

[Click Here](#)



PDF to ppt

Trusted by over 10,000+ studentsWe are your complete resource for starting, growing, and monetizing an online business - from start to finish.Start a BlogI have over a decade of experience building and managing high-traffic authority websites that rank in the search engines and convert like crazy.I've developed a simple, proven process to scale the traffic, stickiness, and conversions of a website. Now I'm looking to teach those same methods to you.Digital marketing can be overwhelming, which is why we've done our best to simplify things down do their base levels.We've designed all of our resources with simplicity in mind. This is our guiding light, as we fully understand that learning digital marketing can be overwhelming at times. We've created our lessons to be simple to read, understand, and implement.The experts behind our lessons aren't just teaching - they're experienced professionals who have done this before. Rest assured you'll be learning from the best in the business.We don't just provide simple information - we back it up with helpful assets and offer actionable recommendations that you can take. Save yourself hours of searching with our reviews, curated collections, and insightful recommendations.Our team has spent years collecting real-world resources and examples that you can reference while building your online business. You'll see our suggestions implemented in the "workbooks", from people we've never even met. The Motorsport Images Collection captures events from 1895 to today's most recent coverage.Discover The CollectionCurated, compelling, and worth your time. Explore our latest gallery of Editors' PicksBrowse Editors' FavoritesExperience AI-Powered Creativity Getting files from Drive Getting files from Dropbox Uploading file 0 of 0 Time left - seconds - Upload speed - MB/s Converting PDF to POWERPOINT... Woops! Something is wrong with your Internet connection... Share -- copy and redistribute the material in any medium or format for any purpose, even commercially. Adapt -- remix, transform, and build upon the material for any purpose, even commercially. The licensor cannot revoke these freedoms as long as you follow the license terms. Attribution -- You must give appropriate credit , provide a link to the license, and indicate if changes were made. You may do so in any reasonable manner, but not in any way that suggests the licensor endorses you or your use. ShareAlike -- If you remix, transform, or build upon the material, you must distribute your contributions under the same license as the original. No additional restrictions -- You may not apply legal terms or technological measures that legally restrict others from doing anything the license permits. You do not have to comply with the license for elements of the material in the public domain or where your use is permitted by an applicable exception or limitation . No warranties are given. The license may not give you all of the permissions necessary for your intended use. For example, other rights such as publicity, privacy, or moral rights may limit how you use the material. The PDF was developed by Adobe back in the early 90s and it has become increasingly popular since the advent of the Internet and Social Media. PDF files typically contain both text and images and it is these images that can be used to increase the file size, in some cases dramatically. Many users, especially those at work, are not probably aware of the file size so that when they receive these files their mailboxes don't get blown up. They also want to ensure they don't take up all their allocated storage on their device. That is why compressing files, especially bigger files like PDF are so popular. You can use the Zamzar PDF compression tool to reduce the size of your PDF file without impacting the quality of your file thereby still allowing you to share or print these files. I am a student. I created one dissertation and saved it in PDF format with a password. However, I lost the password to open it. To get rid of the protection I tried many free solutions found online, but failed. Then I downloaded Passper for PDF to test it. I have to say that it is really easy to use. I paid for the Passper for PDF 1-year plan and it never disappointed me. If you are a student like me, I recommend it! By Jason M. Contreras Portable Document Format, a digital file format For other uses, see PDF (disambiguation). Portable Document FormatAdobe pdfFile name extension.pdfInternet media type application/pdf[1] application/x-pdf application/x-bzpdf application/x-gzpdf type codePDF [1] (including a single trailing space)Uniform Type Identifier (UTI)com.adobe.pdfMagic number%%PDFDeveloped byAdobe Inc. (1991–2008) ISO (2008–)Initial releaseJune 15, 1993, 31 years ago (1993-06-15)Latest release2.0 Extended toPDF/A, PDF/E, PDF/UA, PDF/VT, PDF/XStandardISO 32000-2Open formatYesWebsiteiso.org/standard/575839.html Portable Document Format (PDF), standardized as ISO 32000, is a file format developed by Adobe in 1992 to present documents, including text formatting and images, in a manner independent of application software, hardware, and operating systems.[2][3] Based on the PostScript language, each PDF file encapsulates a complete description of a fixed-layout flat document, including the text, fonts, vector graphics, raster images and other information needed to display it. PDF has its roots in "The Camelot Project" initiated by Adobe co-founder John Warnock in 1991.[4] PDF was developed as ISO 32000 in 2008.[5] The last edition as ISO 32000-2:2020 was published in December 2020.[6] PDF files may contain a variety of content besides flat text and graphics including logical structural elements, interactive elements such as annotations and form-fields, layers, rich media (including video content), three-dimensional objects using USD or PRC, and various other data formats. The PDF specification also provides for encryption and digital signatures, file attachments, and metadata to enable workflows requiring these features. Main article: History of PDF The development of PDF began in 1991 when John Warnock wrote a paper for a project then code-named Camelot, in which he proposed the creation of a simplified version of PostScript called Interchange PostScript (IPS).[7] Unlike traditional PostScript, which was tightly focused on rendering print jobs to output devices, IPS would be optimized for displaying pages to any screen and any platform.[7] Adobe Systems made the PDF specification available free of charge in 1993. In the early years PDF was popular mainly in desktop publishing workflows, and competed with several other formats, including DjVu, Envoy, Common Ground Digital Paper, Farallon Replica and even Adobe's own PostScript format. PDF was a proprietary format controlled by Adobe until it was released as an open standard on July 1, 2008, and published by the International Organization for Standardization as ISO 32000-1:2008.[8][9] at which time control of the specification passed to an ISO Committee of volunteer industry experts. In 2008, Adobe published a Public Patent License to ISO 32000-1 granting royalty-free rights for all patents owned by Adobe necessary to make, use, sell, and distribute PDF-compliant implementations.[10] 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.[11] These proprietary technologies are not standardized, and their specification is published only on Adobe's website.[12][13][14] Many of them are not supported by popular third-party implementations of PDF. ISO 32000-2, published in 2020, replaced the free specification provided by Adobe [15] In December 2020, the second edition of PDF 2.0, ISO 32000-2:2020, was published, with clarifications, corrections, and critical updates to eliminate references [16] (the 32000-2 does not include any proprietary technologies as normative references).[17] In April 2023 the PDF Association made ISO 32000-2 available for download free of charge.[15] 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; and Other multimedia objects. 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 a data compression where appropriate. PostScript is a page description language run in an interpreter to generate an image.[7] It can handle graphics and has standard features of programming languages such as branching and looping.[7] PDF is a subset of PostScript, simplified to remove such control flow features, while graphics commands remain.[7] PostScript was originally designed for a drastically different use case: transmission of one-way linear print jobs in which the PostScript interpreter would collect a series of commands until it encountered the showpage command, then execute all the commands to render a page as a raster image to a printing device.[18] PostScript was distributed for ISO 32000-1 as a normative reference, but it was not included in the file itself. The PDF specification is intended to facilitate the use of PDF specification or rendered pages.[18] If the PDF file was first put into the file in 2012. With the introduction of new print job in the form of an entirely new PostScript file. Thus, any given page in a PostScript file could be accurately rendered only as the cumulative result of executing all preceding commands to draw all previous pages—any of which could affect subsequent pages—plus the commands to draw that particular page, and there was no easy way to bypass that process to skip around to different pages.[18] Traditionally, to go from PostScript to PDF, a source PostScript file (that is, an executable program) is used as the basis for generating PostScript-like PDF code (see, e.g., Adobe Distiller). This is done by applying standard compiler techniques like loop unrolling, inlining and removing unused branches, resulting in code that is purely declarative and static.[18] The result 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. Modern applications write to printer drivers that directly generate PDF rather than going through PostScript first. 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.[18] This avoids the complexity and security risks of an engine with such a higher complexity level. Like Display PostScript, PDF has supported transparent graphics since version 1.4, while standard PostScript does not. PDF enforces the rule that the code for any particular page cannot affect any other pages.[18] That rule is strongly recommended for PostScript code too, but has to be implemented explicitly (see, e.g., the Document Structuring Conventions), as PostScript is a full programming language that allows for such greater flexibilities and it is not limited to the concepts of pages and documents. All data required for rendering is included within the file itself, improving portability.[19] Its disadvantages are: A loss of flexibility, and limitation to a single-use case (citation needed) A (sometimes much) larger file size.[20] PDF files are limited to a maximum of 32,768 objects, 3D drawings can be embedded using USD or PRC, and various other data formats [21][22][23] 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.[24] A COS tree of file consists primarily of objects, which there are nine types:[17] Boolean values, representing true or false Real numbers Integers Strings, enclosed within parentheses (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; and Other multimedia objects. 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 a data compression where appropriate. PostScript is a page description language run in an interpreter to generate an image.[7] It can handle graphics and has standard features of programming languages such as branching and looping.[7] PDF is a subset of PostScript, simplified to remove such control flow features, while graphics commands remain.[7] PostScript was originally designed for a drastically different use case: transmission of one-way linear print jobs in which the PostScript interpreter would collect a series of commands until it encountered the showpage command, then execute all the commands to render a page as a raster image to a printing device.[18] PostScript was distributed for ISO 32000-1 as a normative reference, but it was not included in the file itself. The PDF specification is intended to facilitate the use of PDF specification or rendered pages.[18] If the PDF file was first put into the file in 2012. With the introduction of new print job in the form of an entirely new PostScript file. Thus, any given page in a PostScript file could be accurately rendered only as the cumulative result of executing all preceding commands to draw all previous pages—any of which could affect subsequent pages—plus the commands to draw that particular page, and there was no easy way to bypass that process to skip around to different pages.[18] Traditionally, to go from PostScript to PDF, a source PostScript file (that is, an executable program) is used as the basis for generating PostScript-like PDF code (see, e.g., Adobe Distiller). This is done by applying standard compiler techniques like loop unrolling, inlining and removing unused branches, resulting in code that is purely declarative and static.[18] The result 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. Modern applications write to printer drivers that directly generate PDF rather than going through PostScript first. 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.[18] This avoids the complexity and security risks of an engine with such a higher complexity level. Like Display PostScript, PDF has supported transparent graphics since version 1.4, while standard PostScript does not. PDF enforces the rule that the code for any particular page cannot affect any other pages.[18] That rule is strongly recommended for PostScript code too, but has to be implemented explicitly (see, e.g., the Document Structuring Conventions), as PostScript is a full programming language that allows for such greater flexibilities and it is not limited to the concepts of pages and documents. All data required for rendering is included within the file itself, improving portability.[19] Its disadvantages are: A loss of flexibility, and limitation to a single-use case (citation needed) A (sometimes much) larger file size.[20] PDF files are limited to a maximum of 32,768 objects, 3D drawings can be embedded using USD or PRC, and various other data formats [21][22][23] 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.[24] A COS tree of file consists primarily of objects, which there are nine types:[17] Boolean values, representing true or false Real numbers Integers Strings, enclosed within parentheses (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; and Other multimedia objects. 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 a data compression where appropriate. PostScript is a page description language run in an interpreter to generate an image.[7] It can handle graphics and has standard features of programming languages such as branching and looping.[7] PDF is a subset of PostScript, simplified to remove such control flow features, while graphics commands remain.[7] PostScript was originally designed for a drastically different use case: transmission of one-way linear print jobs in which the PostScript interpreter would collect a series of commands until it encountered the showpage command, then execute all the commands to render a page as a raster image to a printing device.[18] PostScript was distributed for ISO 32000-1 as a normative reference, but it was not included in the file itself. The PDF specification is intended to facilitate the use of PDF specification or rendered pages.[18] If the PDF file was first put into the file in 2012. With the introduction of new print job in the form of an entirely new PostScript file. Thus, any given page in a PostScript file could be accurately rendered only as the cumulative result of executing all preceding commands to draw all previous pages—any of which could affect subsequent pages—plus the commands to draw that particular page, and there was no easy way to bypass that process to skip around to different pages.[18] Traditionally, to go from PostScript to PDF, a source PostScript file (that is, an executable program) is used as the basis for generating PostScript-like PDF code (see, e.g., Adobe Distiller). This is done by applying standard compiler techniques like loop unrolling, inlining and removing unused branches, resulting in code that is purely declarative and static.[18] The result 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. Modern applications write to printer drivers that directly generate PDF rather than going through PostScript first. 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.[18] This avoids the complexity and security risks of an engine with such a higher complexity level. Like Display PostScript, PDF has supported transparent graphics since version 1.4, while standard PostScript does not. PDF enforces the rule that the code for any particular page cannot affect any other pages.[18] That rule is strongly recommended for PostScript code too, but has to be implemented explicitly (see, e.g., the Document Structuring Conventions), as PostScript is a full programming language that allows for such greater flexibilities and it is not limited to the concepts of pages and documents. All data required for rendering is included within the file itself, improving portability.[19] Its disadvantages are: A loss of flexibility, and limitation to a single-use case (citation needed) A (sometimes much) larger file size.[20] PDF files are limited to a maximum of 32,768 objects, 3D drawings can be embedded using USD or PRC, and various other data formats [21][22][23] 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.[24] A COS tree of file consists primarily of objects, which there are nine types:[17] Boolean values, representing true or false Real numbers Integers Strings, enclosed within parentheses (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; and Other multimedia objects. 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 a data compression where appropriate. PostScript is a page description language run in an interpreter to generate an image.[7] It can handle graphics and has standard features of programming languages such as branching and looping.[7] PDF is a subset of PostScript, simplified to remove such control flow features, while graphics commands remain.[7] PostScript was originally designed for a drastically different use case: transmission of one-way linear print jobs in which the PostScript interpreter would collect a series of commands until it encountered the showpage command, then execute all the commands to render a page as a raster image to a printing device.[18] PostScript was distributed for ISO 32000-1 as a normative reference, but it was not included in the file itself. The PDF specification is intended to facilitate the use of PDF specification or rendered pages.[18] If the PDF file was first put into the file in 2012. With the introduction of new print job in the form of an entirely new PostScript file. Thus, any given page in a PostScript file could be accurately rendered only as the cumulative result of executing all preceding commands to draw all previous pages—any of which could affect subsequent pages—plus the commands to draw that particular page, and there was no easy way to bypass that process to skip around to different pages.[18] Traditionally, to go from PostScript to PDF, a source PostScript file (that is, an executable program) is used as the basis for generating PostScript-like PDF code (see, e.g., Adobe Distiller). This is done

[illegible]