

COMP4021  
Internet Computing

# Handling File Uploads

David Rossiter and Gibson Lam

# Uploading Files

- Many web sites handle file uploading from the users' computer to their web site
- Major file sharing web sites include
  - Google drive <https://www.google.com/drive/>
  - Dropbox <https://www.dropbox.com/>
  - iCloud <https://www.icloud.com/>
- You can add the ability to upload files to your own web system

# Handling File Upload

- To handle file upload in the browser:
  - Make sure the `<form>` knows how to correctly send the file to the server
  - Put a file upload control in a `<form>`
- On the server side, PHP code has to be written so that the uploaded file is handled appropriately on the server

# The HTML Form for Handling Files

- Here is how you write the `<form>` tag to handle uploading files:

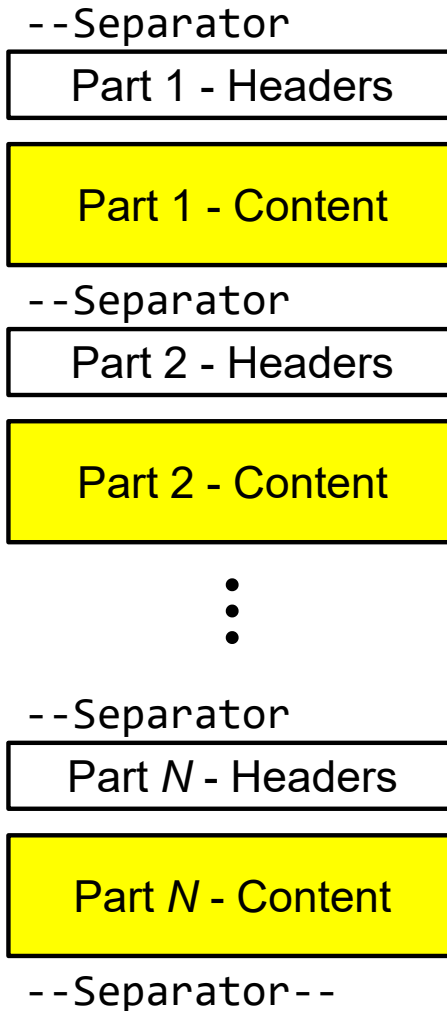
*The encoding type of the form; **must**  
**be set to multipart/form-data***

```
<form enctype="multipart/form-data"  
      method="post" action="target.php">  
                                =URL
```

*You **must use the POST**  
HTTP method*

# The Encoding Type

- The MIME type to send the form data is `multipart/form-data`
- This allows multiple pieces of content to be transmitted in the same transmission to the server
- On the right is an illustration of  $N$  parts being sent to the server in one transmission
- For example, you might be uploading several files at the same time



# Using the POST Method

- When a file upload is involved, the POST method is the only choice
- This makes sense because most files are too large to work with the GET method
  - The maximum for GET is a few thousand bytes

Completely useless for most images, videos, ect.

# The File Upload Input Field

- A file upload input field is written like this:

```
<input type="file" name="datafile">
```

- When the browser sees that in the HTML it shows a button and displays the selected filename, after it is selected:



# An Example HTTP Request

- Here is an HTTP POST request containing a file:

POST /target.php HTTP/1.1

*...some HTTP header information...*

Content-Type: multipart/form-data; separator  
boundary=Boundary-1234567890

*The section  
containing the  
uploaded file*

*The separator*

--Boundary-1234567890

Content-Type: application/pdf

Content-Disposition: form-data;

filename="timetable.pdf";  
name="datafile"

*...content of the file being sent to the server...*

--Boundary-1234567890--

*The separator*



# Handling the File Data in PHP

- To process the uploaded files in PHP, you do it through `$_FILES[]`
- You can access each uploaded file using the name of the input field
- For example, if the input field is named `datafile` , you can read the information in the file by:

```
$_FILES["datafile"]
```

# The Uploaded File Information

- Each uploaded file has an array of information:

<code>\$_FILES["datafile"]["name"]</code>	The filename
<code>\$_FILES["datafile"]["type"]</code>	The MIME type
<code>\$_FILES["datafile"]["tmp_name"]</code> the temporary location may disappear after a while	The temporary location of the file
<code>\$_FILES["datafile"]["size"]</code>	The file size


- These may or may not be useful, depending on what you want to do with the uploaded file

# An Example File Array

```
Array
(
    [datafile] => Array
        (
            [name] => timetable.pdf
            [type] => application/pdf
            [tmp_name] => C:\Temp\phpDAA7.tmp
            [error] => 0
            [size] => 36069
        )
)
```

- This structure shows example information for an uploaded file which had name="datafile" in the form, using `print_r($_FILES);`

`print_recursive`



*0 means there's no error;  
meaning the file has been  
successfully uploaded*

# Moving the Uploaded File

- The uploaded file has been put in a temporary location in the server
- You need to move the file in an appropriate location
  - otherwise the file will be deleted by PHP
- Use `move_uploaded_file()` , like this:
  - The file had name="datafile" in the form

```
move_uploaded_file(
```

```
$_FILES["datafile"]["tmp_name"],
```

```
"uploads/" . $_FILES["datafile"]["name"]
```

```
);
```

*Here . means  
concatenation*

*Move the file to a location under the folder  
"uploads", using the original name of the file*