Advanced CGI Scripts

Cookies

personalized web browsing using cookies: count number of visits

Password Encryption

secure hash algorithm

Login Forms

using cookies for login data
the script cookie_login.py
the script cookie_userpass.py

MCS 275 Lecture 32 Programming Tools and File Management Jan Verschelde, 4 April 2008

MCS 275 L-32

4 April 2008

Cookies

personalized web browsing using cookies: count number of visits

Password Encryption secure hash algorithm

Login Forms

Advanced CGI Scripts

Cookies personalized web browsing using cookies: count number of visits

Password Encryption secure hash algorithm

-ogin Forms
 using cookies for login data
 the script cookie_login.py
 the script cookie_userpass.py

MCS 275 L-32

4 April 2008

Cookies

personalized web browsing

sing cookies: count umber of visits

Password Encryption secure hash algorithm

Login Forms

using cookies for login data the script cookie_login.py the script cookie_userpass.py

・ロト・西ト・西ト・西・ うくの

more clever cgi scripts

Storing information about a client:

- 1. previous selections made
 - ightarrow passing data from one script to another
- 2. personal information \rightarrow identification and passwords
- 3. information from previous visits \rightarrow counting number of visits

Potential applications:

- 1. customize displayed content
- 2. store encrypted password for fast access
- 3. personalized pricing ...

MCS 275 L-32

4 April 2008

Cookies

personalized web browsing

using cookies: count number of visits

Password Encryption secure hash algorithm

Login Forms

more clever cgi scripts

Storing information about a client:

- 1. previous selections made
 - ightarrow passing data from one script to another
- 2. personal information
 - ightarrow identification and passwords
- 3. information from previous visits \rightarrow counting number of visits

Potential applications:

- 1. customize displayed content
- 2. store encrypted password for fast access
- 3. personalized pricing ...

MCS 275 L-32

4 April 2008

Cookies

personalized web browsing

sing cookies: count umber of visits

Password Encryption secure hash algorithm

Login Forms

using cookies for login data the script cookie_login.py the script cookie_userpass.py

more clever cgi scripts

Storing information about a client:

- 1. previous selections made
 - ightarrow passing data from one script to another
- 2. personal information
 - ightarrow identification and passwords
- 3. information from previous visits \rightarrow counting number of visits

Potential applications:

- 1. customize displayed content
- 2. store encrypted password for fast access
- 3. personalized pricing ...

MCS 275 L-32

4 April 2008

Cookies

personalized web browsing

using cookies: count umber of visits

Password Encryption secure hash algorithm

Login Forms

using cookies for login data the script cookie_login.py the script cookie_userpass.py

more clever cgi scripts

Storing information about a client:

- 1. previous selections made
 - ightarrow passing data from one script to another
- 2. personal information
 - ightarrow identification and passwords
- 3. information from previous visits
 - \rightarrow counting number of visits

Potential applications:

- 1. customize displayed content
- 2. store encrypted password for fast access
- 3. personalized pricing ...

MCS 275 L-32

4 April 2008

Cookies

personalized web browsing

sing cookies: count umber of visits

Password Encryption secure hash algorithm

Login Forms

using cookies for login data the script cookie_login.py the script cookie_userpass.py

more clever cgi scripts

Storing information about a client:

- 1. previous selections made
 - ightarrow passing data from one script to another
- 2. personal information
 - ightarrow identification and passwords
- 3. information from previous visits
 - \rightarrow counting number of visits

Potential applications:

- 1. customize displayed content
- 2. store encrypted password for fast access

3. personalized pricing ...

MCS 275 L-32

4 April 2008

Cookies

personalized web browsing

using cookies: count umber of visits

Password Encryption secure hash algorithm

Login Forms

using cookies for login data the script cookie_login.py the script cookie_userpass.py

▲□▶▲□▶▲□▶▲□▶ □ ● ● ●

more clever cgi scripts

Storing information about a client:

- 1. previous selections made
 - ightarrow passing data from one script to another
- 2. personal information
 - ightarrow identification and passwords
- 3. information from previous visits
 - \rightarrow counting number of visits

Potential applications:

- 1. customize displayed content
- 2. store encrypted password for fast access
- 3. personalized pricing ...

MCS 275 L-32

4 April 2008

Cookies

personalized web browsing

using cookies: count number of visits

Password Encryption secure hash algorithm

Login Forms

using cookies for login data the script cookie_login.py the script cookie_userpass.py

▲□▶▲□▶▲□▶▲□▶ □ ● ● ●

storing information about client

Cookies are data

stored by web server on client computer,

managed by the browser.

Using the Cookie module:

```
>>> import Cookie
>>> c = Cookie.Cookie()
>>> c['L'] = 32
>>> c['date'] = 'Fri 4 Apr 2008'
>>> print c
Set-Cookie: L=32
Set-Cookie: date="Fri 4 Apr 2008"
```

Cookies are objects like dictionaries. Reserved keys: expires and path.

MCS 275 L-32

4 April 2008

Cookies

personalized web browsing

sing cookies: count umber of visits

Password Encryption secure hash algorithm

Login Forms

```
・ロト・西ト・山田・山田・山下
```

storing information about client

Cookies are data

- stored by web server on client computer,
- managed by the browser.

Using the Cookie module:

```
>>> import Cookie
>>> c = Cookie.Cookie()
>>> c['L'] = 32
>>> c['date'] = 'Fri 4 Apr 2008'
>>> print c
Set-Cookie: L=32
Set-Cookie: date="Fri 4 Apr 2008"
```

Cookies are objects like dictionaries. Reserved keys: expires and path.

MCS 275 L-32

4 April 2008

Cookies

personalized web browsing

using cookies: count number of visits

Password Encryption secure hash algorithm

Login Forms

using cookies for login data the script cookie_login.py the script cookie_userpass.py

< □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □

storing information about client

Cookies are data

- stored by web server on client computer,
- managed by the browser.

Using the Cookie module:

```
>>> import Cookie
>>> c = Cookie.Cookie()
>>> c['L'] = 32
>>> c['date'] = 'Fri 4 Apr 2008'
>>> print c
Set-Cookie: L=32
Set-Cookie: date="Fri 4 Apr 2008"
```

Cookies are objects like dictionaries. Reserved keys: expires and path.

MCS 275 L-32

4 April 2008

Cookies

personalized web browsing

sing cookies: count umber of visits

Password Encryption secure hash algorithm

Login Forms

```
< □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □
```

storing information about client

Cookies are data

- stored by web server on client computer,
- managed by the browser.

Using the Cookie module:

```
>>> import Cookie
>>> c = Cookie.Cookie()
>>> c['L'] = 32
>>> c['date'] = 'Fri 4 Apr 2008'
>>> print c
Set-Cookie: L=32
Set-Cookie: date="Fri 4 Apr 2008"
```

Cookies are objects like dictionaries. Reserved keys: expires and path.

MCS 275 L-32

4 April 2008

Cookies

personalized web browsing

using cookies: count number of visits

Password Encryption secure hash algorithm

Login Forms

using cookies for login data the script cookie_login.py the script cookie_userpass.py

< □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □

storing information about client

Cookies are data

- stored by web server on client computer,
- managed by the browser.

Using the Cookie module:

```
>>> import Cookie
>>> c = Cookie.Cookie()
>>> c['L'] = 32
>>> c['date'] = 'Fri 4 Apr 2008'
>>> print c
Set-Cookie: L=32
Set-Cookie: date="Fri 4 Apr 2008"
```

Cookies are objects like dictionaries. Reserved keys: expires and path.

MCS 275 L-32

4 April 2008

Cookies

personalized web browsing

using cookies: count number of visits

Password Encryption secure hash algorithm

Login Forms

Advanced CGI Scripts

Cookies personalized web browsing using cookies: count number of visits

Password Encryption secure hash algorithm

Login Forms using cookies for login data the script cookie_login.py the script cookie_userpass.py

MCS 275 L-32

4 April 2008

Cookies

personalized web browsing

using cookies: count number of visits

Password Encryption secure hash algorithm

Login Forms

using cookies for login data the script cookie_login.py the script cookie_userpass.py

・ロト・西ト・西ト・日・ うらの

Count number of times browser visited.

Write a script to

- 1. retrieve cookie, initialize counter to zero,
- 2. or increment the value of counter by one,
- 3. and display counter value on the page.

Script is cookie_counter.py.

- 1. in /Library/WebServer/CGI-Executables,
- 2. browser settings must accept cookies.

Note: each browser has its own cookies.

MCS 275 L-32

4 April 2008

Cookies

personalized web browsing

using cookies: count number of visits

Password Encryption secure hash algorithm

Login Forms

using cookies for login data the script cookie_login.py the script cookie_userpass.py

Count number of times browser visited.

Write a script to

- 1. retrieve cookie, initialize counter to zero,
- 2. or increment the value of counter by one,
- 3. and display counter value on the page.

Script is cookie_counter.py.

- in /Library/WebServer/CGI-Executables,
- 2. browser settings must accept cookies.

Note: each browser has its own cookies.

MCS 275 L-32

4 April 2008

Cookies

personalized web browsing

using cookies: count number of visits

Password Encryption secure hash algorithm

Login Forms

using cookies for login data the script cookie_login.py the script cookie_userpass.py

▲□▶▲□▶▲□▶▲□▶ □ ● ● ●

Count number of times browser visited.

Write a script to

- 1. retrieve cookie, initialize counter to zero,
- 2. or increment the value of counter by one,
- 3. and display counter value on the page.

Script is cookie_counter.py.

- 1. in /Library/WebServer/CGI-Executables,
- 2. browser settings must accept cookies.

Note: each browser has its own cookies.

MCS 275 L-32

4 April 2008

Cookies

personalized web browsing

using cookies: count number of visits

Password Encryption secure hash algorithm

Login Forms

using cookies for login data the script cookie_login.py the script cookie_userpass.py

▲□▶▲□▶▲□▶▲□▶ □ ● ● ●

Count number of times browser visited.

Write a script to

- 1. retrieve cookie, initialize counter to zero,
- 2. or increment the value of counter by one,
- 3. and display counter value on the page.

Script is cookie_counter.py.

- 1. in /Library/WebServer/CGI-Executables,
- 2. browser settings must accept cookies.

Note: each browser has its own cookies.

MCS 275 L-32

4 April 2008

Cookies

personalized web browsing

using cookies: count number of visits

Password Encryption secure hash algorithm

Login Forms

Count number of times browser visited.

Write a script to

- 1. retrieve cookie, initialize counter to zero,
- 2. or increment the value of counter by one,
- 3. and display counter value on the page.

Script is cookie_counter.py.

- 1. in /Library/WebServer/CGI-Executables,
- 2. browser settings must accept cookies.

Note: each browser has its own cookies.

MCS 275 L-32

4 April 2008

Cookies

personalized web browsing

using cookies: count number of visits

Password Encryption secure hash algorithm

Login Forms

Count number of times browser visited.

Write a script to

- 1. retrieve cookie, initialize counter to zero,
- 2. or increment the value of counter by one,
- 3. and display counter value on the page.

Script is cookie_counter.py.

- 1. in /Library/WebServer/CGI-Executables,
- 2. browser settings must accept cookies.

Note: each browser has its own cookies.

MCS 275 L-32

4 April 2008

Cookies

personalized web browsing

using cookies: count number of visits

Password Encryption secure hash algorithm

Login Forms

Count number of times browser visited.

Write a script to

- 1. retrieve cookie, initialize counter to zero,
- 2. or increment the value of counter by one,
- 3. and display counter value on the page.

Script is cookie_counter.py.

- 1. in /Library/WebServer/CGI-Executables,
- 2. browser settings must accept cookies.

Note: each browser has its own cookies.

MCS 275 L-32

4 April 2008

Cookies

personalized web browsing

using cookies: count number of visits

Password Encryption secure hash algorithm

Login Forms

Running cookie_counter.py

000				Mo	zilla Fire	fox
🛶 - 🛶 - 🙋 🕓 🟠 🗑 ht	tp://localhost/cg	gi-bin/cookie_co	ounter.py			
Getting Started Latest Headlines	Apple Yahoo!	Google Maps	YouTube	Wikipedia	News v	Pop
counter: 5						
000	Privacy					C
Main Tabs Content Feeds Privac	y Security Adv	anced				1
	000		Cookies			
History						
de l'anne e	Search:				C	Clea
Remember visited pages fc	The following	cookies are st	ored on vo	ur compute	ar	
Remember what I enter in I	Site			Cookie Nan		
Remember what I've down	v localhost			COOKIE Nan	ne	
10000	localhost counter					
Cookies	► math.uic.e	du				
Accept cookies from sites						
Keep until: they expire	Name: cou	nter				
1222 22	Content: "15\	012."				
Private Data	Host: loca	alhost				
Always clear my private da	Path: /cgi-bin/					
Always clear my private da	Send For: Any type of connection					
Ask me before clearing pri	Expires: at end of session					
	Remove Co	okie) (Re	move All C	ookies)		
	a provinsi na second					

MCS 275 L-32

4 April 2008

Cookies

personalized web browsing

using cookies: count number of visits

Password Encryption secure hash algorithm

Login Forms

using cookies for login data the script cookie_login.py the script cookie_userpass.py

◆□▶ ◆□▶ ◆三▶ ◆三▶ ● のへで

#!/Library/Frameworks/.../bin/python import Cookie, os

```
▲ロ ▶ ▲周 ▶ ▲ 国 ▶ ▲ 国 ▶ ● ● ● ● ●
```

MCS 275 L-32

4 April 2008

using cookies: count number of visits

```
#!/Library/Frameworks/.../bin/python
import Cookie, os
def GetCookie():
   .....
   Retrieves cookie, either initializes counter, Forms
   or increments the counter by one.
   .....
```

▲ロ ▶ ▲周 ▶ ▲ 国 ▶ ▲ 国 ▶ ● ● ● ● ●

MCS 275 L-32

4 April 2008

using cookies: count number of visits

```
#!/Library/Frameworks/.../bin/python
import Cookie, os
                                                        using cookies: count
                                                        number of visits
def GetCookie():
    .....
   Retrieves cookie, either initializes counter, Forms
   or increments the counter by one.
    .....
def main():
    .....
   Retrieves a cookie and writes
   the value of counter to the page.
    .....
   c = GetCookie()
   print c
                               ▲ロ ▶ ▲周 ▶ ▲ 国 ▶ ▲ 国 ▶ ● ● ● ● ●
```

MCS 275 L-32

```
#!/Library/Frameworks/.../bin/python
import Cookie, os
                                                    using cookies: count
                                                    number of visits
def GetCookie():
   .....
   Retrieves cookie, either initializes counter, Forms
   or increments the counter by one.
   .....
def main():
   .....
   Retrieves a cookie and writes
   the value of counter to the page.
   .....
   c = GetCookie()
   print c
   print "Content-Type: text/plain\n"
   print "counter: %d" % c['counter'].value
                             ▲□▶ ▲□▶ ▲□▶ ▲□▶ □ のQ@
```

MCS 275 L-32

```
def GetCookie():
                                                     using cookies: count
                                                     number of visits
    .....
   Retrieves cookie, either initializes counter,
   or increments the counter by one.
    .....
   if os.environ.has key('HTTP COOKIE'):
       c = Cookie.Cookie(os.environ['HTTP COOKIE'])
```

▲□▶ ▲□▶ ▲□▶ ▲□▶ □ のQ@

MCS 275 L-32

```
def GetCookie():
                                                     using cookies: count
                                                     number of visits
    .....
   Retrieves cookie, either initializes counter,
   or increments the counter by one.
    .....
   if os.environ.has key('HTTP COOKIE'):
       c = Cookie.Cookie(os.environ['HTTP COOKIE'])
   else:
       c = Cookie.Cookie()
```

▲ロ ▶ ▲周 ▶ ▲ 国 ▶ ▲ 国 ▶ ● ● ● ● ●

MCS 275 L-32

```
def GetCookie():
                                                    using cookies: count
                                                    number of visits
   .....
   Retrieves cookie, either initializes counter,
   or increments the counter by one.
   .....
   if os.environ.has key('HTTP COOKIE'):
       c = Cookie.Cookie(os.environ['HTTP COOKIE'])
   else:
       c = Cookie.Cookie()
   if not c.has_key('counter'):
       c['counter'] = 0
```

▲ロ ▶ ▲周 ▶ ▲ 国 ▶ ▲ 国 ▶ ● ● ● ● ●

MCS 275 L-32

```
def GetCookie():
                                                    using cookies: count
                                                    number of visits
   11 11 11
   Retrieves cookie, either initializes counter,
   or increments the counter by one.
   .....
   if os.environ.has key('HTTP COOKIE'):
       c = Cookie.Cookie(os.environ['HTTP COOKIE'))
   else:
       c = Cookie.Cookie()
   if not c.has_key('counter'):
       c['counter'] = 0
   else:
       c['counter'] = c['counter'].value + 1
   return c
```

◆□▶ ◆□▶ ◆臣▶ ◆臣▶ □臣 = のへで

MCS 275 L-32

Advanced CGI Scripts

Cookies

personalized web browsing using cookies: count number of visits

Password Encryption secure hash algorithm

Login Forms

using cookies for login data
the script cookie_login.py
the script cookie_userpass.py

MCS 275 L-32

4 April 2008

Cookies

personalized web browsing using cookies: count number of visits

Password Encryption

secure hash algorithm

Login Forms

using cookies for login data the script cookie_login.py the script cookie_userpass.py

・ロト・日本・日本・日本・日本・日本

for password encryption

We can use cookies for login names and passwords.

If passwords are unencrypted, then insecure.

Secure hash algorithm:

- 1. computationally infeasible to compute inverse \rightarrow is *trapdoor* function
- 2. very low collision rate \rightarrow very low chance that two different messages will generate the same key

Server encrypts password before sending to client.

Authentication by comparing encrypted passwords.

MCS 275 L-32

4 April 2008

Cookies

personalized web browsing using cookies: count number of visits

Password Encryption

secure hash algorithm

Login Forms

for password encryption

We can use cookies for login names and passwords. If passwords are unencrypted, then insecure.

Secure hash algorithm:

- 1. computationally infeasible to compute inverse \rightarrow is *trapdoor* function
- 2. very low collision rate

ightarrow very low chance that two different messages will generate the same key

Server encrypts password before sending to client.

Authentication by comparing encrypted passwords.

MCS 275 L-32

4 April 2008

Cookies

personalized web browsing using cookies: count number of visits

Password Encryption

secure hash algorithm

Login Forms

for password encryption

We can use cookies for login names and passwords.

If passwords are unencrypted, then insecure.

Secure hash algorithm:

- 1. computationally infeasible to compute inverse
 - \rightarrow is *trapdoor* function
- 2. very low collision rate

 \rightarrow very low chance that two different messages will generate the same key

Server encrypts password before sending to client. Authentication by comparing encrypted passwords.

MCS 275 L-32

4 April 2008

Cookies

personalized web browsing using cookies: count number of visits

Password Encryption

secure hash algorithm

Login Forms

for password encryption

We can use cookies for login names and passwords.

If passwords are unencrypted, then insecure.

Secure hash algorithm:

- 1. computationally infeasible to compute inverse
 - \rightarrow is *trapdoor* function
- 2. very low collision rate

 \rightarrow very low chance that two different messages will generate the same key

▲ロ ▶ ▲周 ▶ ▲ 国 ▶ ▲ 国 ▶ ● ● ● ● ●

Server encrypts password before sending to client.

Authentication by comparing encrypted passwords.

MCS 275 L-32

4 April 2008

Cookies

personalized web browsing using cookies: count number of visits

Password Encryption

secure hash algorithm

Login Forms

for password encryption

We can use cookies for login names and passwords.

If passwords are unencrypted, then insecure.

Secure hash algorithm:

- 1. computationally infeasible to compute inverse
 - \rightarrow is *trapdoor* function
- 2. very low collision rate

 \rightarrow very low chance that two different messages will generate the same key

Server encrypts password before sending to client.

Authentication by comparing encrypted passwords.

MCS 275 L-32

4 April 2008

Cookies

personalized web browsing using cookies: count number of visits

Password Encryption

secure hash algorithm

Login Forms
Using the sha Module

>>> import sha >>> m = 'this is me'

>>> h = sha.new()

>>> h

<shal HASH object @ 0x99ff80>

>>> h.hexdigest() 'da39a3ee5e6b4b0d3255bfef95601890afd80709'

MCS 275 L-32

4 April 2008

Cookies

personalized web browsing using cookies: count number of visits

Password Encryption

secure hash algorithm

Login Forms

using cookies for login data the script cookie_login.py the script cookie_userpass.py

▲□▶▲□▶▲□▶▲□▶ □ ● ●

Using the sha Module

>>> import sha

- >>> m = 'this is me'
- >>> h = sha.new()

>>> h

<shal HASH object @ 0x99ff80>

>>> h.hexdigest() 'da39a3ee5e6b4b0d3255bfef95601890afd80709'

MCS 275 L-32

4 April 2008

Cookies

personalized web browsing using cookies: count number of visits

Password Encryption

secure hash algorithm

Login Forms

using cookies for login data the script cookie_login.py the script cookie_userpass.py

・ロト・西ト・ヨト・ヨー もくの

Using the sha Module

```
>>> import sha
>>> m = 'this is me'
>>> h = sha.new()
>>> h
<shal HASH object @ 0x99ff80>
>>> h.hexdigest()
```

'da39a3ee5e6b4b0d3255bfef95601890afd80709'

MCS 275 L-32

4 April 2008

Cookies

personalized web browsing using cookies: count number of visits

Password Encryption

secure hash algorithm

Login Forms

using cookies for login data the script cookie_login.py the script cookie_userpass.py

▲□▶▲□▶▲□▶▲□▶ □ ● ● ●

Advanced CGI Scripts

Cookies

personalized web browsing using cookies: count number of visits

Password Encryption secure hash algorithm

Login Forms using cookies for login data

t<mark>he script</mark> cookie_login.py t<mark>he script</mark> cookie_userpass.py

MCS 275 L-32

4 April 2008

Cookies

personalized web browsing using cookies: count number of visits

Password Encryption secure hash algorithm

Login Forms

using cookies for login data

the script cookie_login.py the script cookie_userpass.py

▲□▶▲□▶▲□▶▲□▶ □ ● ●

with cookies to remember data

Accessing cookie_login.py for the first time:

1. user submits login name and password,

- submitted data processed by cookie_userpass.py,
- 3. cookie stores login name and encrypted password.

Connecting to cookie_login.py a second time:

- 1. cookie is retrieved,
- 2. login name is displayed if not empty,
- 3. user must type no password if in cookie.

MCS 275 L-32

4 April 2008

Cookies

personalized web browsing using cookies: count number of visits

Password Encryption secure hash algorithm

Login Forms

using cookies for login data

the script cookie_login.py the script cookie_userpass.py

with cookies to remember data

Accessing cookie_login.py for the first time:

- 1. user submits login name and password,
- 2. submitted data processed by cookie_userpass.py,
- 3. cookie stores login name and encrypted password.

▲ロ ▶ ▲周 ▶ ▲ 国 ▶ ▲ 国 ▶ ● ● ● ● ●

Connecting to cookie_login.py a second time:

- 1. cookie is retrieved,
- 2. login name is displayed if not empty,
- 3. user must type no password if in cookie.

MCS 275 L-32

4 April 2008

Cookies

personalized web browsing using cookies: count number of visits

Password Encryption secure hash algorithm

Login Forms

using cookies for login data

the script cookie_login.py the script cookie_userpass.py

with cookies to remember data

Accessing cookie_login.py for the first time:

- 1. user submits login name and password,
- submitted data processed by cookie_userpass.py,
- 3. cookie stores login name and encrypted password.

▲ロ ▶ ▲周 ▶ ▲ 国 ▶ ▲ 国 ▶ ● ● ● ● ●

Connecting to cookie_login.py **a second time**:

- 1. cookie is retrieved,
- 2. login name is displayed if not empty,
- 3. user must type no password if in cookie.

MCS 275 L-32

4 April 2008

Cookies

personalized web browsing using cookies: count number of visits

Password Encryption secure hash algorithm

Login Forms

using cookies for login data

the script cookie_login.py the script cookie_userpass.py

with cookies to remember data

Accessing cookie_login.py for the first time:

- 1. user submits login name and password,
- submitted data processed by cookie_userpass.py,
- 3. cookie stores login name and encrypted password.

▲ロ ▶ ▲周 ▶ ▲ 国 ▶ ▲ 国 ▶ ● ● ● ● ●

Connecting to cookie_login.py a second time:

- 1. cookie is retrieved,
- 2. login name is displayed if not empty,
- 3. user must type no password if in cookie.

MCS 275 L-32

4 April 2008

Cookies

personalized web browsing using cookies: count number of visits

Password Encryption secure hash algorithm

Login Forms

using cookies for login data

ne script cookie_login.py the script cookie_userpass.py

with cookies to remember data

Accessing cookie_login.py for the first time:

- 1. user submits login name and password,
- submitted data processed by cookie_userpass.py,
- 3. cookie stores login name and encrypted password.

▲ロ ▶ ▲周 ▶ ▲ 国 ▶ ▲ 国 ▶ ● ● ● ● ●

Connecting to cookie_login.py a second time:

- 1. cookie is retrieved,
- 2. login name is displayed if not empty,

3. user must type no password if in cookie.

MCS 275 L-32

4 April 2008

Cookies

personalized web browsing using cookies: count number of visits

Password Encryption secure hash algorithm

Login Forms

using cookies for login data

ne script cookie_login.py the script cookie_userpass.py

with cookies to remember data

Accessing cookie_login.py for the first time:

- 1. user submits login name and password,
- submitted data processed by cookie_userpass.py,
- 3. cookie stores login name and encrypted password.

Connecting to cookie_login.py a second time:

- 1. cookie is retrieved,
- 2. login name is displayed if not empty,
- 3. user must type no password if in cookie.

MCS 275 L-32

4 April 2008

Cookies

personalized web browsing using cookies: count number of visits

Password Encryption secure hash algorithm

Login Forms

using cookies for login data

ne script cookie_login.py the script cookie_userpass.py

Advanced CGI Scripts

Cookies

personalized web browsing using cookies: count number of visits

Password Encryption secure hash algorithm

Login Forms using cookies for login data the script cookie_login.py the script cookie_userpass.p

MCS 275 L-32

4 April 2008

Cookies

personalized web browsing using cookies: count number of visits

Password Encryption secure hash algorithm

_ogin Forms

sing cookies for login data

the script cookie_login.py

the script cookie_userpass.py

・ロト・西ト・西ト・西・ うくの

Script cookie_login.py

```
#!/Library/Frameworks/../bin/python
import cqi, Cookie, os
```

MCS 275 L-32

4 April 2008

Cookies

ersonalized web browsing Ising cookies: count number of visits

Password Encryption secure hash algorithm

_ogin Forms

sing cookies for login data

```
the script cookie_login.py
```

```
・ロト・日本・日本・日本・日本・日本
```

Script cookie_login.py

```
#!/Library/Frameworks/../bin/python
import cqi, Cookie, os
def main():
   ......
   Form to process login.
   .....
   c = GetCookie()
   print c
   print "Content-type: text/html\n"
   print "<html><body>\n"
   AskName(c)
   print "</body></html>\n"
```

MCS 275 L-32

4 April 2008

Cookies

ersonalized web browsing Ising cookies: count number of visits

```
Password
Encryption
secure hash algorithm
```

_ogin Forms

ising cookies for login data

```
the script cookie_login.py
```

```
the script
cookie_userpass.py
```

◆□ > ◆□ > ◆豆 > ◆豆 > ̄豆 → ���

```
def GetCookie():
```

. . .

```
Retrieves cookie, and initializes it.
......
```

if os.environ.has_key('HTTP_COOKIE'): c = Cookie.Cookie(os.environ['HTTP_COOKIE') else:

```
c = Cookie.Cookie()
```

MCS 275 L-32

4 April 2008

the script cookie login.pv

```
▲ロ ▶ ▲周 ▶ ▲ 国 ▶ ▲ 国 ▶ ● ● ● ● ●
```

```
def GetCookie():
   . . .
   Retrieves cookie, and initializes it.
   ......
                                                   the script
   if os.environ.has_key('HTTP_COOKIE'):
                                                   cookie login.pv
       c = Cookie.Cookie(os.environ['HTTP_COOKIE')
   else:
       c = Cookie.Cookie()
   if not c.has_key('login'):
       c['login'] = ''
       c['passw'] = ''
   return c
```

◆ロ → ◆ 団 → ◆ 豆 → ◆ 国 → ◆ 句 へ ④

MCS 275 L-32

4 April 2008

The Function AskName() first part

```
def AskName(c):
    .....
   Form to enter user name, using cookie c
   to show user name.
                                                       the script
    .....
   print """<form method = "post"</pre>
                     action = "cookie userpass.py">
    ......
```

MCS 275 L-32

4 April 2008

```
cookie login.pv
```

▲ロ ▶ ▲周 ▶ ▲ 国 ▶ ▲ 国 ▶ ● ● ● ● ●

The Function AskName() first part

```
def AskName(c):
   .....
   Form to enter user name, using cookie c
   to show user name.
                                                     the script
   .....
   print """<form method = "post"</pre>
                     action = "cookie userpass.py">
   ......
   v = c['login'].value
   w = c['passw'].value
```

 \rightarrow values of v and w determine what will be asked from the user.

▲ロ ▶ ▲周 ▶ ▲ 国 ▶ ▲ 国 ▶ ● ● ● ● ●

MCS 275 L-32

4 April 2008

```
cookie login.pv
```

```
if v == '':
  print """
  Login Name:
   <input type = "text" name = "login"
          size = 20>"""
```

▲□▶ ▲□▶ ▲□▶ ▲□▶ □ のQ@

MCS 275 L-32

4 April 2008

Cookies

personalized web browsing using cookies: count number of visits

Password Encryption secure hash algorithm

Login Forms

sing cookies for login data

```
the script cookie_login.py
```

```
if v == '':
  print """
  Login Name:
   <input type = "text" name = "login"
         size = 20>"""
else:
  print """
  Login Name:
   <input type = "text" name = "login"
         size = 20 value = s^{-1}
```

MCS 275 L-32

4 April 2008

Cookies

personalized web browsing using cookies: count number of visits

Password Encryption secure hash algorithm

Login Forms

sing cookies for login data

```
the script cookie_login.py
```

the script cookie_userpass.py

◆□▶ ◆□▶ ◆三▶ ◆三▶ ● □ ● ●

```
if v == '':
   print """
  Login Name:
   <input type = "text" name = "login"
          size = 20>"""
else:
  print """
   Login Name:
   <input type = "text" name = "login"
          size = 20 value = s^{-1}
if w == '':
   print """ Password:
   <input type = "password" name = "passw"
          size = 20>"""
```

▲□▶ ▲□▶ ▲□▶ ▲□▶ □ のQ@

MCS 275 L-32

4 April 2008

Cookies

personalized web browsing using cookies: count number of visits

```
Password
Encryption
secure hash algorithm
```

Login Forms

ising cookies for login data

```
the script cookie_login.py
```

```
if v == '':
   print """
  Login Name:
   <input type = "text" name = "login"
          size = 20>"""
else:
  print """
   Login Name:
   <input type = "text" name = "login"
          size = 20 value = s^{-1}
if w == '':
   print """ Password:
   <input type = "password" name = "passw"
          size = 20>"""
print """
   <input type = "submit" value = "submit">
   </form>"""
```

MCS 275 L-32

4 April 2008

Cookies

personalized web browsing using cookies: count number of visits

Password Encryption secure hash algorithm

_ogin Forms

ising cookies for login data

```
the script cookie_login.py
```

the script cookie_userpass.py

◆□▶ ◆□▶ ◆ □ ▶ ◆ □ ▶ ● □ ● ● ● ●

Advanced CGI Scripts

Cookies

personalized web browsing using cookies: count number of visits

Password Encryption secure hash algorithm

Login Forms using cookies for login data the script cookie_login.py the script cookie_userpass.py

MCS 275 L-32

4 April 2008

Cookies

ersonalized web browsing using cookies: count number of visits

Password Encryption secure hash algorithm

Login Forms

using cookies for login data the script cookie_login.py

the script cookie_userpass.py

▲□▶▲□▶▲□▶▲□▶ □ ● ●

Script cookie_userpass.py

```
#!/Library/Frameworks/../bin/python
import cgi, Cookie, os, sha
def main():
   ......
   Form to process login.
   .....
   form = cqi.FieldStorage()
   c = GetCookie(form)
   print c
```

MCS 275 L-32

4 April 2008

Cookies

personalized web browsing using cookies: count number of visits

Password Encryption secure hash algorithm

Login Forms

using cookies for login data the script cookie_login.py

```
< □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □
```

Script cookie_userpass.py

```
#!/Library/Frameworks/../bin/python
import cgi, Cookie, os, sha
def main():
   ......
   Form to process login.
   .....
   form = cqi.FieldStorage()
   c = GetCookie(form)
   print c
   print "Content-type: text/html\n"
   print "<html><body>\n"
   error = ProcessName(form)
```

MCS 275 L-32

4 April 2008

Cookies

personalized web browsing using cookies: count number of visits

Password Encryption secure hash algorithm

Login Forms

using cookies for login data the script cookie_login.py

```
< □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □
```

Script cookie_userpass.py

```
#!/Library/Frameworks/../bin/python
import cqi, Cookie, os, sha
def main():
   11 11 11
   Form to process login.
   .....
   form = cqi.FieldStorage()
   c = GetCookie(form)
   print c
   print "Content-type: text/html\n"
   print "<html><body>\n"
   error = ProcessName(form)
   if not error:
      ProcessPass(c)
   print "</body></html>\n"
```

MCS 275 L-32

4 April 2008

Cookies

personalized web browsing using cookies: count number of visits

Password Encryption secure hash algorithm

Login Forms

using cookies for login data the script cookie_login.py

```
< □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □
```

```
def GetCookie(form):
   .....
   Retrieves cookie and uses form to update.
```

......

```
if os.environ.has_key('HTTP_COOKIE'):
```

c = Cookie.Cookie(os.environ['HTTP COOKIECdade) +) Clagin data else:

```
c = Cookie.Cookie()
```

▲ロ ▶ ▲周 ▶ ▲ 国 ▶ ▲ 国 ▶ ● ● ● ● ●

MCS 275 L-32

4 April 2008

```
def GetCookie(form):
    .....
   Retrieves cookie and uses form to update.
    ......
   if os.environ.has_key('HTTP_COOKIE'):
       c = Cookie.Cookie(os.environ['HTTP COOKIECdade) +) Clagin data
   else:
                                                      the script
       c = Cookie.Cookie()
                                                      cookie_userpass.py
   if c.has key('login'):
       if form.has_key('login'):
          c['login'] = form['login'].value
```

◆□▶ ◆□▶ ◆三▶ ◆三▶ ・三 ・ の々で

MCS 275 L-32

4 April 2008

```
def GetCookie(form):
   .....
   Retrieves cookie and uses form to update.
   .....
   if os.environ.has_key('HTTP_COOKIE'):
      c = Cookie.Cookie(os.environ['HTTP COOKIE doch))
   else:
                                                  the script
      c = Cookie.Cookie()
                                                  cookie_userpass.py
   if c.has key('login'):
      if form.has_key('login'):
          c['login'] = form['login'].value
   if c.has key('passw'):
      if form.has key('passw'):
          p = form['passw'].value
          d = sha.new(p).hexdigest()
          c['passw'] = d
   return c
```

MCS 275 L-32

4 April 2008

Processing the Name

```
def ProcessName(form):
   .....
   Processes name of login form.
   Returns True if error, else False.
   .....
   error = False
   try:
      n = form['login'].value
   except KeyError:
      print "please enter your name"
      error = True
   if not error:
      print 'welcome ' + n + '\n'
   return error
```

MCS 275 L-32

4 April 2008

Cookies

personalized web browsing using cookies: count number of visits

Password Encryption secure hash algorithm

Login Forms

using cookies for login data the script cookie_login.py

```
< □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □
```

Processing Password

```
def ProcessPass(c):
    """
    Processes password of login form.
    """
    print "your password is "
    print c['passw'].value
```

 \rightarrow instead of printing the password, compare against password on file.

MCS 275 L-32

4 April 2008

Cookies

personalized web browsing using cookies: count number of visits

Password Encryption secure hash algorithm

Login Forms

▲ロ ▶ ▲周 ▶ ▲ 国 ▶ ▲ 国 ▶ ● ● ● ● ●

using cookies for login data the script cookie_login.py

Summary + Assignments

We covered more of chapter 14 in *Making Use of Python*. Assignments:

- Extend cookie_counter.py so that a different HTML page is displayed based on whether the counter is zero or not.
- 2. Extend cookie_userpass.py for it to do proper password authentication: compare the given password with the one that was earlier entered and stored on file.

MCS 275 L-32

4 April 2008

Cookies

personalized web browsing using cookies: count number of visits

Password Encryption secure hash algorithm

Login Forms

using cookies for login data the script cookie_login.py