PDF 1.7, the sixth edition of the PDF specification that became ISO 32000-1, includes some proprietary technologies defined only by Adobe, such as Adobe XML Forms Architecture (XFA) and JavaScript extension for Acrobat, which are referenced by ISO 32000-1 as normative and indispensable for the full implementation of the ISO 32000-1 specification.[9] These proprietary technologies are not standardized, and their specification is published only on Adobe's website.[10][11][12] Many of them are not supported by popular third-party implementations of PDF.

ISO published ISO 32000-2 in 2017, available for purchase, replacing the free specification provided by Adobe.[13] In December 2020, the second edition of PDF 2.0, ISO 32000-2:2020, was published, with clarifications, corrections, and critical updates to normative references[14] (ISO 32000-2 does not include any proprietary technologies as normative references).[15] In April 2023 the PDF Association made ISO 32000-2 available for download free of charge.[13]

Technical details[edit]
A PDF file is often a combination of vector graphics, text, and bitmap graphics. The basic types of content in a PDF are:

Typeset text stored as content streams (i.e., not encoded in plain text);
Vector graphics for illustrations and designs that consist of shapes and lines;
Raster graphics for photographs and other types of images
Multimedia objects in the document.
In later PDF revisions, a PDF document can also support links (inside document or web page), forms, JavaScript (initially available as a plugin for Acrobat 3.0), or any other types of embedded contents that can be handled using plug-ins.

PDF combines three technologies:

An equivalent subset of the PostScript page description programming language but in declarative form, for generating the layout and graphics.
A font-embedding/replacement system to allow fonts to travel with the documents.
A structured storage system to bundle these elements and any associated content into a single file, with data compression where appropriate.
PostScript language[edit]
PostScript is a page description language run in an interpreter to generate an image. It can handle graphics and has standard features of programming languages such as branching and looping. PDF is a subset of PostScript, simplified to remove such flow control features, while graphics commands remain.

Historically, the PostScript-like PDF code is generated from a source PostScript file (that is, an executable program), with standard compiler techniques like loop unrolling, inlining and removing unused branches, resulting in code that is purely declarative and static. This is then packaged into a container format, together with all necessary dependencies for correct rendering (external files, graphics, or fonts to which the document refers), and compressed.

As a document format, PDF has several advantages over PostScript:

PDF contains only static declarative PostScript code, that can be processed as data, and does not require a full program interpreter or compiler. This avoids the complexity and security risks of an engine with such a higher complexity level.
Like Display PostScript, since version 1.4 PDF supports transparent graphics, while standard PostScript does not.
PDF enforces the rule that the code for a page cannot affect any other pages. That rule is strongly recommended for PostScript code too, but has to be implemented explicitly, as PostScript is a full programming language that allows for such greater flexibilities and is not limited to the concepts of pages and documents.
All data required for rendering is included on the file itself, improving portability.[16]
Its disadvantages are:

Loss of flexibility, and limitation to a single use case.[citation needed]
A (sometimes much) larger size. Although for trivially repetitive content, this is mitigated with compression. (Overall, compared to e.g. a bitmap image, it is still orders of magnitude smaller.)[citation needed]
PDF since v1.6 supports embedding of interactive 3D documents: 3D drawings can be embedded using U3D or PRC and various other data formats.[17][18][19]

File format[edit]
A PDF file is organized using ASCII characters, except for certain elements that may have binary content. The file starts with a header containing a magic number (as a readable string) and the version of the format, for example %PDF-1.7. The format is a subset of a COS ("Carousel" Object Structure) format.[20] A COS tree file consists primarily of objects, of which there are nine types:[15]

Boolean values, representing true or false
Real numbers
Integers
Strings, enclosed within parentheses ((...)) or represented as hexadecimal within single angle brackets (<...>). Strings may contain 8-bit characters.
Names, starting with a forward slash (/)
Arrays, ordered collections of objects enclosed within square brackets ([...])
Dictionaries, collections of objects indexed by names enclosed within double angle brackets (<<...>>)
Streams, usually containing large amounts of optionally compressed binary data, preceded by a dictionary and enclosed between the stream and endstream keywords.
The null object
Comments using 8-bit characters prefixed with the percent sign (%) may be inserted.

Objects may be either direct (embedded in another object) or indirect. Indirect objects are numbered with an object number and a generation number and defined between the obj and endobj keywords if residing in the document root. Beginning with PDF version 1.5, indirect objects (except other streams) may also be located in special streams known as object streams (marked /Type /ObjStm). This technique enables non-stream objects to have standard stream filters applied to them, reduces the size of files that have large numbers of small indirect objects and is especially useful for Tagged PDF. Object streams do not support specifying an object's generation number (other than 0).

An index table, also called the cross-reference table, is located near the end of the file and gives the byte offset of each indirect object from the start of the file.[21] This design allows for efficient random access to the objects in the file, and also allows for small changes to be made without rewriting the entire file (incremental update). Before PDF version 1.5, the table would always be in a special ASCII format, be marked with the xref keyword, and follow the main body composed of indirect objects. Version 1.5 introduced optional cross-reference streams, which have the form of a standard stream object, possibly with filters applied. Such a stream may be used instead of the ASCII cross-reference table and contains the offsets and other information in binary format. The format is flexible in that it allows for integer width specification (using the /W array), so that for example, a document not exceeding 64 KiB in size may dedicate only 2 bytes for object offsets.

At the end of a PDF file is a footer containing

The startxref keyword followed by an offset to the start of the cross-reference table (starting with the xref keyword) or the cross-reference stream object, followed by
The %%EOF end-of-file marker.
If a cross-reference stream is not being used, the footer is preceded by the trailer keyword followed by a dictionary containing information that would otherwise be contained in the cross-reference stream object's dictionary:

A reference to the root object of the tree structure, also known as the catalog (/Root)
The count of indirect objects in the cross-reference table (/Size)