This is a blank rulebase (OPA version 12 August 2017) which incorporates GDS styles. Here follows a few usage notes:

**File list:**

* GDS.css  
  This styles the rulebase according to GDS standards.
* ntafont.css  
  This is the GDS font, note it is copyrighted for government use only. If your service is not licensed to use this font, then delete this file.
* custom.css  
  This includes a few things which are not specifically GDS, but are used to keep the rulebase in line with the general principles behind GDS standards.
* script.js  
  This is custom JavaScript used in the rulebase – explained below

**HTML and CSS usage:**

For the most part, you can create the rulebase in OPA as normal and it will look and feel like a normal government website. But there are a few things to keep in mind.

* Some styles aren’t immediately apparent.
  + The styles applied in the external CSS files are not apparent when editing screens.
  + If you want to see how it will look, you will need to click debug, this then applies all the styles.
* A version number can appear at the top of every screen.
  + Just add a label to the screen, probably near the top.
  + In it, add the attribute you want to appear (perhaps something called “versionNumber”.
  + Put it betwixt the following HTML:

<div class="onScreenVersion">

and

</div>

* + So:

<div class="onScreenVersion">%versionNumber%</div>

* + This will put it in the top-right corner of the screen – out of the way of the questions – but will enable you to see what version any screenshot was taken in (unless they have scrolled down).
* Hint text needs to be done manually.
  + OPA does not (yet) have the ability to add hint text in the way GDS recommends ([see examples here](https://www.gov.uk/service-manual/design/help-text#examples)). We therefore need to do this manually as part of the question text.
  + Note that here, we are referring to the short help text as “hint” text, and the longer expandable/collapsible help text as “help” text (which is dealt with below).
  + For hint text, add it to the question text in the following HTML

<br /><span class="hint-text">

and

</span>

* + So if the question “How many sheds do you have?” has the hint text “enter a number”, the question text should be this:

How many sheds do you have? <br /><span class="hint-text"> Enter a number</span>

* + Alternatively, you could use this:

<div class="hint-text">

and

</div>

* + So:

How many sheds do you have? <div class="hint-text"> Enter a number</div>

* + This second method is better for non-mandatory questions.

**JavaScript usage:**

Some bits of additional functionality are provided by the script.js file. These are:

1. The ability to hide on-screen questions.
2. The ability to have expandable help text ([as per GDS guidelines](https://www.gov.uk/service-manual/design/help-text#using-expanding-help-text)).
3. The ability to hide question labels.
4. Automatically adds “(optional)” to optional questions so you don’t have to.
5. Highlights the next button when errors are on the page
6. Hides invisible next buttons (turned off)
7. To hide on-screen questions:
   * This is used to create auto-answered questions. e.g. “This screen has been displayed”. Where you do not want the question to appear on the screen.
   * Be sure to set a default answer, otherwise the interview will crash.
   * In the question text, add “HideMe” to the beginning of the question. It is case-sensitive and there is no space betwixt “Hide” and “Me”.
   * The whole question will now be hidden, the user will be unable to interact with it. But it will behave as any other question answered by the user.
8. To add expandable help text:
   * Add 2 labels next to each other, one will be the help “title” and the other the help “content”.
   * Set the first to Heading 3
   * Encase the help title in the following HTML

<div class="help‑title" onclick="helpExpandCollapse(this)">

and

</div>

* + For example, if the help title was “What is help text?” it would look like this:

<div class="help‑title" onclick="helpExpandCollapse(this)"> What is help text?</div>

* + Encase the help content in the following HTML

<div class="help-content">

and

</div>

* + For example, if the help title was “Help text is text like this that appears when the title is clicked.” it would look like this:

<div class="help-content">Help text is text like this that appears when the title is clicked.</div>

1. Hiding a label.
   * Just like number 1, writing “NoLabel” will hide the label. Unlike number 1, this leaves the question visible and answerable.
   * Use for instances where question text is not required. One example of this would be for when collecting a postal address. Address line 2, which should be a non-mandatory field, does not need a label.
   * When writing in “NoLabel”, it must be case-sensitive and not be followed or preceded by anything – even a space – otherwise it won’t work as intended.
2. Optional questions
   * You can turn this off at the top of script.js
   * For the most-part, you do not need to do anything. The word “optional” in brackets will be added automatically to all non-mandatory questions.
   * It will not be added to questions with a NoLabel label.
   * Questions which have hint text but the “(optional)” after the hint text – this is why there are different approaches for hint text depending on whether the question is mandatory or not.
3. Errors on page
   * You can turn this off at the top of script.js
   * It will add a message beneath the next button if there are errors found on the page – it is useful for long pages, otherwise it isn’t needed.
   * The message can be customised in custom.css
4. Hide hidden buttons
   * **THIS IS NOW SUPERFLUOUS AS THE BACK BUTTON IS PUT AT THE TOP OF THE SCREEN AS PER GDS STANDARDS**
   * You can turn this on at the top of script.js – it is turned off by default
   * This is for assessments where there is a back button on the final screen.
   * This brings the back button in line with the other buttons.
5. Highlight errors
   * You can turn this on at the top of script.js – it is turned off by default
   * This only works when embedding.
   * It is designed for long pages when an error occurs, it will need to be repeated at the bottom.
6. Disable read only tab stop
   * You can turn this off at the top of script.js
   * This prevents read-only questions from being tabbed inoto.