The Form API

The Moodle Form API is strongly based on the PHP Extension and Application Repository (PEAR) HTML QuickForm Version 3.2 (https://pear.php.net/manual/en/package.html.html-quickform.php) libraries. This is an example of one of the legacy issues. The PEAR libraries have been re-written for PHP5 as HTML QuickForm2, which is currently at Version 2.2.0.

The Form API (cont)

- Most Moodle forms using this API are extensions of the `moodleform` class which is defined in the ***/lib/formslib.php*** file. The class is described as a '*wrapper that separates QuickForm's syntax from Moodle code*'
- Exception is activity modules which tend to extend the `moodleform_mod` class which provides module-specific functionality e.g. `standard_intro_elements()`

The Form API (cont)

The `moodleform` and the `moodleform_mod` classes do not extend the `HTML_QuickForm` class directly ...

Instead there is a property `$_form`, which is an instance of the `MoodleQuickForm` class, which, in turn, does extend the PEAR `HTML_QuickForm_DHTMLRulesTableless` class.

The Form API (cont)

- Elements of the form are defined in the `definition()` method which must always be overridden;
- Creating form elements normally requires the use the `MoodleQuickForm addElement()` method. Remember `MoodleQuickForm` is defined in the `$this->_form` property.

The Form API (cont)

Traditionally, the `$this->_mform` property is assigned to a `$mform` variable in the definition method for example:

```
public function definition() {
    $mform = $this->_form;
    // Adding the "general" fieldset, where all the common settings are showed.
    $mform->addElement('header', 'general', get_string('general', 'form'));
```

The `$mform` object is what is passed to our block plugin – receiving `$mform` is common for many plugins;

The Form API (cont)

The API supports:

- **Basic** elements, the standard HTML form elements.
- **Custom** elements are extensions and are created by calling the `registerElementType()` of the `MoodleQuickForm` class in most cases, extend an existing `MoodleQuickForm` element class. It is unlikely you will ever need this.
- **Advanced** elements are core Moodle ***custom*** elements. Handling file uploads is a special case and is discussed separately in this course.

The Form API (cont)

All elements added using the `$mform->addElement()` or created using the connected `$mform→createElement()` call expect at least three parameters:

- the element type – hidden, select etc
- the form name for the element
- the text for the form's prompt - usually by calling `get_string()` function

Most also expect additional parameters such as options for the select element.

Optionally, an attributes array parameter which defines the element's HTML attributes can be specified. The actual order where this attributes parameter is expected varies depending on the element's definition requirements.

A list of all the basic and advanced elements is provided as a handout in this course.

The Form API (cont)

- Only two required steps to add an element to a form;
  - Define the element - `$mform->addElement();`
  - Set the element's type with `$mform->setType()` by specifying the input's parameter type constants, such as `PARAM_TEXT`, `PARAM_INT` as per the 'Moodlelib API' discussed previously. UI elements, such as header, do not require this to be set.

There several optional but useful additional methods to be covered in a later lesson.