

## Introduction

Moodle stands for Modular Object Oriented Dynamic Learning Environment, it is an open source tool and is described as “a learning platform designed to provide educators, administrators and learners with a single robust, secure and integrated system to create personalised learning environments”. This tutorial describes some ways Moodle has been used to develop online learning materials for the ISIS Neutron and Muon Source Totara page, and the e-Neutrons online learning platform. Note that it does not seek to be a comprehensive course, but rather to provide an adequate introduction, alongside some specific methods that the author has found useful.

Many existing learning resources already cover the content in this course – online documentation on the use of Moodle, tutorials on HTML, etc. For this reason there are many links to relevant information throughout this course, which the reader is encouraged to make use of and explore further. Using them is not necessary to complete the tutorial, however this tutorial alone cannot be as comprehensive as these resources, which may also be more up to date. Hence, it is recommended that the reader use the referenced outside materials as a part of the course where possible.

At the time of writing (March 2019) 3.6 is the most current version of Moodle, however for the purposes of compatibility with the ISIS Totara pages and e-Neutrons, this tutorial will also show illustrations based on Moodle 3.5, the version used for ISIS Totara.

## 1. Starting your own Moodle Server

An excellent start to using Moodle is to set up a locally hosted Moodle server; this can then be used to try out and experiment with the various options Moodle offers in a controlled environment.

How to set up a Moodle server is detailed in its documentation:

- For a standalone Windows 7/8/10 computer see [https://docs.moodle.org/36/en/Complete\\_install\\_packages\\_for\\_Windows](https://docs.moodle.org/36/en/Complete_install_packages_for_Windows) (page also contains a link for instructions to Mac OS X packages)
- To set up a purpose dedicated server on a Linux computer see [https://docs.moodle.org/36/en/Installing\\_Moodle](https://docs.moodle.org/36/en/Installing_Moodle)

The following instructions describe the installation for a Windows computer as detailed in the documentation linked above.

First, check that your computer meets the system requirements to locally host the server. For Moodle 3.1 and later the requirements are a minimum of 256 MB RAM (though 1GB is recommended), 500 MB free Fixed Disk (note more space will be needed depending on user uploads), and Windows Vista/7/8/10/2008/2012.

The Microsoft Visual C++ 2015 Redistributable package is also required for PHP, and can be downloaded here: <https://www.microsoft.com/en-us/download/details.aspx?id=48145>. This must be the **vc\_redist.x86.exe** download as the PHP build is 32-bit.

Now you should install the complete Moodle package; this is a zip file containing a webserver called Apache, plus Moodle and Moodle's required MySQL database and PHP program. There are three parts to this process:

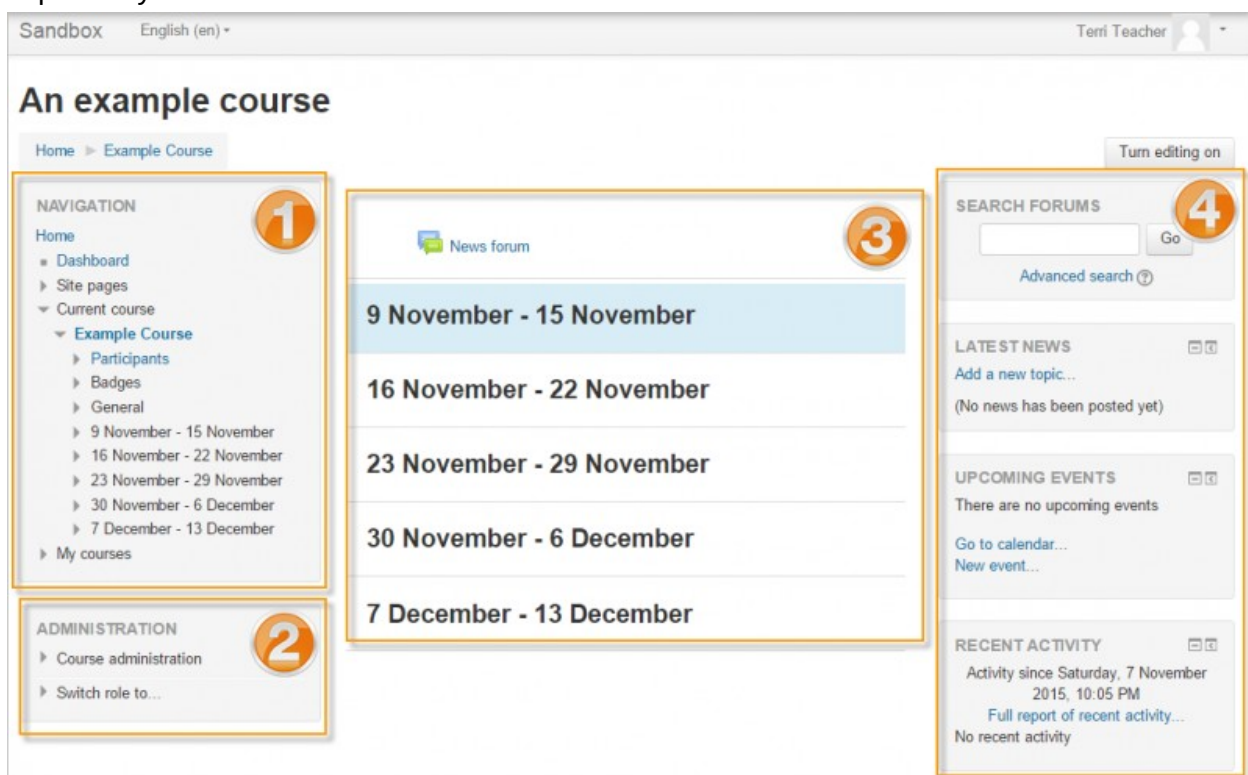
1. Download and unpack the complete install package from <https://download.moodle.org/windows> by extracting the zip file. Do **not** rename the "server" subfolder.
2. Start the web-server. Use the "Start Moodle.exe" file which you should find in the top directory you extracted. Once the "Start Moodle.exe" program is open, don't close it; use "Stop Moodle.exe" for that purpose.
3. Install Moodle using a web browser. Start your web browser and type localhost, or <http://127.0.0.1> or <http://localhost> in the address bar. You will either start your first time Moodle installation or if it is already installed you will enter the Moodle site's Front Page or Login screen.

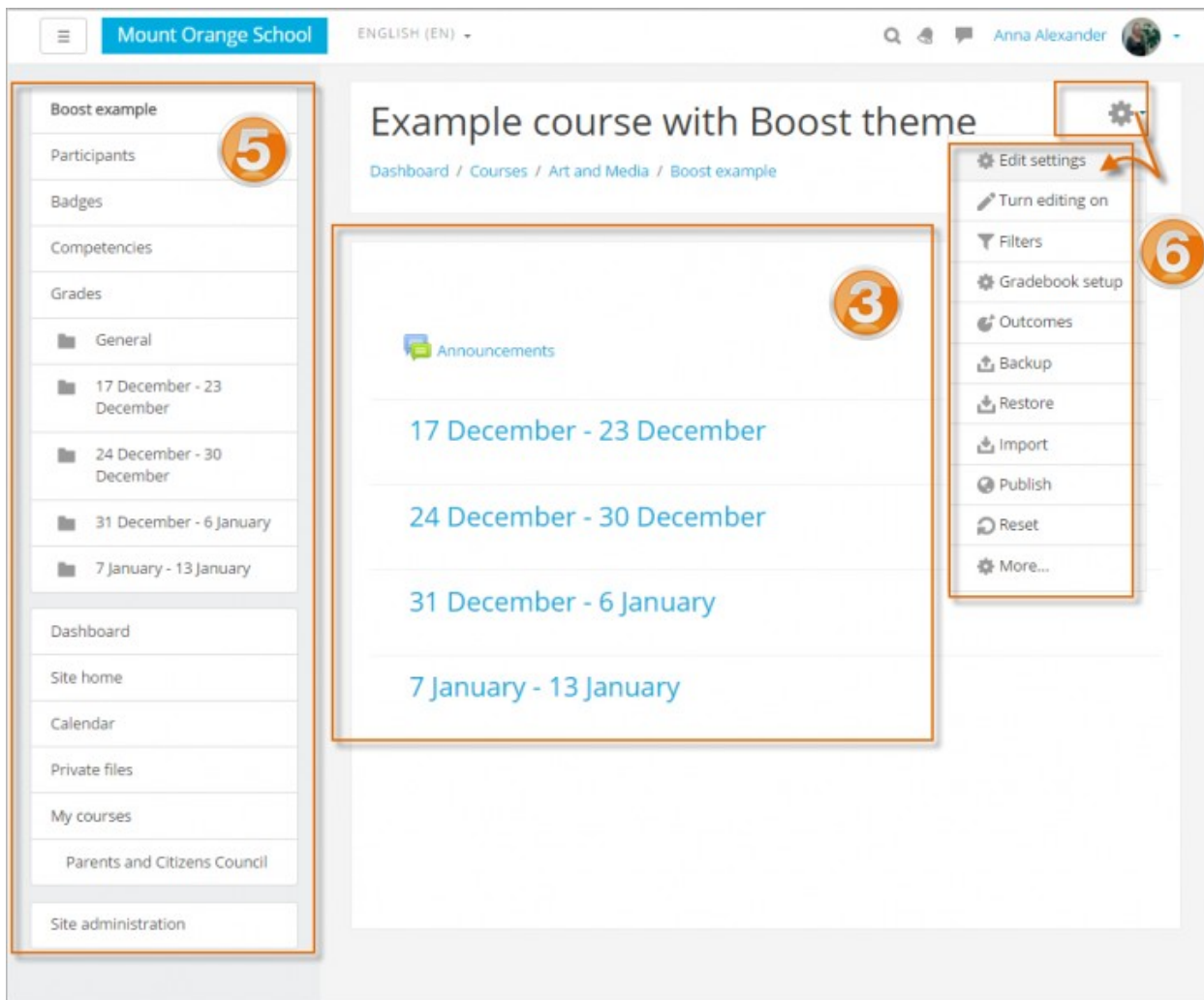
In addition to experimenting with Moodle's features, you can use your server to complete the remainder of this tutorial. The Moodle version of this tutorial, which covers all the content in this document as well as showcasing some of the things that can be done with Moodle, is found with this file and is called "Making Moodle Tutorials Backup.mbz". To continue this tutorial via the Moodle course, add it to your server by following the instructions on restoring courses in section "4.1 Course Restore" below.

## 2. Course Homepage and Editing Moodle Content

For documentation on the course homepage, see [https://docs.moodle.org/36/en/Course\\_homepage](https://docs.moodle.org/36/en/Course_homepage).

The main page of a Moodle server is the course homepage, the layout of which is determined by a theme. Some typical examples are shown below, these have the "clean" and "boost" themes respectively.





The labelled parts of the above images refer to the:

1. Navigation block

Normally visible on all pages, this block helps you find your way around the course and site.

2. Administration block

Again, normally visible on all pages, this block gives different levels of access to teachers and students.

3. Course sections

Here is where the learning materials are displayed. This element may be arranged in one or multiple weeks, topics, forums or other (non-standard) layouts.

4. Side blocks

Which blocks you see depend on what the administrator has selected and what you as teacher choose to add.

5. Navigation drawer

This replaces the Navigation block and Administration block in the Boost theme. For course section links to appear, 'Always link course sections' (linkcoursesections) must be enabled by an administrator in the Site administration.

## 6. Gear menu

This replaces the course administration settings in the Boost theme.

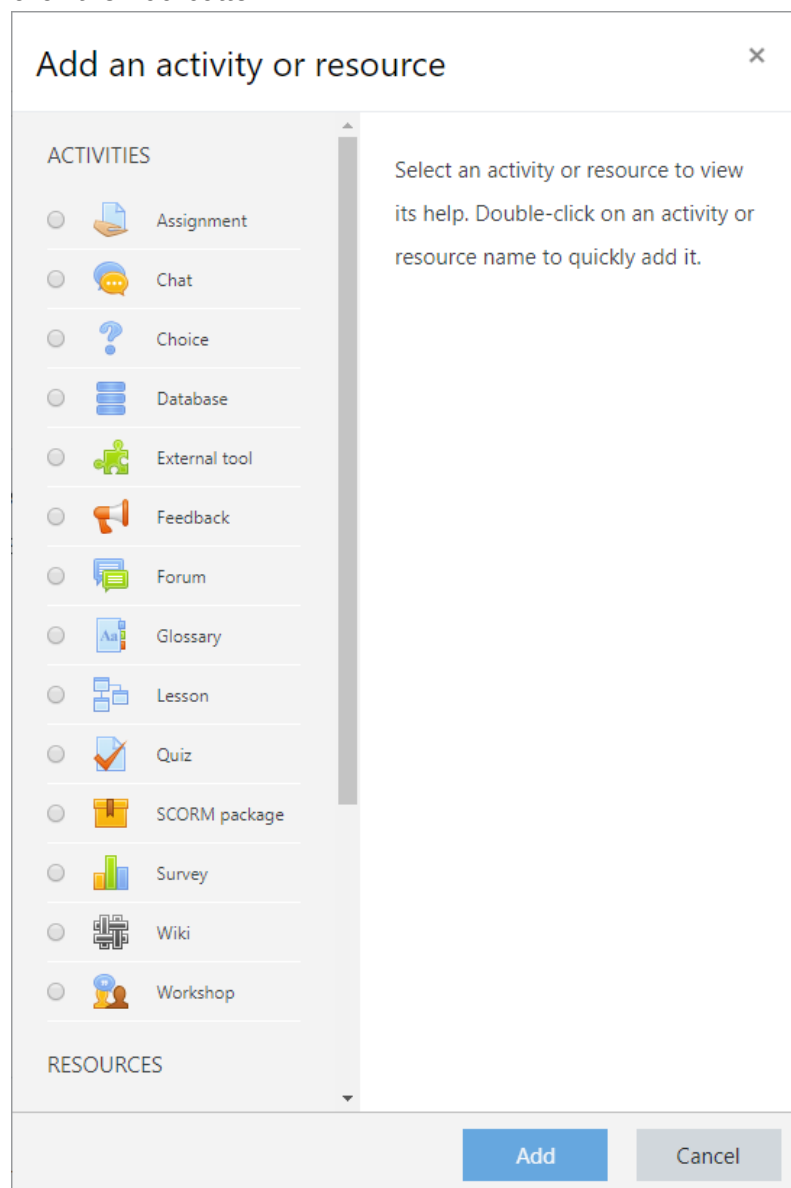
### 2. 1 Adding and Editing Content

Adding and editing activities and resources is described in Moodle's online documentation here:

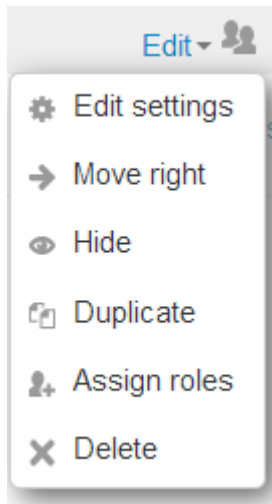
[https://docs.moodle.org/36/en/Course\\_homepage#Activities\\_and\\_resources](https://docs.moodle.org/36/en/Course_homepage#Activities_and_resources)

To add an activity or resource to the course, first turn on editing – either from the gear menu or administration block. Then click “Add activity or resource” to open the activity chooser, which is shown below. Select an activity or resource, then click the Add button.

You must turn on editing to alter an element on the course homepage; this shows icons representing different functions next to the resources, activity and topics. These icons perform different functions such as edit/move/copy/delete/hide.



The screenshot shows a modal dialog box titled "Add an activity or resource" with a close button (X) in the top right corner. The dialog is divided into two main sections: "ACTIVITIES" and "RESOURCES". The "ACTIVITIES" section is currently expanded and contains a list of 15 items, each with a radio button, an icon, and a label: Assignment, Chat, Choice, Database, External tool, Feedback, Forum, Glossary, Lesson, Quiz, SCORM package, Survey, Wiki, and Workshop. The "RESOURCES" section is collapsed. To the right of the list, there is a text box that says: "Select an activity or resource to view its help. Double-click on an activity or resource name to quickly add it." At the bottom right of the dialog, there are two buttons: "Add" (in blue) and "Cancel" (in grey).



Icons are often grouped together (left) for easier editing on smaller screens.



The pencil icon allows you to edit inline the title of the resource / activity



The move icon allows you to move items or sections by dragging and dropping.



This move icon appears if you don't have Ajax enabled.



The groups icon allows you to change between no groups or separate/visible groups



The move here icon appears when moving a course element without Ajax. Click into the box to relocate your item.



The up/down arrows allow you to move course sections up or down and appear if you do not have Ajax enabled.

To edit the text of resources and activities, Moodle has a text editor, the use of which is described here: [https://docs.moodle.org/36/en/Text\\_editor](https://docs.moodle.org/36/en/Text_editor)

The default text editor in Moodle is called Atto, it turns your text into HTML code which can be displayed online, functioning similarly to a word processor – although it does not have all the features one might expect from a program like Microsoft Word. There is also an alternative text editor called TinyMCE.

The basic features of the Atto toolbar are labelled and described below



1. Expands the toolbar
2. Selects text style, only heading types and main body text (paragraph) are available.
3. Makes selected text bold
4. Makes selected text italic
5. Puts selected text into a bulleted list
6. Puts selected text into a numbered list
7. Creates a link from the selected text

8. Unlink selected text
9. Add image
10. Add smiley (if enabled)
11. Add media
12. Record audio
13. Record video
14. manage embedded files

When the toolbar is expanded with button 1 the following options are also available:

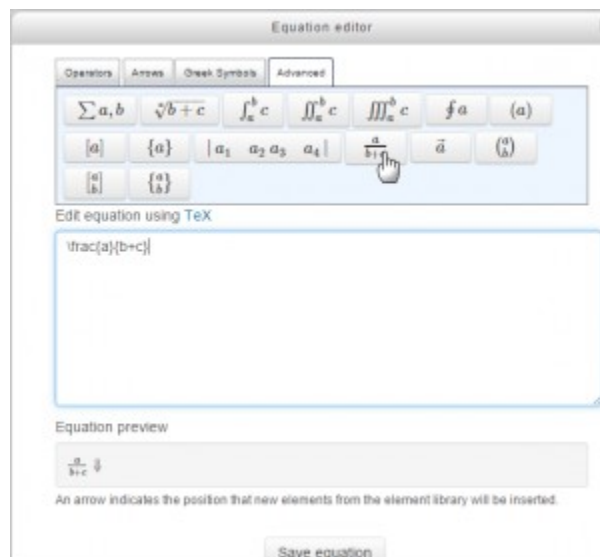


These buttons:

1. Underline the selected text
2. Strikethrough the text
3. Make selected text subscript
4. Make selected text superscript
5. Justifies text to the left/centre/right
6. Decrease/increase indent (works only on whole lines/paragraphs at a time)
7. Equation editor
8. Special charater button
9. Create table
10. Clear formatting
11. Undo/redo
12. Accessibility checker
13. Screenreader helper
14. View HTML code

When making scientific learning materials one function you are likely to use frequently is the Equation Editor, described here [https://docs.moodle.org/36/en/Text\\_editor#Equation\\_editor](https://docs.moodle.org/36/en/Text_editor#Equation_editor).

This allows you to write mathematical formulae using MathJax or TeX notation, while allowing you to view what you have written. You can also copy or write TeX directly into your text if you are confident it is correct. An example of writing an equation with TeX is shown below.



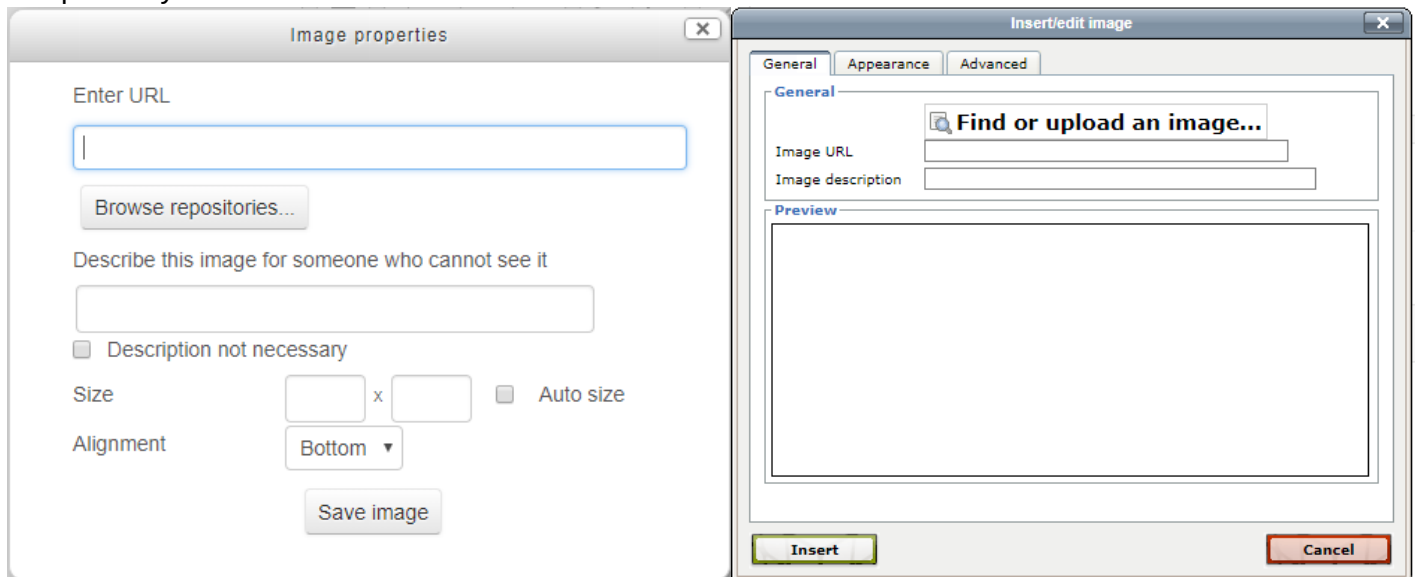
### 3. Adding Images

Adding images to your activities and resources is detailed in Moodle's documentation here: <https://docs.moodle.org/36/en/Images>

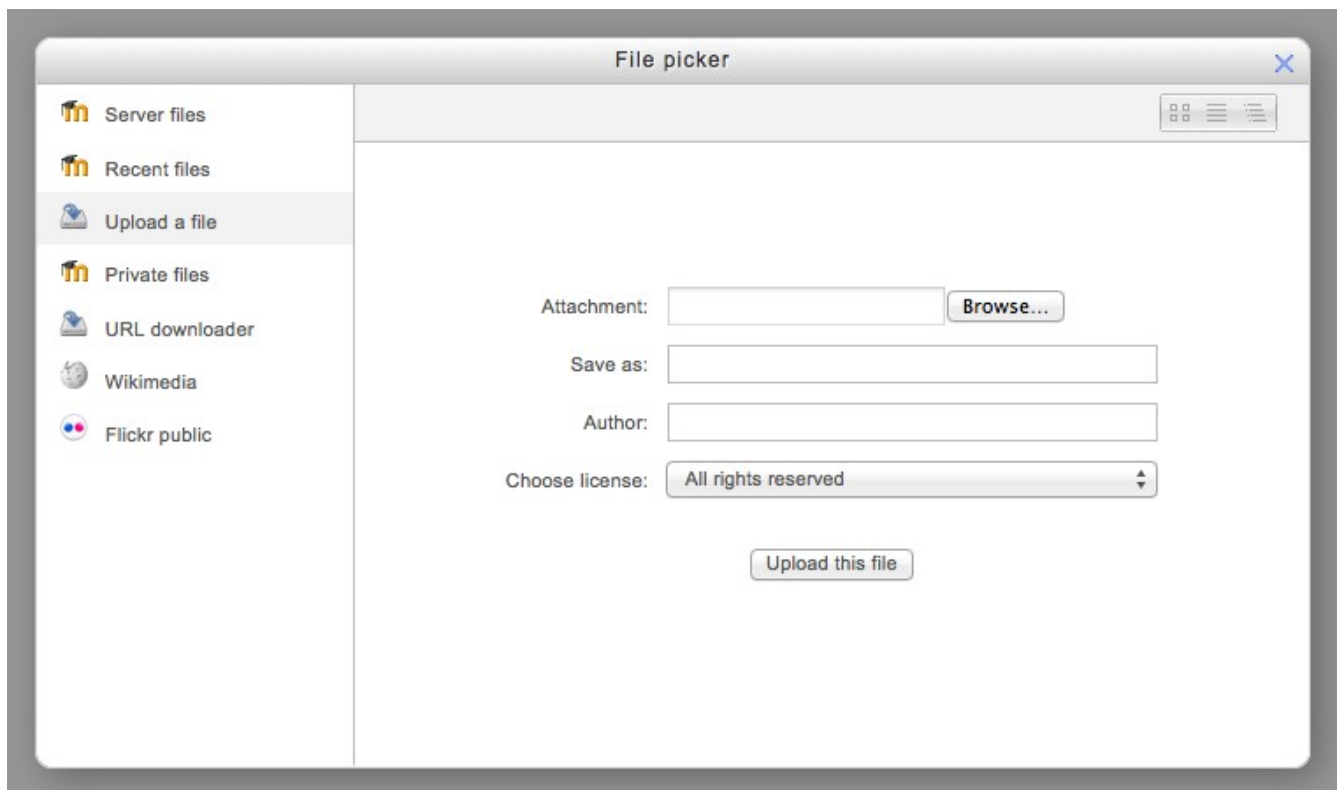
In the menu bar of your text editor (either Atto or TinyMCE) there will be a button to add an image to your page or lesson, which should look like this:



By clicking it you can choose to add images from your computer, or existing repositories on the server by using the “Browse repositories...” or “Find or upload an image...” buttons, in Atto and TinyMCE respectively.



Clicking one of these will open the File Picker window, more information on this can be found in Moodle's documentation here: [https://docs.moodle.org/36/en/File\\_picker](https://docs.moodle.org/36/en/File_picker). It is also shown below.



From here you can choose from files stored on the server, recently used files, uploading a file, private server files, URL downloader, files from Wikimedia and Flickr public.

Once you have uploaded or otherwise selected a file, the window will return to the upload window (for TinyMCE) or will show a new one. Either way, the window will have various options for the image properties – it will display the URL/location of the image, and allow you to write an image description for when the image can't be loaded, or the user has a screen reader. There are also options for the image's resolution and alignment, which are found in a separate "appearance" tab if you are using TinyMCE.

Other than choosing how your image appears, this window is also useful if you wish to reference the image in HTML, as the image URL given is the one which should be used as its src attribute in the code. For more on HTML attributes see [https://www.w3schools.com/html/html\\_attributes.asp](https://www.w3schools.com/html/html_attributes.asp). You can view the window again by right clicking on the image, then left clicking so it is selected in blue, and then using the add image button again, or simply double clicking in Atto.

#### 4. Creating Backups and Restoring Courses

To transfer complete Moodle pages from one site to another, see using Course Backup and Restore here: <https://docs.moodle.org/36/en/Backup>.



In a course, topic or activity you wish to backup, go to the Gear menu or Administration block and select “Backup” (shown on the right, along with “Restore”). This will lead you to a screen with options to select activities, blocks, filters and other items you wish to back up.

Users with appropriate permissions, such as administrators and managers can also include users, their information, groupings etc. It is recommended to not do this if you have the option; information on the user-base of learning resources without a set schedule is rarely of interest and is best left alone.

The next pages give options of which specific items (e.g. from a course) to back up, you can also pick by activity type. Finally, there is an option to review and then confirm before continuing; the backup should now be complete, and the file produced can be downloaded.

Downloading the file (rather than leaving it in Moodle’s database) is particularly useful for transferring content from one course or Moodle to another, via the Restore feature detailed below.

#### 4.1 Course Restore

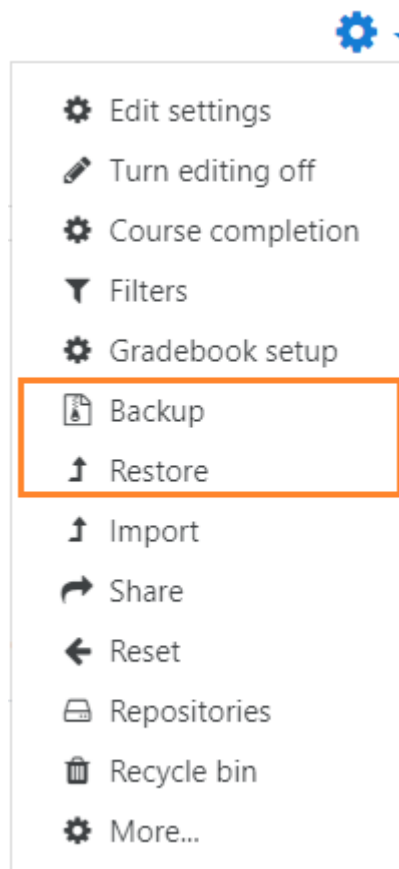
The Restore option is found in the same menu as the Backup button. Clicking it will allow you to upload a backup file, or choose one from the course backup area or user private backup area. Once you have picked a file, confirm that everything is as required and move on to choosing the destination – where the backup should be restored to. You can then select activities, blocks, filters etc. to restore, before going on to select which items to keep as well as changing other settings, such as course name and start date. Finally, you can review your choices and complete the process.

A version of this tutorial is available for Moodle 3.6 as a backup file; it covers all the content in this document as well as showcasing some of the things that can be done with Moodle. If you wish, you can restore this backup course to your own Moodle server, and continue this tutorial that way.

### 5. Using HTML in Moodle

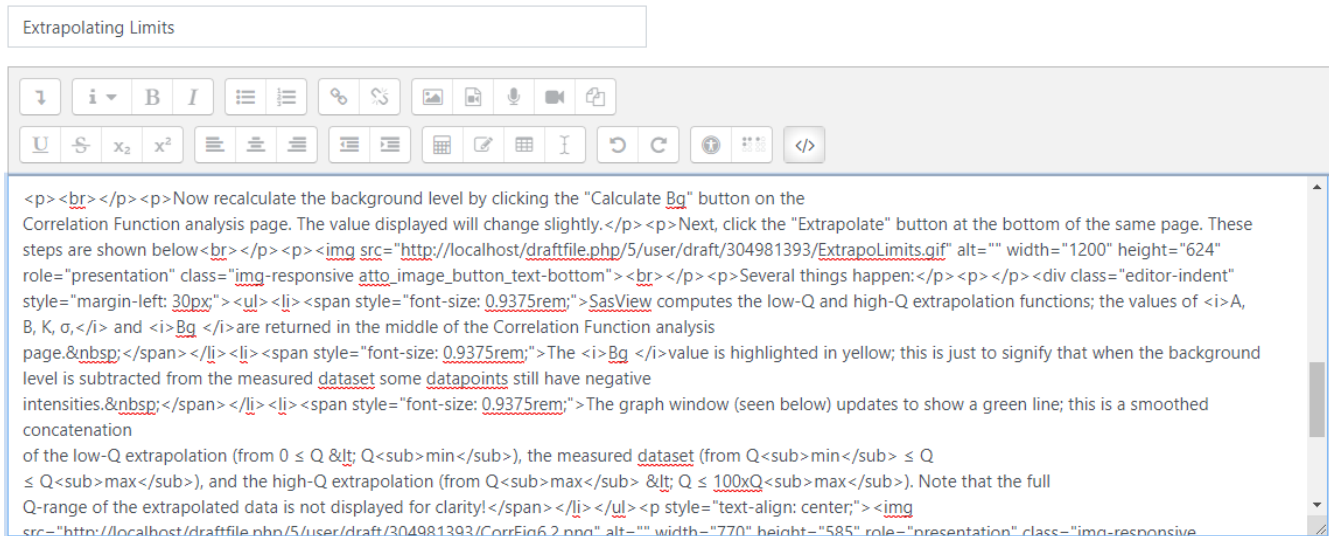
HTML is the standard code for rendering webpages, and this part of the course aims to give some advice on how to use it in conjunction with Moodle and its text editor(s), which are described here: [https://docs.moodle.org/36/en/Text\\_editor](https://docs.moodle.org/36/en/Text_editor). It does **not** aim to teach the reader HTML itself, for that an excellent tutorial is provided by w3 schools, found here: <https://www.w3schools.com/html/default.asp>.

By default, Moodle uses HTML for text formatting (although Moodle auto, Plain text and Markdown are sometimes available, as described here: [https://docs.moodle.org/36/en/Formatting\\_text](https://docs.moodle.org/36/en/Formatting_text)). This uses the Atto text editor by default, which is an excellent convenience for formatting text without requiring knowledge of HTML. However, it produces some very unpleasant looking code, which can be difficult to navigate if you have a desire or need to look at it directly. To look at the HTML code of a page, use the < > button in the expanded toolbar (labelled 14 in the section 2.1 Adding and Editing Content).



## Aside: Why write the HTML myself when Atto can do it for me?

You may wish to write HTML directly into your Moodle pages because it's much more flexible. The available text editors do not give you total freedom, and provide limited options of what you can write, so doing the code yourself is useful for making interactive features and other more novel purposes.



Above is an example of some HTML written by Atto. While it's functional, and looks fine when displayed, it is clear the code itself leaves much to be desired. Its layout doesn't automatically include indents or even new lines, and styles and tags appear which have no clear reason to be there.

The TinyMCE HTML editor, available as an alternative to Atto, also produces HTML. While it lacks some features, such as an equation editor (though one can still write TeX straight in), the code it produces can be automatically cleaned up so that it is much clearer and easier to navigate.

Unfortunately, ISIS Totara does not provide an option to use a non-default text formatting tool. With this in mind it may be best to, if possible, first write a page elsewhere using TinyMCE and then transfer the raw HTML code over. This will produce the same result as if the HTML had been written in Atto, except for when sourcing files local to the website (these will need to be uploaded to Totara and then their locations referenced individually).

If using Atto is unavoidable and you wish to code in HTML, consider what you will need to write directly as code and put that in first to save confusion. It can also be useful to put in paragraphs with a few words in them around these elements as placeholders, as text editors sometimes cope badly when trying to start new paragraphs which follow HTML not wrapped in a paragraph.

Alternatively, if you are comfortable writing whole documents in the HTML section of an editor, then writing purely in that will mean you have the flexibility desired and an easily navigable HTML version to edit or use elsewhere. Note, however, that one cannot use the Tab button while in a text editor, as Moodle uses Tab to skip to the next dialogue box. This makes laying out code with indents near impossible.

## 6. Displaying Other Resources

### 6.1 HTML Files

Rather than writing HTML using one of Moodle's editors, you can also upload HTML files for it to display. This can be useful if you wish to include a pop-up which stands on its own without necessarily connecting to any other pages, or an example page with specific formatting, or – perhaps most usefully – a page which uses javascript.

To do this upload an HTML file as a file resource either by simply dragging and dropping it into your course, or adding a File resource through the “Add an activity or resource” menu. Note that if your HTML relies on a CSS or Javascript file – which are used for easier formatting or adding functionality – you will need to add this file by editing the resource's settings and adding the new file. When you do this, be sure to set your HTML as the main file (click on it and select set main file). This is described in more detail here [https://docs.moodle.org/36/en/File\\_resource\\_settings](https://docs.moodle.org/36/en/File_resource_settings).

There are also various settings available on how a file will appear. The *embed* option displays the file within a Moodle page, *force-download* downloads the file to the user's computer, *open* simply opens the file in their browser, and *In pop-up* opens the file in a new window. If you want to display the file elsewhere on your Moodle – for instance, in frame in another page – it is best to choose *open* as your display setting to reduce clutter.

### 6.2 Outside Resources

To display content from a website outside of a Moodle site, you can use the URL resource as described here: [https://docs.moodle.org/36/en/URL\\_resource](https://docs.moodle.org/36/en/URL_resource). This is very similar to how you might use an uploaded file.

You may wish to display this sort of content alongside your own, within a lesson or page. This can be done through the use of HTML Iframes, for online instructions on this see: [https://www.w3schools.com/html/html\\_iframe.asp](https://www.w3schools.com/html/html_iframe.asp).

An Iframe is a closed HTML element defined by the <iframe> tag, it creates a window within a window which allows you to display content from one website within another. This can be useful for if you wish to display source material from another site alongside your own content. We will now go through an example of how to use an Iframe.

1. First, go into the HTML view of your Moodle text editor and make your Iframe element by typing <iframe></iframe>.
2. The next steps set the attributes the frame will have. These should be placed within the opening tag (after <iframe and before >) NOT between it and the closing tag.
3. The first attribute to set is a source website to display (src="#"); let's choose Moodle's documentation page on the URL resource (seen above). Note that not all websites can be displayed in frame, depending on their security and how they generate content.

4. We should also determine the style of the frame, which will allow us to choose its height, width, etc. Add the attribute style="#" into the opening tag of the iframe.
5. You can now put in style values to use (inside the quotes after style=), these can include many CSS style values as described here: [https://www.w3schools.com/html/html\\_css.asp](https://www.w3schools.com/html/html_css.asp). In this case we will just put in "height: 500px; width: 100%;" note the separation between height and width by a semicolon, and the colon to separate their values from the parameter itself. In this example we've determined the height and width differently – the height has a set value of 500 pixels, whereas the width will be 100% of the window the frame appears in.
6. Finally, as an optional step, we can give the Iframe a name, by adding the attribute name="#" with the chosen name replacing the #. This is useful for if you wish have a link which, when clicked, displays what it links to in the Iframe rather than its parent window/a new window etc.

## 7. Interactive Activity Example – Image Map

One way to introduce interactivity is through use of an HTML image map, image maps are included as part of the W3 Schools tutorial on images here [https://www.w3schools.com/html/html\\_images.asp](https://www.w3schools.com/html/html_images.asp).

Here we will go through an example of how image maps can be made and used, if you wish to know more about HTML consult the online tutorial from w3 schools.

Our first step is to add an image to a given activity or resource, how to do this is described in the "Adding Images" part of this course, and the image we will use can be found at <http://domaingang.com/wp-content/uploads/2012/02/example.png> or in the folder with this document. Next, switch to the HTML code view in your chosen editor and find the code which corresponds to the image. This will start with "<img" and contain the location/URL of the image given when you added it. A simple way to get an image URL is by right clicking on the image, then left clicking so it is selected in blue before using the add image button, or just double clicking in Atto.

Once you have located the relevant code (this may be difficult in Atto if the page contains other content) note how it appears. It may be inside a paragraph element <p></p>, this is not strictly necessary but will not stop us from making an image map, and can be useful for choosing how the image is justified. Within the image tag (after <img but before the >) there will be attributes of the image including height and width in pixels, in addition to the image's source (src) and potentially an "alt" attribute which gives a text alternative to the image.

In Atto the image will also have a "role" attribute which is associated with the image's purpose, and a "class" attribute which likely reads something like "img-responsive atto\_image\_button\_text-bottom". The attribute of most interest at this stage is the last one – the class – as the responsive part means that the image will change sizes with the window it is in. While this is useful in most contexts, it can mean a static map gets out of place as the image adjusts, so the class attribute and its value should be removed if present.

Now, to use an image map we must add a new attribute – usemap – to tell the image it will have a map, and to define which one to use if we have many. Put the attribute in the <img> tag along with the other attributes (not inside one of their values in quote marks!) and give it a value by setting it equal to something in quote marks, such as "#firstmap". We will then use this as the name of our map, in the

HTML view of your editor start a new tag by writing `<map name="firstmap">` using the name you picked, then on a new line write `</map>`, this end tag shows where code pertaining to the map ends.

In order to define which areas of the map should do something when clicked on, we add `<area>` elements to the map, *between the opening tag and closing tag*. These can have several attributes of their own, which are listed here: [https://www.w3schools.com/tags/tag\\_area.asp](https://www.w3schools.com/tags/tag_area.asp), the ones we are concerned with are shape, coords, href and target. Respectively, these determine the shape of an area which can be clicked on, the co-ordinates of that area, a hypertext reference to a file or website to link the area to, and a target where that link will be opened. Put one of each inside the area tag, after the "`<area`" but before the closing ">", separated by a space.

The shapes attribute has options of a rectangle (rect), circle (circle) or polygon (poly). This example will go through the use of a rectangle, so type `= "rect"` after the corresponding shape attribute (see w3 schools for more on the other options).

Once the type of shape is defined, its location and dimensions are specified in the co-ordinates (coords) attribute. For a rectangle this requires four values – the distance of each of the rectangle's sides from the left, top, right and bottom sides of the image, respectively. As you might imagine, estimating what these co-ordinates should be, or finding them by trial and error, can be somewhat time consuming. To solve this, online tools such as <https://www.image-map.net/> can help with finding the correct co-ordinates to use. Simply select an image from your computer or a website then click in two places to select where the corners of your rectangle should go, and look at the code to find the co-ordinates. For our test image, let us make the area over the word Test be our clickable area, the co-ordinates for this will be `= "58,184,413,289"`.

*Note:* These tools can provide a basis for all the code of an image map, however be cautious when doing this, as the image source it uses may not be correct, and they may not be able to reference some targets for the link, such as in frame displays.

Next, we will select what the area will link to, this can be a website or an HTML file, and in this case let us select the Wikipedia page for a test, [https://en.wikipedia.org/wiki/Test\\_\(assessment\)](https://en.wikipedia.org/wiki/Test_(assessment)), simply put this URL as the variable for href.

Finally, we must decide where this link will open, for this there are five options:

- `_blank` - Opens the linked document in a new window or tab
- `_self` - Opens the linked document in the same window/tab as it was clicked (this is default)
- `_parent` - Opens the linked document in the parent frame
- `_top` - Opens the linked document in the full body of the window
- `frameName` - Opens the linked document in a named frame, such as one opened in frame

In this case, let's select `_blank`, to open a new tab. If you had an `Iframe` on the same moodle page you could also put the name of that frame to display something there when your image map is clicked on.

Your code describing the image and map should now read something like this:

```


<map name="firstmap">
  <area shape="rect" coords="58,184,413,289"
  href="https://en.wikipedia.org/wiki/Test_(assessment)"
  target="_blank"/>
</map>
```

Save and display the Moodle page, when you mouse over the word Test in the image you should have the option to click, and when you do so it will open the Wikipedia page in a new tab! In order to add more clickable areas, simply put in more area elements inside the map. If you wish to have multiple images with multiple maps, simply create more map elements with different names to refer to them by.

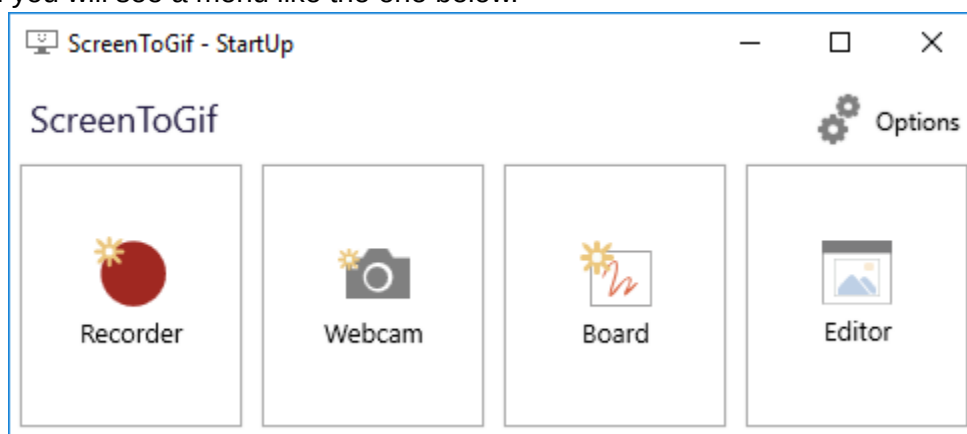
Interactive activities can also be made through the use of Javascript, however learning and implementing the Javascript required to produce these features is beyond the scope of this course. A useful tutorial can however be found here: <https://www.w3schools.com/js/default.asp>

Note that you cannot put Javascript directly into Moodle's HTML editor, but will instead have to use it in a pure HTML file, which you can then upload.

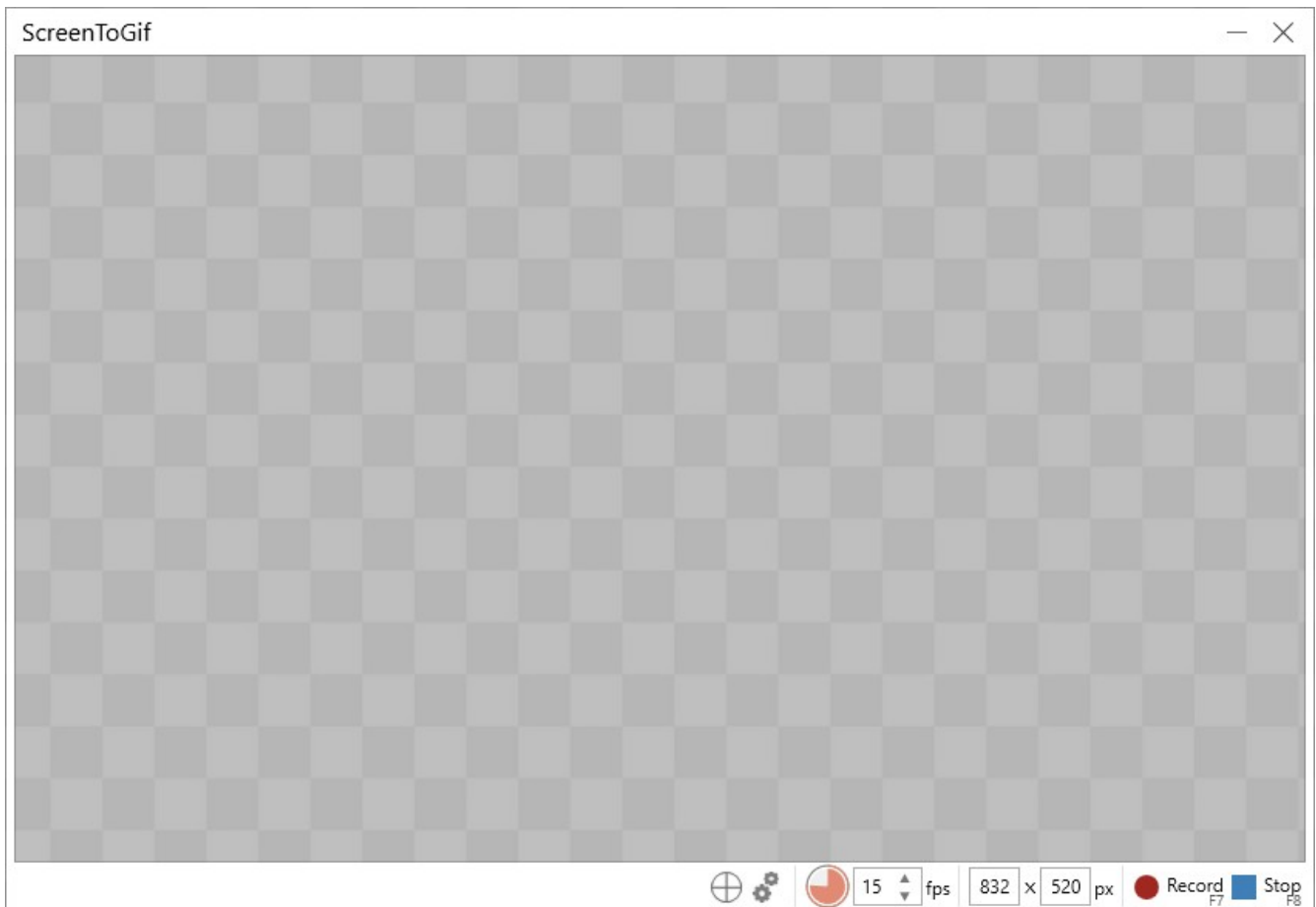
## 8. Making Gifs

In order to make gifs for animated illustrations or demonstrating the use of software, a very useful tool is ScreenToGif <https://www.screentogif.com/>, which does as its name would suggest – takes a recording of your screen and makes it into a gif. The software is free and open source, and intuitive to use. A user guide for it can be found here: <https://github.com/NickeManarin/ScreenToGif/wiki/help>. See below for the basics.

First, download and run the installer from the ScreenToGif website (you will need admin permissions to do this), the installer and set-up wizard will take you through the installation process. When you run the program you will see a menu like the one below.



The option which will be of most use is Recorder, which allows you to easily turn what you view on your screen into an animated gif. Selecting it brings up the following window frame.



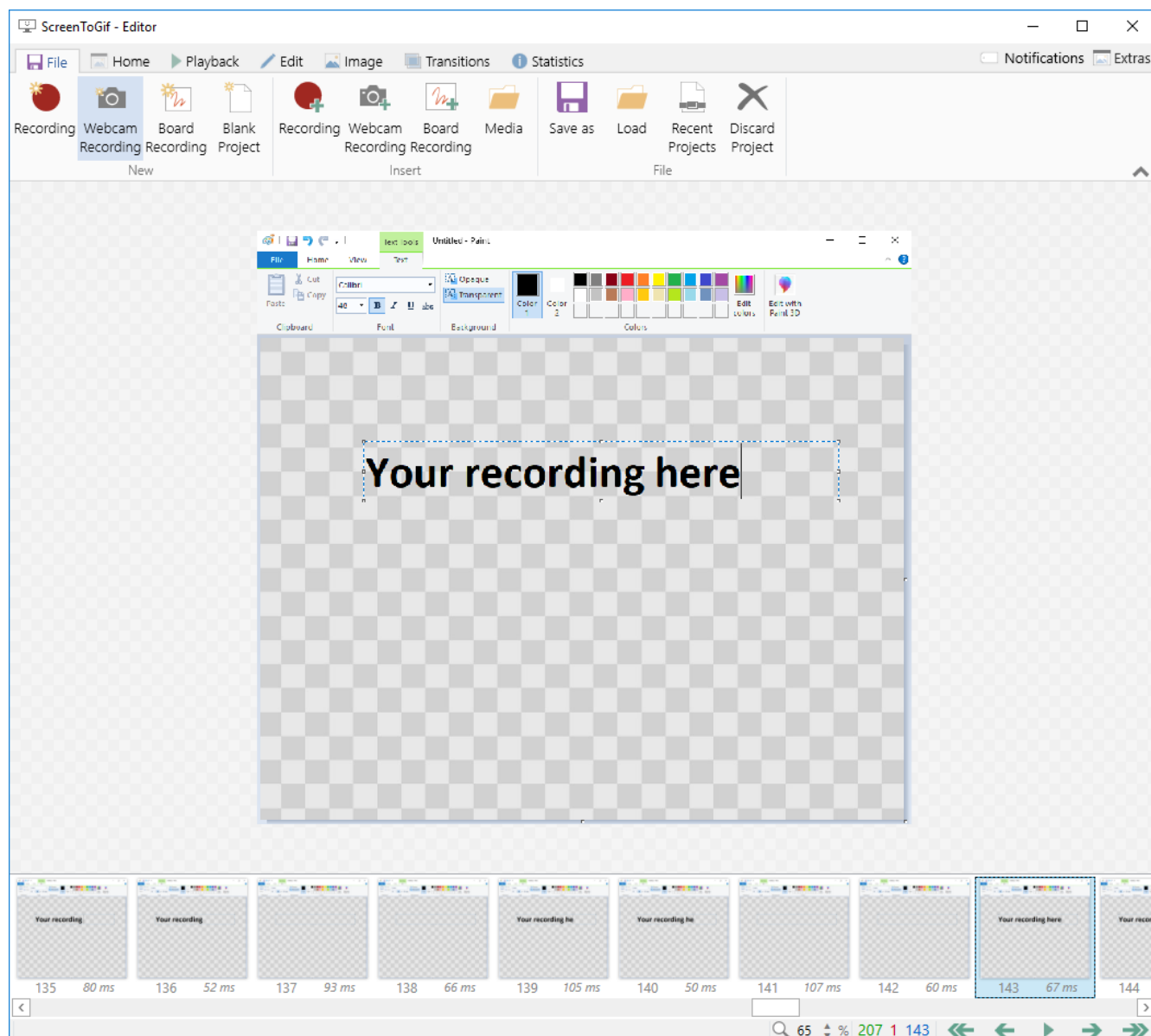
The chequered area is what will be recorded (it will be transparent in your view), and the window can be adjusted to the desired size and location by clicking and dragging its edges.

At the bottom right is a menu bar which – from left to right – allows you to target a window to record, change various settings, adjust the number of frames per second recorded, manually set the resolution, and start & stop recording.

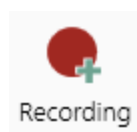
To make a recording, first line up the frame as desired – either manually or by clicking and holding the targetting tool and dragging it over the desired window. Next, click record to start capturing images of what is in the frame; while you're recording a new pause button will appear in the menu, for if you wish to stop recording and then resume. Once you are happy you have recorded what you need, click stop. Don't worry if you recorded for longer than strictly needed, as you can cut out excess frames in the next steps.



After you stop recording a new window (like the one shown below) will open, from here you can edit and save the project. For instance, the example below shows a capture of text being written in Paint, but if you look at the recorded images at the bottom of the window you will see that in some of them the text has vanished, then reappeared. This will result in a flickering effect in the final animation, but by clicking on the frames in which the text is absent and then deleting them we can remove this effect.



Other tools include adding extra frames before or after your first recording, using the recording p button:



Or playing back the recording to see how it looks in the **Playback** tab.



Once you are satisfied with the recording, click “save as” and pick where you would like to put the file, you can also choose to overwrite files of the same name. Clicking save will convert the images into a gif, though this process may take a couple of minutes.

