Web Applications

Available Information and Input

HTML Forms

Web Applications using PHP

Manipulating the PHP Configuration

Server Variables

Information Available to PHP Scripts

- Information on the PHP environment
- Information on the web server and client request
- Form data
- Cookie/Session data
- Miscellaneous
  - string date(format)
    returns the current date/time presented according to format
    for example, `date('H:i, l, F Y')`
    which results in `12:20 Thursday, 8 March 2012`
  - int time()
    returns the current time measured in the number of seconds
    since January 1 1970 00:00:00 GMT

Available Information and Input

PHP Environment

- `phpinfo()` displays information about the PHP installation and
  EGPCS data (Environment, GET, POST, Cookie, and Server data) for
  the current client request
- `phpinfo(part)` displays selected information

```
<html lang="en-GB"><head></head><body>
<?php
  phpinfo(); // Show all information
  phpinfo(INFO_VARIABLES); // Show only info on EGPCS data
</body></html>
```

```
http://cgi.csc.liv.ac.uk/~ullrich/COMP284/examples/phpinfo.php
```

INFO_GENERAL
The configuration, php.ini location, build date, web server,
INFO_CONFIGURATION
Local and master values for PHP directives
INFO_MODULES
Loaded modules
INFO_VARIABLES
All EGPCS data

```
// Show only info on EGPCS data
phpinfo(INFO_VARIABLES);
```

The following functions can be used to access and change the
configuration of PHP from within a PHP script:

- `array ini_get_all()`
  returns all the registered configuration options
- `string ini_get(option)`
  returns the value of the configuration option on success
- `string ini_set(option, value)`
  sets the value of the given configuration option to a new value
  the configuration option will keep this new value during the script’s
  execution and will be restored afterwards
- `void ini_restore(option)`
  restores a given configuration option to its original value

When considering Python CGI programming we have used HTML forms
that generated a client request that was handled by a
Python CGI program:

```
<form action="http://student.csc.liv.ac.uk/cgi-bin/cgivwrap/uh/demo" method="post">
  ...<br>
</form>
```

Now we will use a PHP script instead:

```
<form action="http://student.csc.liv.ac.uk/~uh/demo.php" method="post">
  ...<br>
</form>
```

- The PHP script file name must be stored in a directory accessible by the web
  server, for example `~/HOME/public_html`, and be readable by the web
  server
- The PHP script file name must have the extension .php, e.g. `demo.php`
**Forms in PHP: Example (1)**

```php
<?php
// extract_names.php

function extract_names($url) {
    $text = file_get_contents($url);
    if ($text === false)
        return "ERROR: INVALID URL!"
    else {
        $correct = preg_match_all("/(name=(.*))\)/", $text, $matches, PREG_PATTERN_ORDER);
        if ($correct == 0) return "ERROR: NO NAMES FOUND"
        $count = array_count_values($matches[1]);
        arsort($count);
        foreach ($count as $name => $number) {
            $table .= "<tr><td>$name</td><td>$number</td></tr>"
        }
        $table = "<table><thead><tr><th>Name</th><th>No of occur</th></tr></thead><tbody>".$table."</tbody></table>
        return $table;
    }
}

extract_names("http://cgi.csc.liv.ac.uk/~ullrich/COMP519/examples/extract_names.php");
?>
```

**Revision and Further Reading**

Read
- Chapter 11: Form Handling

---

**Available Information and Input**

**Forms in PHP: Example (1)**

```php
<?php
// extract_names.php

function extract_names($url) {
    $text = file_get_contents($url);
    if ($text === false)
        return "ERROR: INVALID URL!"
    else {
        $correct = preg_match_all("/(name=(.*))\)/", $text, $matches, PREG_PATTERN_ORDER);
        if ($correct == 0) return "ERROR: NO NAMES FOUND"
        $count = array_count_values($matches[1]);
        arsort($count);
        foreach ($count as $name => $number) {
            $table .= "<tr><td>$name</td><td>$number</td></tr>"
        }
        $table = "<table><thead><tr><th>Name</th><th>No of occur</th></tr></thead><tbody>".$table."</tbody></table>
        return $table;
    }
}

extract_names("http://cgi.csc.liv.ac.uk/~ullrich/COMP519/examples/extract_names.php");
?>
```

**Revision and Further Reading**

Read
- Chapter 11: Form Handling

---

**Available Information and Input**

**Forms in PHP: Example (1)**

```php
<?php
// extract_names.php

function extract_names($url) {
    $text = file_get_contents($url);
    if ($text === false)
        return "ERROR: INVALID URL!"
    else {
        $correct = preg_match_all("/(name=(.*))\)/", $text, $matches, PREG_PATTERN_ORDER);
        if ($correct == 0) return "ERROR: NO NAMES FOUND"
        $count = array_count_values($matches[1]);
        arsort($count);
        foreach ($count as $name => $number) {
            $table .= "<tr><td>$name</td><td>$number</td></tr>"
        }
        $table = "<table><thead><tr><th>Name</th><th>No of occur</th></tr></thead><tbody>".$table."</tbody></table>
        return $table;
    }
}

extract_names("http://cgi.csc.liv.ac.uk/~ullrich/COMP519/examples/extract_names.php");
?>
```

**Revision and Further Reading**

Read
- Chapter 11: Form Handling

---

**Available Information and Input**

**Forms in PHP: Example (1)**

```php
<?php
// extract_names.php

function extract_names($url) {
    $text = file_get_contents($url);
    if ($text === false)
        return "ERROR: INVALID URL!"
    else {
        $correct = preg_match_all("/(name=(.*))\)/", $text, $matches, PREG_PATTERN_ORDER);
        if ($correct == 0) return "ERROR: NO NAMES FOUND"
        $count = array_count_values($matches[1]);
        arsort($count);
        foreach ($count as $name => $number) {
            $table .= "<tr><td>$name</td><td>$number</td></tr>"
        }
        $table = "<table><thead><tr><th>Name</th><th>No of occur</th></tr></thead><tbody>".$table."</tbody></table>
        return $table;
    }
}

extract_names("http://cgi.csc.liv.ac.uk/~ullrich/COMP519/examples/extract_names.php");
?>
```

**Revision and Further Reading**

Read
- Chapter 11: Form Handling

---

**Available Information and Input**

**Forms in PHP: Example (1)**

```php
<?php
// extract_names.php

function extract_names($url) {
    $text = file_get_contents($url);
    if ($text === false)
        return "ERROR: INVALID URL!"
    else {
        $correct = preg_match_all("/(name=(.*))\)/", $text, $matches, PREG_PATTERN_ORDER);
        if ($correct == 0) return "ERROR: NO NAMES FOUND"
        $count = array_count_values($matches[1]);
        arsort($count);
        foreach ($count as $name => $number) {
            $table .= "<tr><td>$name</td><td>$number</td></tr>"
        }
        $table = "<table><thead><tr><th>Name</th><th>No of occur</th></tr></thead><tbody>".$table."</tbody></table>
        return $table;
    }
}

extract_names("http://cgi.csc.liv.ac.uk/~ullrich/COMP519/examples/extract_names.php");
?>
```

**Revision and Further Reading**

Read
- Chapter 11: Form Handling