* **What are the main benefits of HTL over JSP?** - Security and project efficiency are the main benefits, which are detailed on the [Overview](https://experienceleague.adobe.com/docs/experience-manager-htl/using/overview.html?lang=en).
* **Does HTL have any limitations that JSP doesn’t?** - HTL doesn’t really have limitations compared to JSP in the sense that what can be done with JSP should also be achievable with HTL. However, HTL is by design stricter than JSP in several aspects, and what can be achieved all in a single JSP file, might need to be separated into a Java class or a JavaScript file to be achievable in HTL. But this is generally desired to ensure a good separation of concerns between the logic and the markup.
* **Does HTL support JSP Tag Libraries?** - No, but as shown in the [Loading Client Libraries](https://experienceleague.adobe.com/docs/experience-manager-htl/using/getting-started/getting-started.html?lang=en#loading-client-libraries) section, the [template & call](https://experienceleague.adobe.com/docs/experience-manager-htl/using/htl/block-statements.html?lang=en#template-call) statements offer a similar pattern.
* **Can the HTL features be extended on an AEM project?** - No, they cannot. HTL has powerful extension mechanisms for reuse of logic - the [Use-API](https://experienceleague.adobe.com/docs/experience-manager-htl/using/getting-started/getting-started.html?lang=en#use-api-for-accessing-logic) - and of markup (the [template & call](https://experienceleague.adobe.com/docs/experience-manager-htl/using/htl/block-statements.html?lang=en#template-call) statements), which can be used to modularize the code of projects.