely create and edit Web pages using any Web browser.

# networking and the internet

## markup languages

### Internet Basics

evolution of the web

IP addresses and URLs

client/server and HTTP

### Markup Languages

HTML, XML, MathML

MathML generated by Maple

### Retrieving Data

the weather forecast

### CGI Programming

showing current date and time in browser

### Summary + Assignments

## 1 Internet Basics

evolution of the web

IP addresses and URLs

client/server and HTTP

## 2 Markup Languages

HTML, XML, MathML

MathML generated by Maple

## 3 Retrieving Data

the weather forecast

## 4 CGI Programming

showing current date and time in browser

## 5 Summary + Assignments

# Network Addresses

and domain names

Each node on the internet has an *IP address*.

An IP address consists of four bytes.

Each node has a symbolic name.

For example, `icarus.cc.uic.edu`.

The `edu` stands for universities. The other principal domains are `gov` (government), `mil` (military), `com` (commercial), and `org` (other organizations).

The command `nslookup icarus.cc.uic.edu`  
or `host icarus.cc.uic.edu` returns  
the numerical IP address for `icarus`: `128.248.155.80`

The Internet is a *Wide Area Network* (WAN), linking machines over a greater distance. A *Local Area Network* (LAN), links computers in one room or building.

# Network Addresses

## and domain names

Each node on the internet has an *IP address*.

An IP address consists of four bytes.

Each node has a symbolic name.

For example, `icarus.cc.uic.edu`.

The `edu` stands for universities. The other principal domains are `gov` (government), `mil` (military), `com` (commercial), and `org` (other organizations).

The command `nslookup icarus.cc.uic.edu`  
or `host icarus.cc.uic.edu` returns  
the numerical IP address for `icarus`: `128.248.155.80`

The Internet is a *Wide Area Network* (WAN), linking machines over a greater distance. A *Local Area Network* (LAN), links computers in one room or building.

# Network Addresses

## and domain names

Each node on the internet has an *IP address*.

An IP address consists of four bytes.

Each node has a symbolic name.

For example, `icarus.cc.uic.edu`.

The `edu` stands for universities. The other principal domains are `gov` (government), `mil` (military), `com` (commercial), and `org` (other organizations).

The command `nslookup icarus.cc.uic.edu`  
or `host icarus.cc.uic.edu` returns  
the numerical IP address for `icarus`: `128.248.155.80`

The Internet is a *Wide Area Network* (WAN), linking machines over a greater distance. A *Local Area Network* (LAN), links computers in one room or building.

# Network Addresses

and domain names

## Internet Basics

evolution of the web

IP addresses and URLs

client/server and HTTP

## Markup Languages

HTML, XML, MathML

MathML generated by Maple

## Retrieving Data

the weather forecast

## CGI Programming

showing current date and time in browser

## Summary + Assignments

Each node on the internet has an *IP address*.

An IP address consists of four bytes.

Each node has a symbolic name.

For example, `icarus.cc.uic.edu`.

The `edu` stands for universities. The other principal domains are `gov` (government), `mil` (military), `com` (commercial), and `org` (other organizations).

The command `nslookup icarus.cc.uic.edu` or `host icarus.cc.uic.edu` returns the numerical IP address for `icarus`: `128.248.155.80`

The Internet is a *Wide Area Network* (WAN), linking machines over a greater distance. A *Local Area Network* (LAN), links computers in one room or building.

# Uniform Resource Locator (URL)

## Internet Basics

evolution of the web

IP addresses and URLs

client/server and HTTP

## Markup Languages

HTML, XML, MathML

MathML generated by Maple

## Retrieving Data

the weather forecast

## CGI Programming

showing current date and time in browser

## Summary + Assignments

A URL is an addressing scheme to provide a path to an internet resource.

Example:

```
http://www.math.uic.edu/~jan/mcs260.html.
```

The format of a URL is

```
protocol://host.domain-name/path/dataname
```

where

- `protocol` refers to the type of protocol to be used
- `host` refers to the server where the resource is stored
- `domain-name` contains the name and type of the domain of the server
- `path/dataname` refers to the location of the data

To preview pages offline, use the protocol `file`.

# Uniform Resource Locator (URL)

## Internet Basics

evolution of the web

IP addresses and URLs

client/server and HTTP

## Markup Languages

HTML, XML, MathML

MathML generated by Maple

## Retrieving Data

the weather forecast

## CGI Programming

showing current date and time in browser

## Summary + Assignments

A URL is an addressing scheme to provide a path to an internet resource.

**Example:**

```
http://www.math.uic.edu/~jan/mcs260.html.
```

The format of a URL is

```
protocol://host.domain-name/path/dataname
```

where

- `protocol` refers to the type of protocol to be used
- `host` refers to the server where the resource is stored
- `domain-name` contains the name and type of the domain of the server
- `path/dataname` refers to the location of the data

To preview pages offline, use the protocol `file`.

# Uniform Resource Locator (URL)

## Internet Basics

evolution of the web

IP addresses and URLs

client/server and HTTP

## Markup Languages

HTML, XML, MathML

MathML generated by Maple

## Retrieving Data

the weather forecast

## CGI Programming

showing current date and time in browser

## Summary + Assignments

A URL is an addressing scheme to provide a path to an internet resource.

**Example:**

```
http://www.math.uic.edu/~jan/mcs260.html.
```

**The format of a URL is**

```
protocol://host.domain-name/path/dataname
```

**where**

- `protocol` refers to the type of protocol to be used
- `host` refers to the server where the resource is stored
- `domain-name` contains the name and type of the domain of the server
- `path/dataname` refers to the location of the data

To preview pages offline, use the protocol `file`.

# Uniform Resource Locator (URL)

## Internet Basics

evolution of the web

IP addresses and URLs

client/server and HTTP

## Markup Languages

HTML, XML, MathML

MathML generated by Maple

## Retrieving Data

the weather forecast

## CGI Programming

showing current date and time in browser

## Summary + Assignments

A URL is an addressing scheme to provide a path to an internet resource.

**Example:**

```
http://www.math.uic.edu/~jan/mcs260.html.
```

The format of a URL is

```
protocol://host.domain-name/path/dataname
```

where

- `protocol` refers to the type of protocol to be used
- `host` refers to the server where the resource is stored
- `domain-name` contains the name and type of the domain of the server
- `path/dataname` refers to the location of the data

To preview pages offline, use the protocol `file`.

# Uniform Resource Locator (URL)

## Internet Basics

evolution of the web

IP addresses and URLs

client/server and HTTP

## Markup Languages

HTML, XML, MathML

MathML generated by Maple

## Retrieving Data

the weather forecast

## CGI Programming

showing current date and time in browser

## Summary + Assignments

A URL is an addressing scheme to provide a path to an internet resource.

**Example:**

```
http://www.math.uic.edu/~jan/mcs260.html.
```

The format of a URL is

```
protocol://host.domain-name/path/dataname
```

where

- `protocol` refers to the type of protocol to be used
- `host` refers to the server where the resource is stored
- `domain-name` contains the name and type of the domain of the server
- `path/dataname` refers to the location of the data

To preview pages offline, use the protocol `file`.

# Uniform Resource Locator (URL)

## Internet Basics

evolution of the web

IP addresses and URLs

client/server and HTTP

## Markup Languages

HTML, XML, MathML

MathML generated by Maple

## Retrieving Data

the weather forecast

## CGI Programming

showing current date and time in browser

## Summary + Assignments

A URL is an addressing scheme to provide a path to an internet resource.

Example:

```
http://www.math.uic.edu/~jan/mcs260.html.
```

The format of a URL is

```
protocol://host.domain-name/path/dataname
```

where

- `protocol` refers to the type of protocol to be used
- `host` refers to the server where the resource is stored
- `domain-name` contains the name and type of the domain of the server
- `path/dataname` refers to the location of the data

To preview pages offline, use the protocol `file`.

# Uniform Resource Locator (URL)

## Internet Basics

evolution of the web  
IP addresses and URLs

client/server and HTTP

## Markup Languages

HTML, XML, MathML  
MathML generated by Maple

## Retrieving Data

the weather forecast

## CGI Programming

showing current date and time in browser

## Summary + Assignments

A URL is an addressing scheme to provide a path to an internet resource.

Example:

```
http://www.math.uic.edu/~jan/mcs260.html.
```

The format of a URL is

```
protocol://host.domain-name/path/dataname
```

where

- `protocol` refers to the type of protocol to be used
- `host` refers to the server where the resource is stored
- `domain-name` contains the name and type of the domain of the server
- `path/dataname` refers to the location of the data

To preview pages offline, use the protocol `file`.

# Uniform Resource Locator (URL)

## Internet Basics

evolution of the web  
IP addresses and URLs

client/server and HTTP

## Markup Languages

HTML, XML, MathML  
MathML generated by Maple

## Retrieving Data

the weather forecast

## CGI Programming

showing current date and time in browser

## Summary + Assignments

A URL is an addressing scheme to provide a path to an internet resource.

Example:

```
http://www.math.uic.edu/~jan/mcs260.html.
```

The format of a URL is

```
protocol://host.domain-name/path/dataname
```

where

- `protocol` refers to the type of protocol to be used
- `host` refers to the server where the resource is stored
- `domain-name` contains the name and type of the domain of the server
- `path/dataname` refers to the location of the data

To preview pages offline, use the protocol `file`.

# networking and the internet markup languages

## Internet Basics

evolution of the web  
IP addresses and  
URLs

**client/server and  
HTTP**

## Markup Languages

HTML, XML, MathML  
MathML generated  
by Maple

## Retrieving Data

the weather forecast

## CGI Programming

showing current date  
and time in browser

## Summary + Assignments

### 1 Internet Basics

evolution of the web  
IP addresses and URLs  
**client/server and HTTP**

### 2 Markup Languages

HTML, XML, MathML  
MathML generated by Maple

### 3 Retrieving Data

the weather forecast

### 4 CGI Programming

showing current date and time in browser

### 5 Summary + Assignments

# Client/Server Networks

## Internet Basics

evolution of the web

IP addresses and  
URLs

client/server and  
HTTP

## Markup Languages

HTML, XML, MathML

MathML generated  
by Maple

## Retrieving Data

the weather forecast

## CGI Programming

showing current date  
and time in browser

## Summary + Assignments

A *client* is a computer in the network that *requests* for access to data and services from another computer.

A *server* is a computer in the network *receives and processes* requests from clients.

Access permissions are determined by the server.

A *client/server network* consists of several computers connected in a network, acting as clients and/or servers.

Client/server computing emerged in the nineties to distribute applications (such as database administration) over a network.

# Client/Server Networks

## Internet Basics

evolution of the web

IP addresses and  
URLs

client/server and  
HTTP

## Markup Languages

HTML, XML, MathML

MathML generated  
by Maple

## Retrieving Data

the weather forecast

## CGI Programming

showing current date  
and time in browser

## Summary + Assignments

A *client* is a computer in the network that *requests* for access to data and services from another computer.

A *server* is a computer in the network *receives and processes* requests from clients.

Access permissions are determined by the server.

A *client/server network* consists of several computers connected in a network, acting as clients and/or servers.

Client/server computing emerged in the nineties to distribute applications (such as database administration) over a network.

# Client/Server Networks

## Internet Basics

evolution of the web  
IP addresses and  
URLs  
client/server and  
HTTP

## Markup Languages

HTML, XML, MathML  
MathML generated  
by Maple

## Retrieving Data

the weather forecast

## CGI Programming

showing current date  
and time in browser

## Summary + Assignments

A *client* is a computer in the network that *requests* for access to data and services from another computer.

A *server* is a computer in the network *receives and processes* requests from clients.

Access permissions are determined by the server.

A *client/server network* consists of several computers connected in a network, acting as clients and/or servers.

Client/server computing emerged in the nineties to distribute applications (such as database administration) over a network.

# Client/Server Networks

## Internet Basics

evolution of the web  
IP addresses and  
URLs  
client/server and  
HTTP

## Markup Languages

HTML, XML, MathML  
MathML generated  
by Maple

## Retrieving Data

the weather forecast

## CGI Programming

showing current date  
and time in browser

## Summary + Assignments

A *client* is a computer in the network that *requests* for access to data and services from another computer.

A *server* is a computer in the network *receives and processes* requests from clients.

Access permissions are determined by the server.

A *client/server network* consists of several computers connected in a network, acting as clients and/or servers.

Client/server computing emerged in the nineties to distribute applications (such as database administration) over a network.

Internet  
Basics

evolution of the web  
IP addresses and  
URLs

client/server and  
HTTP

Markup  
Languages

HTML, XML, MathML  
MathML generated  
by Maple

Retrieving  
Data

the weather forecast

CGI  
Programming

showing current date  
and time in browser

Summary +  
Assignments

# Hypertext Transfer Protocol (HTTP)

exchange data between the client and the server

HTTP is based on request-response between a web browser (the client) and a web server.

A typical transaction between browser and server:

- 1 A TCP/IP connection is established between browser and server.
- 2 The browser sends a request for a web page.
- 3 The server locates the file and responds, sending the content of the requested web page.
- 4 The TCP/IP connection is closed.

Internet  
Basics

evolution of the web  
IP addresses and  
URLs

client/server and  
HTTP

Markup  
Languages

HTML, XML, MathML  
MathML generated  
by Maple

Retrieving  
Data

the weather forecast

CGI  
Programming

showing current date  
and time in browser

Summary +  
Assignments

# Hypertext Transfer Protocol (HTTP)

exchange data between the client and the server

HTTP is based on request-response between a web browser (the client) and a web server.

A typical transaction between browser and server:

- 1 A TCP/IP connection is established between browser and server.
- 2 The browser sends a request for a web page.
- 3 The server locates the file and responds, sending the content of the requested web page.
- 4 The TCP/IP connection is closed.

Internet  
Basics

evolution of the web  
IP addresses and  
URLs

client/server and  
HTTP

Markup  
Languages

HTML, XML, MathML  
MathML generated  
by Maple

Retrieving  
Data

the weather forecast

CGI  
Programming

showing current date  
and time in browser

Summary +  
Assignments

# Hypertext Transfer Protocol (HTTP)

exchange data between the client and the server

HTTP is based on request-response between a web browser (the client) and a web server.

A typical transaction between browser and server:

- 1 A TCP/IP connection is established between browser and server.
- 2 The browser sends a request for a web page.
- 3 The server locates the file and responds, sending the content of the requested web page.
- 4 The TCP/IP connection is closed.

Internet  
Basics

evolution of the web  
IP addresses and  
URLs

client/server and  
HTTP

Markup  
Languages

HTML, XML, MathML  
MathML generated  
by Maple

Retrieving  
Data

the weather forecast

CGI  
Programming

showing current date  
and time in browser

Summary +  
Assignments

# Hypertext Transfer Protocol (HTTP)

exchange data between the client and the server

HTTP is based on request-response between a web browser (the client) and a web server.

A typical transaction between browser and server:

- 1 A TCP/IP connection is established between browser and server.
- 2 The browser sends a request for a web page.
- 3 The server locates the file and responds, sending the content of the requested web page.
- 4 The TCP/IP connection is closed.

# Creating your own homepage

## Internet Basics

evolution of the web

IP addresses and  
URLs

client/server and  
HTTP

## Markup Languages

HTML, XML, MathML

MathML generated  
by Maple

## Retrieving Data

the weather forecast

## CGI Programming

showing current date  
and time in browser

## Summary + Assignments

Use your icarus account.

*Another good reason to learn Unix!*

Personal web pages are stored in the subdirectory  
`public_html` in your homedirectory.

Make sure this file is accessible, by setting permissions  
`chmod a+r *` of all files in `public_html`.

URLs will start as

`http://www2.uic.edu/~your_netid.`

# Creating your own homepage

## Internet Basics

evolution of the web  
IP addresses and  
URLs

client/server and  
HTTP

## Markup Languages

HTML, XML, MathML  
MathML generated  
by Maple

## Retrieving Data

the weather forecast

## CGI Programming

showing current date  
and time in browser

## Summary + Assignments

Use your icarus account.

*Another good reason to learn Unix!*

Personal web pages are stored in the subdirectory  
`public_html` in your homedirectory.

Make sure this file is accessible, by setting permissions  
`chmod a+r *` of all files in `public_html`.

URLs will start as

`http://www2.uic.edu/~your_netid.`

# Creating your own homepage

## Internet Basics

evolution of the web

IP addresses and  
URLs

client/server and  
HTTP

## Markup Languages

HTML, XML, MathML

MathML generated  
by Maple

## Retrieving Data

the weather forecast

## CGI Programming

showing current date  
and time in browser

## Summary + Assignments

Use your `icarus` account.

*Another good reason to learn Unix!*

Personal web pages are stored in the subdirectory  
`public_html` in your homedirectory.

Make sure this file is accessible, by setting permissions  
`chmod a+r *` of all files in `public_html`.

URLs will start as

`http://www2.uic.edu/~your_netid.`

# Creating your own homepage

## Internet Basics

evolution of the web  
IP addresses and  
URLs

client/server and  
HTTP

## Markup Languages

HTML, XML, MathML  
MathML generated  
by Maple

## Retrieving Data

the weather forecast

## CGI Programming

showing current date  
and time in browser

## Summary + Assignments

Use your `icarus` account.

*Another good reason to learn Unix!*

Personal web pages are stored in the subdirectory  
`public_html` in your homedirectory.

Make sure this file is accessible, by setting permissions  
`chmod a+r *` of all files in `public_html`.

URLs will start as

`http://www2.uic.edu/~your_netid`.

# networking and the internet markup languages

## Internet Basics

evolution of the web  
IP addresses and  
URLs  
client/server and  
HTTP

## Markup Languages

HTML, XML, MathML  
MathML generated  
by Maple

## Retrieving Data

the weather forecast

## CGI Programming

showing current date  
and time in browser

## Summary + Assignments

### 1 Internet Basics

evolution of the web  
IP addresses and URLs  
client/server and HTTP

### 2 Markup Languages

HTML, XML, MathML  
MathML generated by Maple

### 3 Retrieving Data

the weather forecast

### 4 CGI Programming

showing current date and time in browser

### 5 Summary + Assignments

# Markup Languages

to write web pages

To write web pages, we use

**HTML** hypertext markup language  
written to display information, the language in  
which web pages are written.

**XML** extensible markup language  
XML is a widely supported open technology for  
describing data.

**MathML** mathematical markup language  
MathML can display complex mathematical  
expressions. As it is created with XML,  
MathML is a so-called *XML vocabulary*.

The world wide web consortium (<http://www.w3c.org>)  
is a source for many protocol standards for the web.

# Markup Languages

to write web pages

To write web pages, we use

**HTML** hypertext markup language  
written to display information, the language in  
which web pages are written.

**XML** extensible markup language  
XML is a widely supported open technology for  
describing data.

**MathML** mathematical markup language  
MathML can display complex mathematical  
expressions. As it is created with XML,  
MathML is a so-called *XML vocabulary*.

The world wide web consortium (<http://www.w3c.org>)  
is a source for many protocol standards for the web.

# Markup Languages

to write web pages

To write web pages, we use

**HTML** hypertext markup language  
written to display information, the language in  
which web pages are written.

**XML** extensible markup language  
XML is a widely supported open technology for  
describing data.

**MathML** mathematical markup language  
MathML can display complex mathematical  
expressions. As it is created with XML,  
MathML is a so-called *XML vocabulary*.

The world wide web consortium (<http://www.w3c.org>)  
is a source for many protocol standards for the web.

# Our First Web Page

a sample of HTML: `hello_world.html`

```
<HTML>
```

```
<HEAD>
```

```
<TITLE> MCS 260 Lec 18: hello world </TITLE>
```

```
</HEAD>
```

```
<BODY>
```

```
<H1> Hello World! </H1>
```

```
This is our <EM>first</EM> web page!
```

```
<P>
```

```
To go to <A HREF="http://www.uic.edu">  
UIC's home page </A>, click on the  
underlined text.
```

```
</BODY>
```

```
</HTML>
```

# XML to exchange data

a sample of XML: `hello_world.xml`

## XML focuses on data, not its formatting

```
<xml>
<head>our first xml example</head>
<body>
<article>
<title>hello world</title>
<date>22 February 2010</date>
<href>http://www.uic.edu</href>
<content>This is our first XML example.</content>
</article>
</body>
</xml>
```

A browser will display *the document tree*.

XML + HTML = XHTML

# XML to exchange data

a sample of XML: `hello_world.xml`

## Internet Basics

evolution of the web  
IP addresses and URLs  
client/server and HTTP

## Markup Languages

HTML, XML, MathML  
MathML generated by Maple

## Retrieving Data

the weather forecast

## CGI Programming

showing current date and time in browser

## Summary + Assignments

## XML focuses on data, not its formatting

```
<xml>
<head>our first xml example</head>
<body>
<article>
<title>hello world</title>
<date>22 February 2010</date>
<href>http://www.uic.edu</href>
<content>This is our first XML example.</content>
</article>
</body>
</xml>
```

A browser will display *the document tree*.

XML + HTML = XHTML

# XML to exchange data

a sample of XML: `hello_world.xml`

## Internet Basics

evolution of the web  
IP addresses and URLs  
client/server and HTTP

## Markup Languages

HTML, XML, MathML  
MathML generated by Maple

## Retrieving Data

the weather forecast

## CGI Programming

showing current date and time in browser

## Summary + Assignments

## XML focuses on data, not its formatting

```
<xml>
<head>our first xml example</head>
<body>
<article>
<title>hello world</title>
<date>22 February 2010</date>
<href>http://www.uic.edu</href>
<content>This is our first XML example.</content>
</article>
</body>
</xml>
```

A browser will display *the document tree*.

XML + HTML = XHTML

# XML to exchange data

a sample of XML: `hello_world.xml`

## Internet Basics

evolution of the web  
IP addresses and URLs  
client/server and HTTP

## Markup Languages

HTML, XML, MathML  
MathML generated by Maple

## Retrieving Data

the weather forecast

## CGI Programming

showing current date and time in browser

## Summary + Assignments

XML focuses on data, not its formatting

```
<xml>
<head>our first xml example</head>
<body>
<article>
<title>hello world</title>
<date>22 February 2010</date>
<href>http://www.uic.edu</href>
<content>This is our first XML example.</content>
</article>
</body>
</xml>
```

A browser will display *the document tree*.

XML + HTML = XHTML

# Document Object Model (DOM)

## Internet Basics

evolution of the web  
IP addresses and URLs  
client/server and HTTP

## Markup Languages

HTML, XML, MathML  
MathML generated by Maple

## Retrieving Data

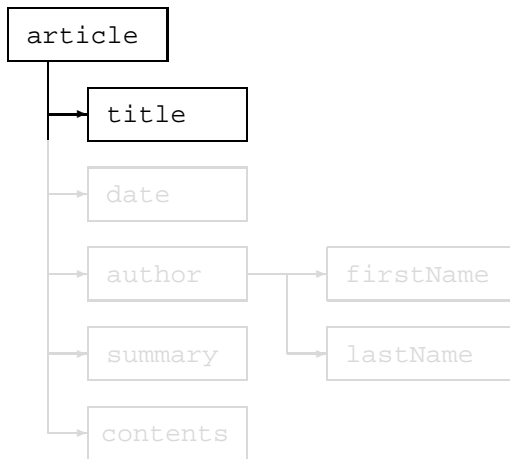
the weather forecast

## CGI Programming

showing current date and time in browser

## Summary + Assignments

The tree structure for `article.xml`:



# Document Object Model (DOM)

## Internet Basics

evolution of the web  
IP addresses and URLs  
client/server and HTTP

## Markup Languages

HTML, XML, MathML  
MathML generated by Maple

## Retrieving Data

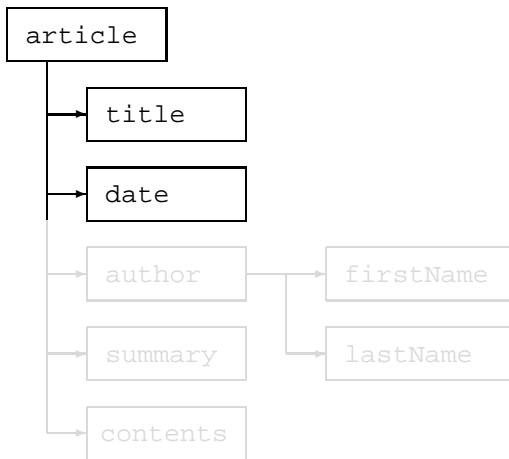
the weather forecast

## CGI Programming

showing current date and time in browser

## Summary + Assignments

The tree structure for `article.xml`:



# Document Object Model (DOM)

## Internet Basics

evolution of the web  
IP addresses and URLs  
client/server and HTTP

## Markup Languages

HTML, XML, MathML  
MathML generated by Maple

## Retrieving Data

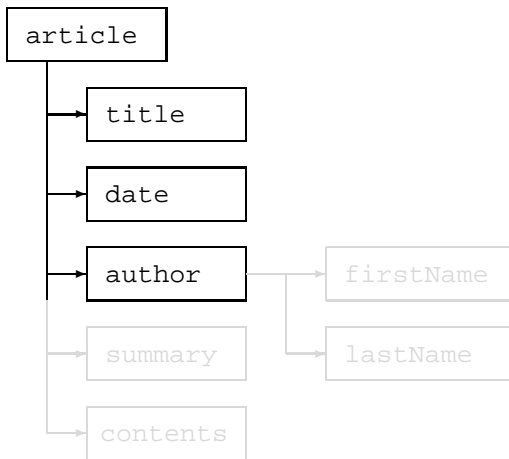
the weather forecast

## CGI Programming

showing current date and time in browser

## Summary + Assignments

The tree structure for `article.xml`:



# Document Object Model (DOM)

## Internet Basics

evolution of the web  
IP addresses and URLs  
client/server and HTTP

## Markup Languages

HTML, XML, MathML  
MathML generated by Maple

## Retrieving Data

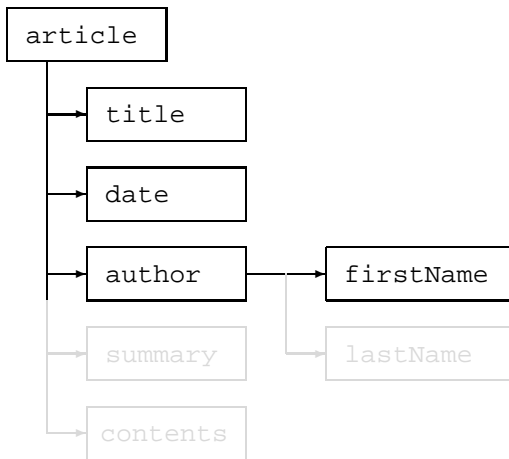
the weather forecast

## CGI Programming

showing current date and time in browser

## Summary + Assignments

The tree structure for `article.xml`:



# Document Object Model (DOM)

## Internet Basics

evolution of the web  
IP addresses and URLs  
client/server and HTTP

## Markup Languages

HTML, XML, MathML  
MathML generated by Maple

## Retrieving Data

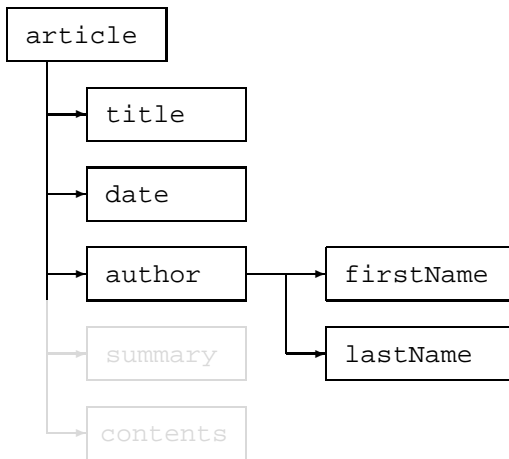
the weather forecast

## CGI Programming

showing current date and time in browser

## Summary + Assignments

The tree structure for `article.xml`:



# Document Object Model (DOM)

## Internet Basics

evolution of the web  
IP addresses and URLs  
client/server and HTTP

## Markup Languages

HTML, XML, MathML  
MathML generated by Maple

## Retrieving Data

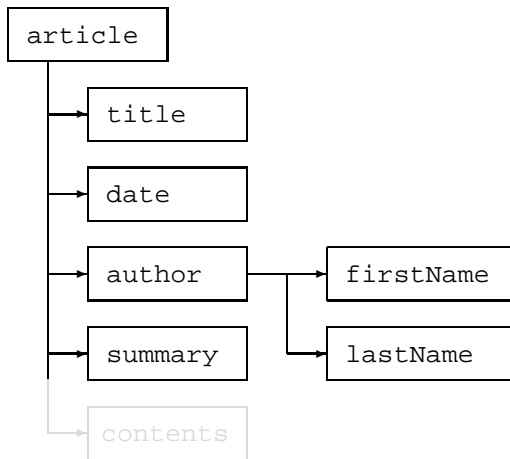
the weather forecast

## CGI Programming

showing current date and time in browser

## Summary + Assignments

The tree structure for `article.xml`:



# Document Object Model (DOM)

## Internet Basics

evolution of the web  
IP addresses and URLs  
client/server and HTTP

## Markup Languages

HTML, XML, MathML  
MathML generated by Maple

## Retrieving Data

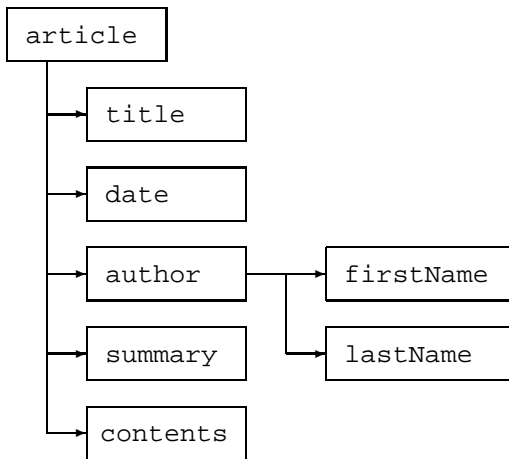
the weather forecast

## CGI Programming

showing current date and time in browser

## Summary + Assignments

The tree structure for `article.xml`:



# networking and the internet markup languages

## Internet Basics

evolution of the web  
IP addresses and  
URLs  
client/server and  
HTTP

## Markup Languages

HTML, XML, MathML  
**MathML generated  
by Maple**

## Retrieving Data

the weather forecast

## CGI Programming

showing current date  
and time in browser

## Summary + Assignments

### 1 Internet Basics

evolution of the web  
IP addresses and URLs  
client/server and HTTP

### 2 Markup Languages

HTML, XML, MathML  
**MathML generated by Maple**

### 3 Retrieving Data

the weather forecast

### 4 CGI Programming

showing current date and time in browser

### 5 Summary + Assignments

# Typesetting Mathematical Expressions

using Maple and MathML

## Internet Basics

evolution of the web  
IP addresses and URLs  
client/server and HTTP

## Markup Languages

HTML, XML, MathML

MathML generated by Maple

## Retrieving Data

the weather forecast

## CGI Programming

showing current date and time in browser

## Summary + Assignments

Mathematical expressions like

$$\int_{-1}^{10} \frac{e^{-x^2} \sin(20x)}{\sqrt{2+x^8}} dx$$

are encoded with  $\text{\LaTeX}$  as

$$\int_{-1}^{10} \frac{e^{-x^2} \sin(20 x)}{\sqrt{2+x^8}} dx$$

With Maple (also to produce  $\text{\LaTeX}$  code):

```
> f := Int(exp(-x^2)*sin(20*x)/sqrt(2+x^8), x=-1..10)
```

Menu: File, Export As, choose HTML with MathML to generate MathML.

# Typesetting Mathematical Expressions

using Maple and MathML

## Internet Basics

evolution of the web  
IP addresses and URLs  
client/server and HTTP

## Markup Languages

HTML, XML, MathML

MathML generated by Maple

## Retrieving Data

the weather forecast

## CGI Programming

showing current date and time in browser

## Summary + Assignments

Mathematical expressions like

$$\int_{-1}^{10} \frac{e^{-x^2} \sin(20x)}{\sqrt{2+x^8}} dx$$

are encoded with  $\text{\LaTeX}$  as

$$\int_{-1}^{10} \frac{e^{-x^2} \sin(20 x)}{\sqrt{2+x^8}} dx$$

With Maple (also to produce  $\text{\LaTeX}$  code):

```
> f := Int(exp(-x^2)*sin(20*x)/sqrt(2+x^8), x=-1..10)
```

Menu: File, Export As, choose HTML with MathML to generate MathML.

## MathML generated by Maple

Internet  
Basics

evolution of the web  
IP addresses and URLs  
client/server and HTTP

Markup  
Languages

HTML, XML, MathML

**MathML generated  
by Maple**

Retrieving  
Data

the weather forecast

CGI  
Programming

showing current date  
and time in browser

Summary +  
Assignments

```

<p align="center">
<math xmlns='http://www.w3.org/1998/Math/MathML'>
  <mrow>
    <mi>f</mi>
  </mrow>
  <mo>:=</mo>
  <mrow>
    <mrow>
      <munderover>
        <mo>&Integral;</mo>
        <mn>-1</mn>
        <mn>10</mn>
      </munderover>
    </mrow>
  </mrow>

```

...

encodes  $f := \int_{-1}^{10} .$

# networking and the internet markup languages

## Internet Basics

evolution of the web  
IP addresses and URLs  
client/server and HTTP

## Markup Languages

HTML, XML, MathML  
MathML generated by Maple

## Retrieving Data

the weather forecast

## CGI Programming

showing current date  
and time in browser

## Summary + Assignments

### 1 Internet Basics

evolution of the web  
IP addresses and URLs  
client/server and HTTP

### 2 Markup Languages

HTML, XML, MathML  
MathML generated by Maple

### 3 Retrieving Data

the weather forecast

### 4 CGI Programming

showing current date and time in browser

### 5 Summary + Assignments

## python forecast.py

Internet  
Basics

evolution of the web  
IP addresses and  
URLs  
client/server and  
HTTP

Markup  
Languages

HTML, XML, MathML  
MathML generated  
by Maple

Retrieving  
Data

the weather forecast

CGI  
Programming

showing current date  
and time in browser

Summary +  
Assignments

```
$ python forecast.py
opening http://weather.noaa.gov/pub/data/forecasts/state/il/ilz013.txt ...
```

TODAY	TUE	WED	THU	FRI	SAT	SUN
NOV 10	NOV 11	NOV 12	NOV 13	NOV 14	NOV 15	NOV 16

CHICAGO DOWNTOWN						
PTCLDY	MOCLDY	CLOUDY	MOCLDY	PTCLDY	MOCLDY	PTCLDY
/40	33/45	38/47	42/50	43/49	38/45	32/44
/10	10/40	70/40	20/30	30/30	30/20	10/10

CHICAGO OHARE						
PTCLDY	MOCLDY	CLOUDY	MOCLDY	PTCLDY	MOCLDY	PTCLDY
/40	33/45	37/46	41/49	42/48	37/45	30/43
/10	10/40	70/40	20/30	30/30	30/20	10/10

# Data from the Web

## Internet Basics

evolution of the web  
IP addresses and  
URLs  
client/server and  
HTTP

## Markup Languages

HTML, XML, MathML  
MathML generated  
by Maple

## Retrieving Data

the weather forecast

## CGI Programming

showing current date  
and time in browser

## Summary + Assignments

The module `urllib` exports `urlopen`,  
`urlopen` returns a file like object.

Template for retrieving data from web pages:

```
from urllib import urlopen
url = ' < internet address > '
f = urlopen(url)
s = f.readline()
```

To get the weather forecast: `http://www.weather.gov/`  
National Oceanic and Atmospheric Administration's  
National Weather Service

The web site provides data for downloading:  
`http://weather.noaa.gov/pub/data/`

# Data from the Web

## Internet Basics

evolution of the web  
IP addresses and  
URLs  
client/server and  
HTTP

## Markup Languages

HTML, XML, MathML  
MathML generated  
by Maple

## Retrieving Data

the weather forecast

## CGI Programming

showing current date  
and time in browser

## Summary + Assignments

The module `urllib` exports `urlopen`,  
`urlopen` returns a file like object.

Template for retrieving data from web pages:

```
from urllib import urlopen
url = ' < internet address > '
f = urlopen(url)
s = f.readline()
```

To get the weather forecast: [http://www.weather.gov/  
National Oceanic and Atmospheric Administration's  
National Weather Service](http://www.weather.gov/National%20Oceanic%20and%20Atmospheric%20Administration's%20National%20Weather%20Service)

The web site provides data for downloading:  
<http://weather.noaa.gov/pub/data/>

# Data from the Web

## Internet Basics

evolution of the web  
IP addresses and  
URLs  
client/server and  
HTTP

## Markup Languages

HTML, XML, MathML  
MathML generated  
by Maple

## Retrieving Data

the weather forecast

## CGI Programming

showing current date  
and time in browser

## Summary + Assignments

The module `urllib` exports `urlopen`,  
`urlopen` returns a file like object.

Template for retrieving data from web pages:

```
from urllib import urlopen
url = ' < internet address > '
f = urlopen(url)
s = f.readline()
```

To get the weather forecast: <http://www.weather.gov/>  
National Oceanic and Atmospheric Administration's  
National Weather Service

The web site provides data for downloading:  
<http://weather.noaa.gov/pub/data/>

# The Script forecast.py

## Internet Basics

evolution of the web  
IP addresses and  
URLs  
client/server and  
HTTP

## Markup Languages

HTML, XML, MathML  
MathML generated  
by Maple

## Retrieving Data

the weather forecast

## CGI Programming

showing current date  
and time in browser

## Summary + Assignments

```
from urllib import urlopen
host = 'http://weather.noaa.gov/'
fcst = '/pub/data/forecasts/state/'
url = host + fcst + '/il/ilz013.txt'
print 'opening ' + url + ' ...\n'
f = urlopen(url)

while True:
    s = f.readline()
    if s == '': break
    L = s.split(' ')
    if 'FCST' in L:
        s = f.readline()
        s = s + f.readline(); print s
    if 'CHICAGO' in L:
        s = s + f.readline()
        s = s + f.readline()
        s = s + f.readline(); print s
```

# The Script forecast.py

## Internet

### Basics

evolution of the web  
IP addresses and URLs  
client/server and HTTP

## Markup

### Languages

HTML, XML, MathML  
MathML generated by Maple

## Retrieving

### Data

the weather forecast

## CGI

### Programming

showing current date and time in browser

## Summary +

### Assignments

```
from urllib import urlopen
host = 'http://weather.noaa.gov/'
fcst = '/pub/data/forecasts/state/'
url = host + fcst + '/il/ilz013.txt'
print 'opening ' + url + ' ...\n'
f = urlopen(url)
while True:
    s = f.readline()
    if s == '': break
    L = s.split(' ')
    if 'FCST' in L:
        s = f.readline()
        s = s + f.readline(); print s
    if 'CHICAGO' in L:
        s = s + f.readline()
        s = s + f.readline()
        s = s + f.readline(); print s
```

# The Script forecast.py

## Internet

### Basics

evolution of the web  
IP addresses and URLs  
client/server and HTTP

## Markup

### Languages

HTML, XML, MathML  
MathML generated by Maple

## Retrieving

### Data

the weather forecast

## CGI

### Programming

showing current date and time in browser

## Summary +

### Assignments

```
from urllib import urlopen
host = 'http://weather.noaa.gov/'
fcst = '/pub/data/forecasts/state/'
url = host + fcst + '/il/ilz013.txt'
print 'opening ' + url + ' ...\n'
f = urlopen(url)
while True:
    s = f.readline()
    if s == '': break
    L = s.split(' ')
    if 'FCST' in L:
        s = f.readline()
        s = s + f.readline(); print s
    if 'CHICAGO' in L:
        s = s + f.readline()
        s = s + f.readline()
        s = s + f.readline(); print s
```

# The Script forecast.py

## Internet

### Basics

evolution of the web  
IP addresses and URLs  
client/server and HTTP

## Markup

### Languages

HTML, XML, MathML  
MathML generated by Maple

## Retrieving

### Data

the weather forecast

## CGI

### Programming

showing current date and time in browser

## Summary +

### Assignments

```
from urllib import urlopen
host = 'http://weather.noaa.gov/'
fcst = '/pub/data/forecasts/state/'
url = host + fcst + '/il/ilz013.txt'
print 'opening ' + url + ' ...\n'
f = urlopen(url)
while True:
    s = f.readline()
    if s == '': break
    L = s.split(' ')
    if 'FCST' in L:
        s = f.readline()
        s = s + f.readline(); print s
    if 'CHICAGO' in L:
        s = s + f.readline()
        s = s + f.readline()
        s = s + f.readline(); print s
```

# networking and the internet markup languages

## Internet Basics

evolution of the web  
IP addresses and  
URLs  
client/server and  
HTTP

## Markup Languages

HTML, XML, MathML  
MathML generated  
by Maple

## Retrieving Data

the weather forecast

## CGI Programming

showing current date  
and time in browser

## Summary + Assignments

- 1 Internet Basics  
evolution of the web  
IP addresses and URLs  
client/server and HTTP
- 2 Markup Languages  
HTML, XML, MathML  
MathML generated by Maple
- 3 Retrieving Data  
the weather forecast
- 4 CGI Programming  
showing current date and time in browser
- 5 Summary + Assignments

Internet  
Basics

evolution of the web  
IP addresses and  
URLs  
client/server and  
HTTP

Markup  
Languages

HTML, XML, MathML  
MathML generated  
by Maple

Retrieving  
Data

the weather forecast

CGI  
Programming

showing current date  
and time in browser

Summary +  
Assignments

# Python and CGI

## Common Gateway Interface

A sequel to GUIs: run programs through a browser.  
Web interfaces are also *event driven*.

Python as scripting language for the web:  
transform XML into XHTML pages.

The *Common Gateway Interface* (CGI) describes a set of protocols through which CGI programs interact with web servers and browsers.

CGI is common because it is not specific to any operating system or to any programming language or to any web server software.

We will use Python to generate dynamic web pages.

Internet  
Basics

evolution of the web  
IP addresses and  
URLs  
client/server and  
HTTP

Markup  
Languages

HTML, XML, MathML  
MathML generated  
by Maple

Retrieving  
Data

the weather forecast

CGI  
Programming

showing current date  
and time in browser

Summary +  
Assignments

# Python and CGI

## Common Gateway Interface

A sequel to GUIs: run programs through a browser.  
Web interfaces are also *event driven*.

Python as scripting language for the web:  
transform XML into XHTML pages.

The *Common Gateway Interface* (CGI) describes a set of protocols through which CGI programs interact with web servers and browsers.

CGI is common because it is not specific to any operating system or to any programming language or to any web server software.

We will use Python to generate dynamic web pages.

Internet  
Basics

evolution of the web  
IP addresses and  
URLs  
client/server and  
HTTP

Markup  
Languages

HTML, XML, MathML  
MathML generated  
by Maple

Retrieving  
Data

the weather forecast

CGI  
Programming

showing current date  
and time in browser

Summary +  
Assignments

# Python and CGI

## Common Gateway Interface

A sequel to GUIs: run programs through a browser.  
Web interfaces are also *event driven*.

Python as scripting language for the web:  
transform XML into XHTML pages.

The *Common Gateway Interface* (CGI) describes a set of protocols through which CGI programs interact with web servers and browsers.

CGI is common because it is not specific to any operating system or to any programming language or to any web server software.

We will use Python to generate dynamic web pages.

Internet  
Basics

evolution of the web  
IP addresses and  
URLs  
client/server and  
HTTP

Markup  
Languages

HTML, XML, MathML  
MathML generated  
by Maple

Retrieving  
Data

the weather forecast

CGI  
Programming

showing current date  
and time in browser

Summary +  
Assignments

# Python and CGI

## Common Gateway Interface

A sequel to GUIs: run programs through a browser.  
Web interfaces are also *event driven*.

Python as scripting language for the web:  
transform XML into XHTML pages.

The *Common Gateway Interface* (CGI) describes a set of protocols through which CGI programs interact with web servers and browsers.

CGI is common because it is not specific to any operating system or to any programming language or to any web server software.

We will use Python to generate dynamic web pages.

Internet  
Basics

evolution of the web  
IP addresses and  
URLs  
client/server and  
HTTP

Markup  
Languages

HTML, XML, MathML  
MathML generated  
by Maple

Retrieving  
Data

the weather forecast

CGI  
Programming

showing current date  
and time in browser

Summary +  
Assignments

# Python and CGI

## Common Gateway Interface

A sequel to GUIs: run programs through a browser.  
Web interfaces are also *event driven*.

Python as scripting language for the web:  
transform XML into XHTML pages.

The *Common Gateway Interface* (CGI) describes a set of protocols through which CGI programs interact with web servers and browsers.

CGI is common because it is not specific to any operating system or to any programming language or to any web server software.

We will use Python to generate dynamic web pages.

# Showing Current Time

## Internet Basics

evolution of the web  
IP addresses and  
URLs  
client/server and  
HTTP

## Markup Languages

HTML, XML, MathML  
MathML generated  
by Maple

## Retrieving Data

the weather forecast

## CGI Programming

showing current date  
and time in browser

## Summary + Assignments

### Three steps to run Python scripts on the web:

- 1 Open browser at `http://localhost`  
What should be visible is the message `If you can see this, it means that the installation of the Apache web server software on this system was successful.`
- 2 On MacOS X:  
`/Library/Webserver/CGI-Executables`  
contains Python scripts.
- 3 Write the Python script to show current time.  
From the module `time` we use the functions `time()` and `ctime()`.

# Showing Current Time

## Internet Basics

evolution of the web  
IP addresses and  
URLs  
client/server and  
HTTP

## Markup Languages

HTML, XML, MathML  
MathML generated  
by Maple

## Retrieving Data

the weather forecast

## CGI Programming

showing current date  
and time in browser

## Summary + Assignments

### Three steps to run Python scripts on the web:

- 1 Open browser at `http://localhost`  
What should be visible is the message `If you can see this, it means that the installation of the Apache web server software on this system was successful.`
- 2 On MacOS X:  
`/Library/Webserver/CGI-Executables`  
contains Python scripts.
- 3 Write the Python script to show current time.  
From the module `time` we use the functions `time()` and `ctime()`.

# Showing Current Time

## Internet Basics

evolution of the web  
IP addresses and  
URLs  
client/server and  
HTTP

## Markup Languages

HTML, XML, MathML  
MathML generated  
by Maple

## Retrieving Data

the weather forecast

## CGI Programming

showing current date  
and time in browser

## Summary + Assignments

Three steps to run Python scripts on the web:

- 1 Open browser at `http://localhost`  
What should be visible is the message `If you can see this, it means that the installation of the Apache web server software on this system was successful.`
- 2 On MacOS X:  
`/Library/Webserver/CGI-Executables`  
contains Python scripts.
- 3 Write the Python script to show current time.  
From the module `time` we use the functions `time()` and `ctime()`.

## showtime.py

Internet  
Basics

evolution of the web  
IP addresses and  
URLs  
client/server and  
HTTP

Markup  
Languages

HTML, XML, MathML  
MathML generated  
by Maple

Retrieving  
Data

the weather forecast

CGI  
Programming

showing current date  
and time in browser

Summary +  
Assignments

```
#!/Library/Frameworks/Python.framework/Versions/Current/bin/python
```

```
import time
print "Content-Type: text/plain\n"
print time.ctime(time.time())
```

Two points:

- 1 The first line is the location of the Python interpreter.
- 2 The first `print` indicates that plain text is written (opposed to HTML code).

## showtime.py

Internet  
Basics

evolution of the web  
IP addresses and  
URLs  
client/server and  
HTTP

Markup  
Languages

HTML, XML, MathML  
MathML generated  
by Maple

Retrieving  
Data

the weather forecast

CGI  
Programming

showing current date  
and time in browser

Summary +  
Assignments

```
#!/Library/Frameworks/Python.framework/Versions/Current/bin/python
```

```
import time  
print "Content-Type: text/plain\n"  
print time.ctime(time.time())
```

## Two points:

- 1 The first line is the location of the Python interpreter.
- 2 The first `print` indicates that plain text is written (opposed to HTML code).

## showtime.py

Internet  
Basics

evolution of the web  
IP addresses and  
URLs  
client/server and  
HTTP

Markup  
Languages

HTML, XML, MathML  
MathML generated  
by Maple

Retrieving  
Data

the weather forecast

CGI  
Programming

showing current date  
and time in browser

Summary +  
Assignments

```
#!/Library/Frameworks/Python.framework/Versions/Current/bin/python
```

```
import time  
print "Content-Type: text/plain\n"  
print time.ctime(time.time())
```

Two points:

- 1 The first line is the location of the Python interpreter.
- 2 The first `print` indicates that plain text is written (opposed to HTML code).

# Summary + Assignments

## Internet

### Basics

evolution of the web  
IP addresses and URLs  
client/server and HTTP

## Markup

### Languages

HTML, XML, MathML  
MathML generated by Maple

## Retrieving

### Data

the weather forecast

## CGI

### Programming

showing current date and time in browser

## Summary +

### Assignments

Background: §4.2,3 in *Computer Science, an overview*.  
More of chapter 5 of *Python Programming in Context*.

Assignments:

- 1 Make your own web page.  
Consult the ACCC help pages on web publishing.
- 2 Use Maple to generate MathML to display monomials, e.g.  $8x^3$ , and general polynomials.
- 3 Add a legend to the `forecast.py` script, using a dictionary to spell out the abbreviations PTCLDY, MOCLDY, etc.
- 4 Design a GUI dedicated to browsing weather forecasts. Which widgets will you use? What is the layout?
- 5 Write Python code for the GUI of the previous exercise.