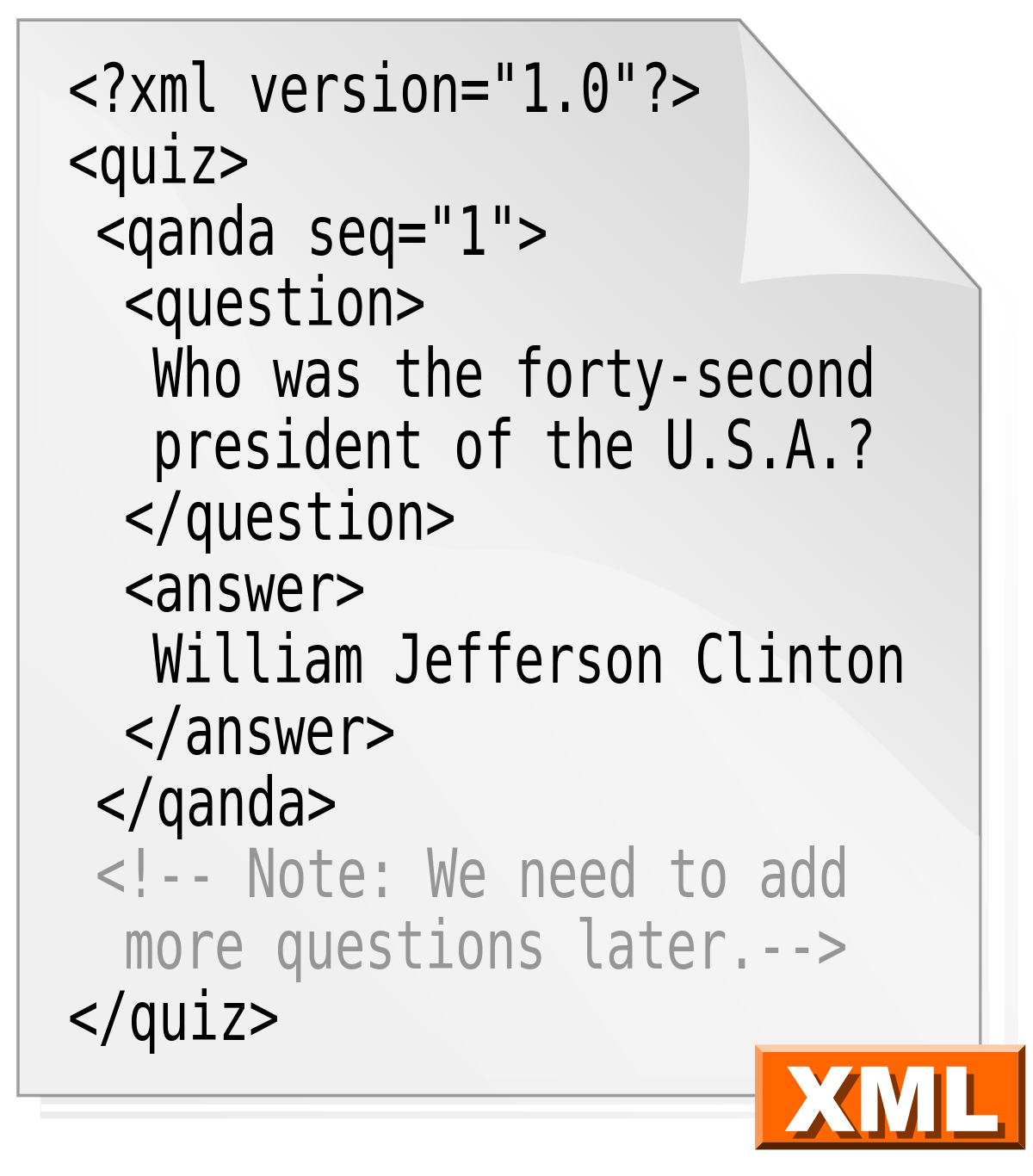
Languages

XML:

**Extensible Markup Language** (**XML**) is a markup language that defines a set of rules for encoding documents in a format that is both human-readable and machine-readable.



JSON:

**JavaScript Object Notation** (**JSON**) is an open-standard file format that uses human-readable text to transmit data objects consisting of attribute–value pairs and array data types (or any other serializable value). It is a very common data format used for asynchronous browser–server communication, including as a replacement for XML in some AJAX-style systems.

{

**"firstName"**: "John",

**"lastName"**: "Smith",

**"isAlive"**: **true**,

**"age"**: 27,

**"address"**: {

**"streetAddress"**: "21 2nd Street",

**"city"**: "New York",

**"state"**: "NY",

**"postalCode"**: "10021-3100"

},

**"phoneNumbers"**: [

{

**"type"**: "home",

**"number"**: "212 555-1234"

},

{

**"type"**: "office",

**"number"**: "646 555-4567"

},

{

**"type"**: "mobile",

**"number"**: "123 456-7890"

}

],

**"children"**: [],

**"spouse"**: **null**

}

PHP:

**PHP: Hypertext Preprocessor** (or simply **PHP**) is a server-side scripting language designed for Web development, and also used as a general-purpose programming language. It was originally created by [Rasmus Lerdorf](https://en.wikipedia.org/wiki/Rasmus_Lerdorf) in 1994, the PHP [reference implementation](https://en.wikipedia.org/wiki/Reference_implementation) is now produced by The PHP Group.[[6]](https://en.wikipedia.org/wiki/PHP#cite_note-about_PHP-6) PHP originally stood for *Personal Home Page*,[[5]](https://en.wikipedia.org/wiki/PHP#cite_note-History_of_PHP-5) but it now stands for the [recursive initialism](https://en.wikipedia.org/wiki/Recursive_initialism) *PHP: Hypertext Preprocessor*.

<!DOCTYPE html>

<html>

<head>

<title>PHP Test</title>

</head>

<body>

<?php **echo** '<p>Hello World</p>'; ?>

</body>

</html>

Comparison:

**PHP:**

Adv:

* Ideal for standard *web* development and rapid development
* **Easy integration with MySQL**
* $\_GET and $\_POST built into the language
* C-esque syntax
* Primarily used on an Apache server (Routinely, you will hear of a full LAMP stack: Linux OS, Apache server, MySQL database, and PHP)
* Opensource

Disadv:

* **Used only for basic and moderately advanced websites**
* Cant do advanced tasks

**Python:**  
Adv:

* A general purpose language
* **As noted previously, has extensive libraries**
* Requires a framework to be implemented when developing websites
* Syntax makes heavy use of whitespace, unlike C/C++, PHP, etc. which use { } to enclose blocks of code.
* Depending on the framework, your server can vary (...in a ridiculous nutshell.)
* **Very powerful and can do advanced tasks**
* Opensource

Disadv:

* **Requires a framework nad syntax wastes lot of white space**

**Ruby on Rails:**

**Advantages:**

* **Can be embedded into Hypertext Markup Language (HTML)**
* **Super advanced string and text manipulation techniques.**
* A Very High-Level Language (VHLL).
* It allows encapsulation of data methods within objects.
* Maintaining scalable and big programs written in Ruby is easy.
* Pure OOP (Object-Oriented Programming).
* Can be easily connected to MySQL, DB2, Oracle, and Sybase.
* Ability to write multithreaded applications with a simple API.
* It offers advanced array class.
* Better security features.
* It gives you the ability to write external libraries in Ruby or C.
* It has a debugger.
* It has flexible syntax.
* Powerful string handling.
* Opensource

**Limitations:**

* **The learning curve can be steep.**
* **Slower processing time (CPU time)**
* Development and updates are slower than ASP.Net.
* It doesn't have enough informational resources.

**.NET:**

**Advantages:**

* ASP.NET will significantly reduce the amount of code necessary to build large applications.
* **It comes with just-in-time compilation, native optimization, early binding and caching services right out of the box, all of which gives you better performance.**
* **Your applications are safe and secured, with built-in Windows authentication and per-application configuration.**
* The ASP.NET framework comes with a rich toolbox and designer in the form of the Visual Studio integrated development environment.
* The source code and HTML are together. Hence, ASP.NET pages are easy to write and maintain. Moreover, the source code is executed on the server, providing a lot of flexibility and power to the web pages.
* It gives you simplicity since ASP.NET makes it easy to perform usual tasks, from client authentication and simple form submission to deployment and site configuration.
* It allows you to keep your application constantly available to handle requests. All the processes are scrupulously managed and monitored by the ASP.NET runtime, so when a process ends a new one can be created in its place.
* It is entirely a server-side technology so, before it is sent to the browser, ASP.NET code executes on the server.
* It allows you to choose the language which best suits your application, **being language-independent.**
* The Web server continuously monitors the components, pages, and applications running on it. If it notices any infinite loops, memory leaks, other illegal activities, it immediately destroys those activities and restarts itself.
* It allows easy deployment. There is no need to register components since the configuration information is built-in.
* It is an application which runs faster and counters large volumes of users without facing performance problems.
* One significant advantage of ASP.NET is its **speed**. After a while, it generates native-code assemblies, and you're looking at C/C++ speed when handling requests.
* Made Opensource in 2014

**Limitations:**

* **Limited Development and Debugging Tools.**
* Mixes layout (HTML) and logic (scripting code).
* Interpreted and Loosely-Typed Code.
* No real state management.
* **Updates files only when the server is down.**
* Obscure Configuration Settings.
* Strongly typed languages are less than ideal for web development. Web development is inherently a dynamic environment.

**Java:**

Advantages:

 **Learning Curve**

**There is an extremely short learning curve** with Java as it is easy to write, compile, debug, than other major programming languages like C++, Objective-C, C#. That’s a huge benefit, because this ensures that Java programmers become productive in an extremely short span of time. Aside from that, what also makes Java more appealing is its fluent English-like syntax. This means that it is easy to read the logic.

 **Object Oriented Programming**

Here comes another interesting advantage of using Java. This allows you to create modular programs and reusable code keeping the system extensible as well as flexible.

 **Rich APIs**

Java offers APIs for various activities like Database connection, networking, I/O, XML parsing, utilities, and much more.

 **Powerful Opensource Rapid Development Tools**

Large numbers of opensource development tools that use Java have made Java an even more powerful option for developers i.e. Eclipse and Netbeans. They are extremely strong platforms and have contributed effectively in making Java what it is today. It makes coding simpler, and provides extremely powerful capabilities for debugging.

 **Open Source Libraries**

The large number of open source libraries and well matured with industrial support have also ensured that Java gets to be used everywhere. Some of these libraries have been contributed by many popular organizations like Apache, Google, Yahoo, Facebook, LinkedIn etc. That makes Java even more popular.

 **Free**

How could we miss on this point? When it comes to development, we are today faced with intense competition. Many small and medium enterprises want their software development and/or web app development done for their businesses, but they don’t really have a big budget for that. With Java being free, it makes it cost effective.

 **Community Support**

Finally, there is extensive community support that Java has managed to muster. That goes a long way in helping new Java developers learn the art and become productive soon.

* Opensource

Disadv:

 **Java compiler is not well optimized yet compared to C++.**

 There is no separation of specification from implementation.

 **Memory management, with Java, is a little expensive.**

 The lack of templates can limit the ability of Java to create high quality data structures.

 One can find some bugs in browsers and example programs.