Class 4

Serve-Side Validations

BY:

BRUNO ANGELO MEDEIROS
bruno.medeiros@sc.senai.br

IMAGES

EXTENSION

The **getimagesize()** should be the most definitive way of working out whether the file is an image. To check if a file is an image or not, you can use the function below:

```php
1.  function is_image($path)
2.  {
3.      $image = getimagesize($path);
4.      $image_type = $image[2];
5.
6.      if(in_array($image_type , array(IMAGETYPE_GIF , IMAGETYPE_JPEG ,IMAGETYPE_
    PNG , IMAGETYPE_BMP)))
7.      {
8.          return true;
9.      }
10.     return false;
11. }
```

The function provides a lot of information about an image file, including its type.

SIZE

If we need to validate the maximum or minimum size the uploaded image has, we can use the following code:

```php
1.  if ($_FILES["fileToUpload"]["size"] > 1024*1024*2) { // 2MB
2.      echo "Sorry, your file is too large.";
3.  }
```

DIMENSION

To verify the image width and height we can use:

```php
1.  list($width, $height) = getimagesize('path_to_image');
```

JSON

Structure

The function **json_last_error** returns the last error occurred during the JSON encoding and decoding. So the way to check the valid JSON is:

```php
1.  $result = json_decode($json);
2.  if (json_last_error() === JSON_ERROR_NONE) {
3.      // JSON is valid
4.  }
5.  // OR this is equivalent
6.  if (json_last_error() === 0) {
7.      // JSON is valid
8.  }
```

FILES

An example PHP Fileinfo function, to validate Office and other documents is:

```php
1.  $finfo = finfo_open(FILEINFO_MIME_TYPE);
2.  $mtype = finfo_file( $finfo, $filename );
3.  finfo_close($finfo);
```

FORMS

EMAIL

We use **filter_var()** mostly when we need to validate email. For Example:

```
1.  if (empty($_POST["email"])) {
2.      $emailErr = "Email is required";
3.  } else {
4.      $email = $_POST["email"];
5.      // check if e-mail address is well-formed
6.      if (!filter_var($email, FILTER_VALIDATE_EMAIL)) {
7.          $emailErr = "$email is not a valid email address";
8.      }
9.  }
```

EMPTY INPUTS

To validate if an input is empty we can use something like this:

```
1.  if (empty($_POST["name"])) {
2.      $nameErr = "Name is required";
3.  } else {
4.      $name = $_POST["name"];
5.      // check if name only contains letters and whitespace
6.      if (!preg_match("/^[a-zA-Z ]*$/",$name)) {
7.          $nameErr = "Only letters and white space allowed";
8.      }
9.  }
```

NUMBERS

We use **filter_var** also to validate integer numbers.

```
1.  if (empty($_POST["number"])) {
2.      $numberErr = "Number is required";
3.  } else {
4.      $number = $_POST["number"];
5.      // check if number is integer
6.      if (!filter_var($number, FILTER_VALIDATE_INT)) {
7.          $numberErr  = "$number is not a integer number.";
8.      }
9.  }
```

DATE

We use this function to check if the date is in the correct format.

```
1.  function validateDate($date)
2.  {
3.      $d = DateTime::createFromFormat('Y-m-d', $date);
4.      return $d && $d->format('Y-m-d') === $date;
5.  }
```

Note:

- Y is to represent a full year with 4 digits.
- m is a numeric representation of one month, with the zero to the left.
- d is a day of the month, made of 2 digits with the zero to the left.