Journey with the evolution of the web :

-The names of the web developments are 1.0, 2.0 and 3.0 in the media

Web 1.0
"Read Only",
Decentralized
Web 2.0
Participatory,
Centralized
Web 3
No Intermediaries,
Decentralized

First there was Web 1.0 :

- Web 1.0 was the first stage of the World Wide Web revolution, usually referred to as read-only web. This was how the Internet we know today started in the first place, where the websites were merely informational and comprised entirely static content; they were only linked together by hyperlinks and lacked any interactive content or design elements.

- This term would be used to describe the simple "shop front" websites of the past.

- It spawned the dot-com bubble, which lasted from 1995 to 2000 and included many internet-based businesses. These were the companies that emerged with Web 1.0:

Secondly there was Web 2.0 :

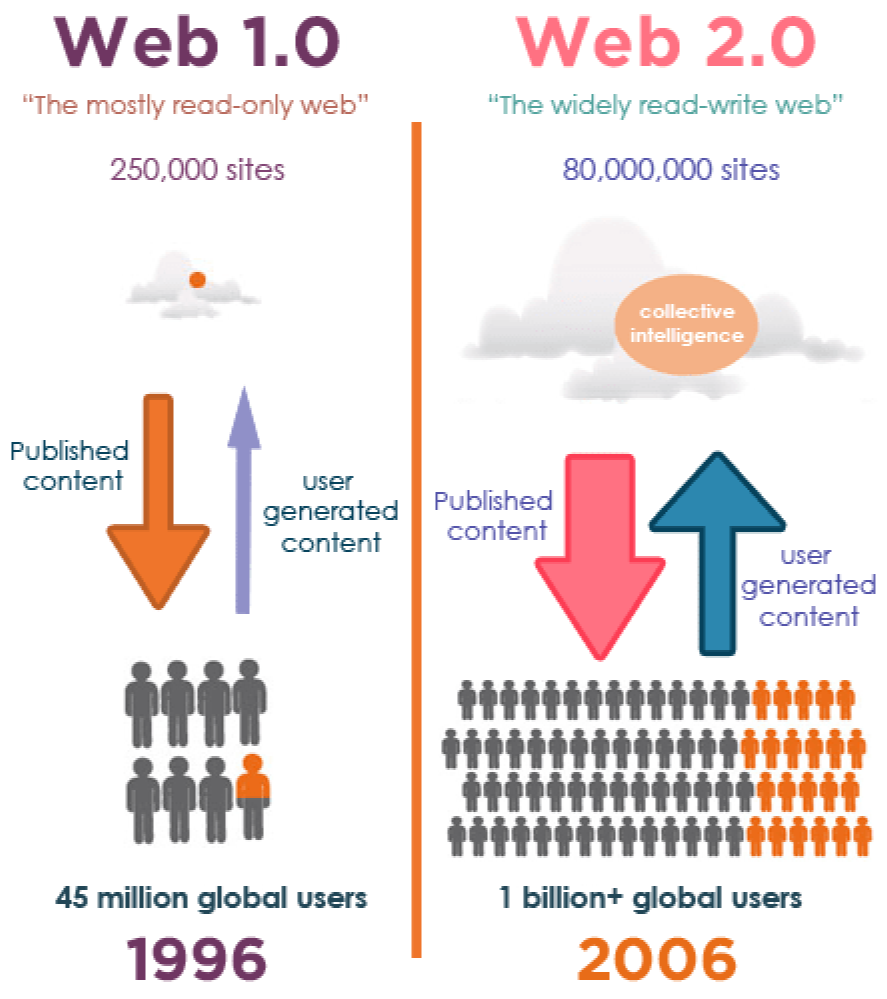
- Web 2.0 has been used to describe this web phenomenon. Examples include social networking sites such as Facebook and MySpace, which allowed users to create their own profiles, share files and interact with each other on a larger scale.

- And a site like YouTube allowed people to publish their own media content

- In summary, if Web 2.0 has stood for anything it is the creation and sharing of information delivered via the web.

- Web 2.0 was the second stage of the evolution of the web, also called the read-write web and it was the phase when websites grew in terms of user interaction.

- It was the period when websites became more focused on user-generated content, usability, and interoperability for end-users, leading them to become the - participative social web.



- During Web 2.0, terms like blogs, social media, and video streaming gained popularity. This time period is also acknowledged for the ease with which music and video clips could be exchanged.

- It opened doors to podcasting, blogging, tagging, curating with RSS, social bookmarking, social networking, social media, web content voting, etc. It was the birthplace of Youtube, Wiki, Flickr, Facebook, and so on.

- Also, blogging became popular with the introduction of WordPress which started as a PHP & MySQL-led blogging platform and has now advanced to become a full content managed system (CMS) which powers over a quarter of the web and e-Commerce completely revolutionized the way we shop

- Thus, Web 2.0 brought a fundamental shift where people were allowed to share their perspectives, opinions, thoughts, and experiences via a number of online tools and platforms.

- It brought us the concept - ' Web as Platform ', where software applications are built upon the Web as opposed to upon the desktop

- This was when websites began using web browser technologies such as AJAX and Javascript frameworks.

- This period continued to see the origin of APIs(Application Programming Interface) - a software intermediary that allows two applications to communicate with one another.

now we are in Web 3.0 :

- The future of Web 3.0 points to universal applications which can be read and used by a large number of devices and software types, making the ways in which we indulge business and leisure increasingly convenient.

- The one main concept with Web 3.0 is "the data web" or "semantic web", which in principle involves making structured data available on the Internet.

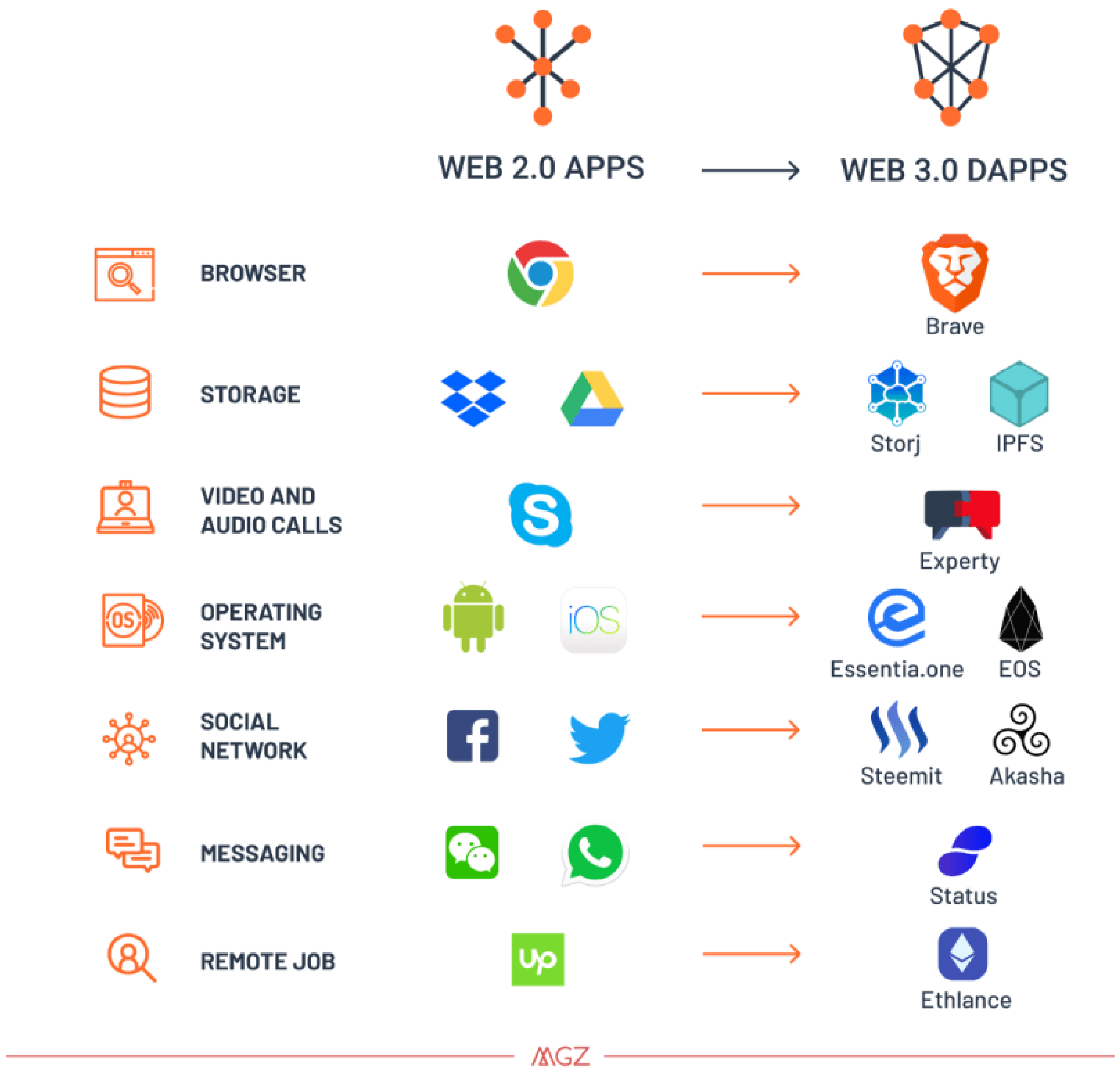
- Business applications which previously needed to be 'purchased and installed' can now, be easily accessed through your web browser.

- Essentially, more and more niche web services are emerging that can all share and access a central pool or "cloud" of data. This has largely been facilitated with web standards such as XML and in particular HR-XML for the recruitment industry.

- Web based systems can also form a solid basis to help implement and deliver new features, i.e. allowing access via mobile devices or creating web services that share and distribute data. eploy®, for example has been built right from the start as a true browser system and is already making use of many of these concepts.

- Indeed, one of the most significant implications of decentralization and blockchain technology is in the area of data ownership and compensation.

- As we move toward Web 3.0 and the technologies that support it would mature and become scalable, thus, the decentralized blockchain protocol will allow individuals to connect to an internet where they can own and be properly compensated for their time and data, eclipsing an exploitative and unjust web in which giant, centralized repositories own and profit from it.



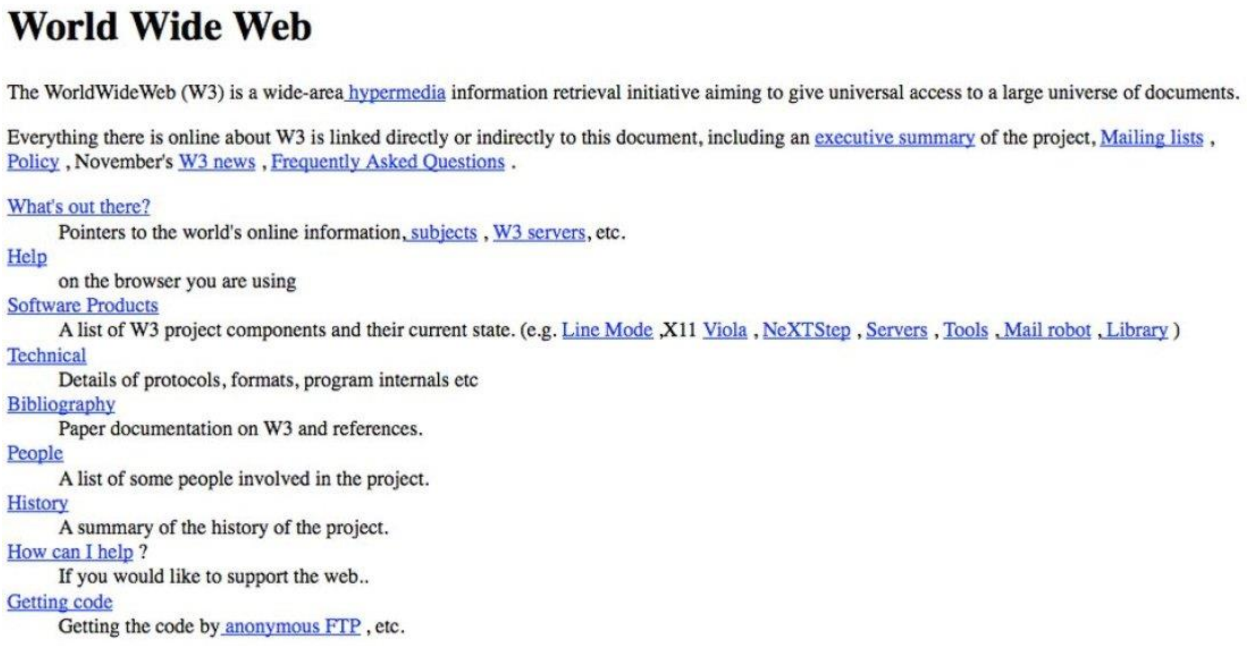
Now we'll take a look at how web files were viewed in the past :

Websites: past and present

A BRIEF HISTORY OF WEB DESIGN

- The first web page was published on 6 August 1991 by Tim Berners-Lee of the European Organization for Nuclear Research, more commonly known as CERN. The web page included information on the World Wide Web and outlined formats, protocols, and how you could create web pages.

- The address was http://info.cern.ch/hypertext/WWW/TheProject.html and in 1992 it looked like this:



World Wide Web

The WorldWideWeb (W3) is a wide-area hypermedia information retrieval initiative aiming to give universal access to a large universe of documents.

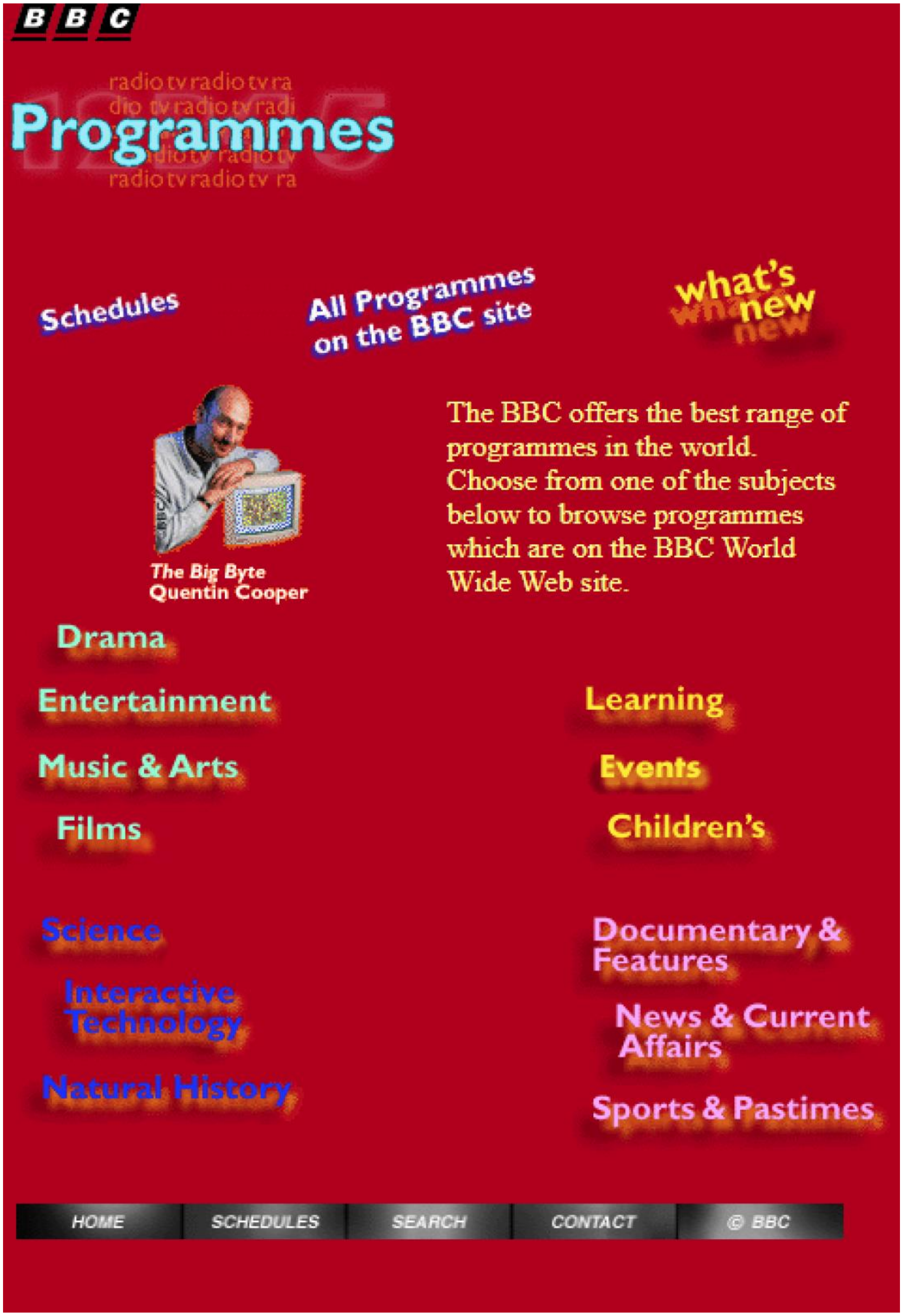
Everything there is online about W3 is linked directly or indirectly to this document, including an executive summary of the project, Mailing lists , Policy , November's W3 news , Frequently Asked Questions .

What's out there?
Pointers to the world's online information, subjects , W3 servers, etc.
Help
on the browser you are using
Software Products
A list of W3 project components and their current state. (e.g. Line Mode ,X11 Viola , NeXTStep , Servers , Tools , Mail robot , Library)
Technical
Details of protocols, formats, program internals etc
Bibliography
Paper documentation on W3 and references.
People
A list of some people involved in the project.
History
A summary of the history of the project.
How can I help ?
If you would like to support the web..
Getting code
Getting the code by anonymous FTP , etc.

- The early web pages were largely text-based, as the internet connections were slow and the technology limited. No special design or page structure was present, but the basic HTML tags for headers, paragraphs, and links made their debut.

MID-90S: TABLES AND COLORS

- Fast forward to 1996, the happy days when DVDs launched in Japan, Spice Girls conquered the world and aliens tried to conquer the world in Independence Day. Things had definitely changed since 1991, as the homepage of BBC illustrates



BBC

radio tv radio tv ra
dio tv radio tv radi
tv radio tv radio tv
radio tv radio tv ra

Programmes

Schedules

All Programmes on the BBC site

what's new

The Big Byte
Quentin Cooper

The BBC offers the best range of programmes in the world. Choose from one of the subjects below to browse programmes which are on the BBC World Wide Web site.

Drama

Entertainment

Music & Arts

Films

Learning

Events

Children's

Science

Interactive Technology

Natural History

Documentary & Features

News & Current Affairs

Sports & Pastimes

HOME | SCHEDULES | SEARCH | CONTACT | © BBC

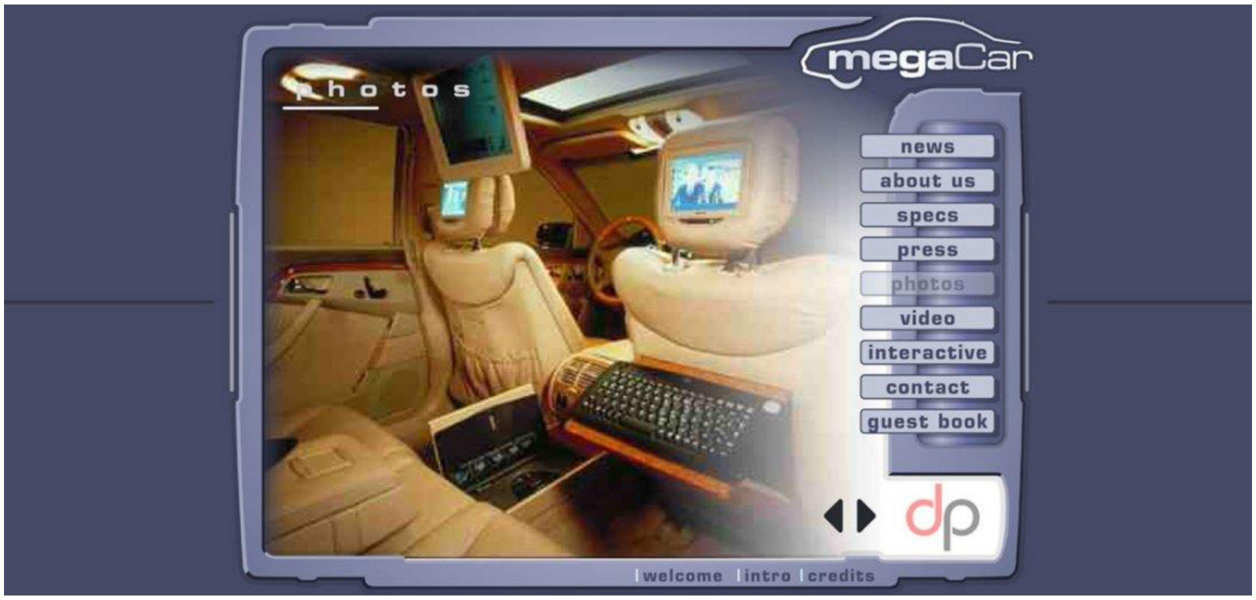
- Gone were the plain texts and layouts of the first websites, and instead we got colors, table-based layouts, flashy GIFs, and website counters. Although websites later moved away from table-based design, the idea of structure continued.

LATE 90S: THE WEB AND FLASH

- The late 90s introduced a new innovation: Flash. Growing impatient with the lack of support for animation and video on the web, Macromedia attempted to hijack the web with the introduction of Flash technology.

- While Flash isn't web, the tech nonetheless introduced web designers to an entire new world of possibilities to "augment" their table designs—including splash plages, 3D buttons, colour-changing navigation, and more, well, flashy elements.

- Kim Dotcom's data security company Megacar.com is a point in case:



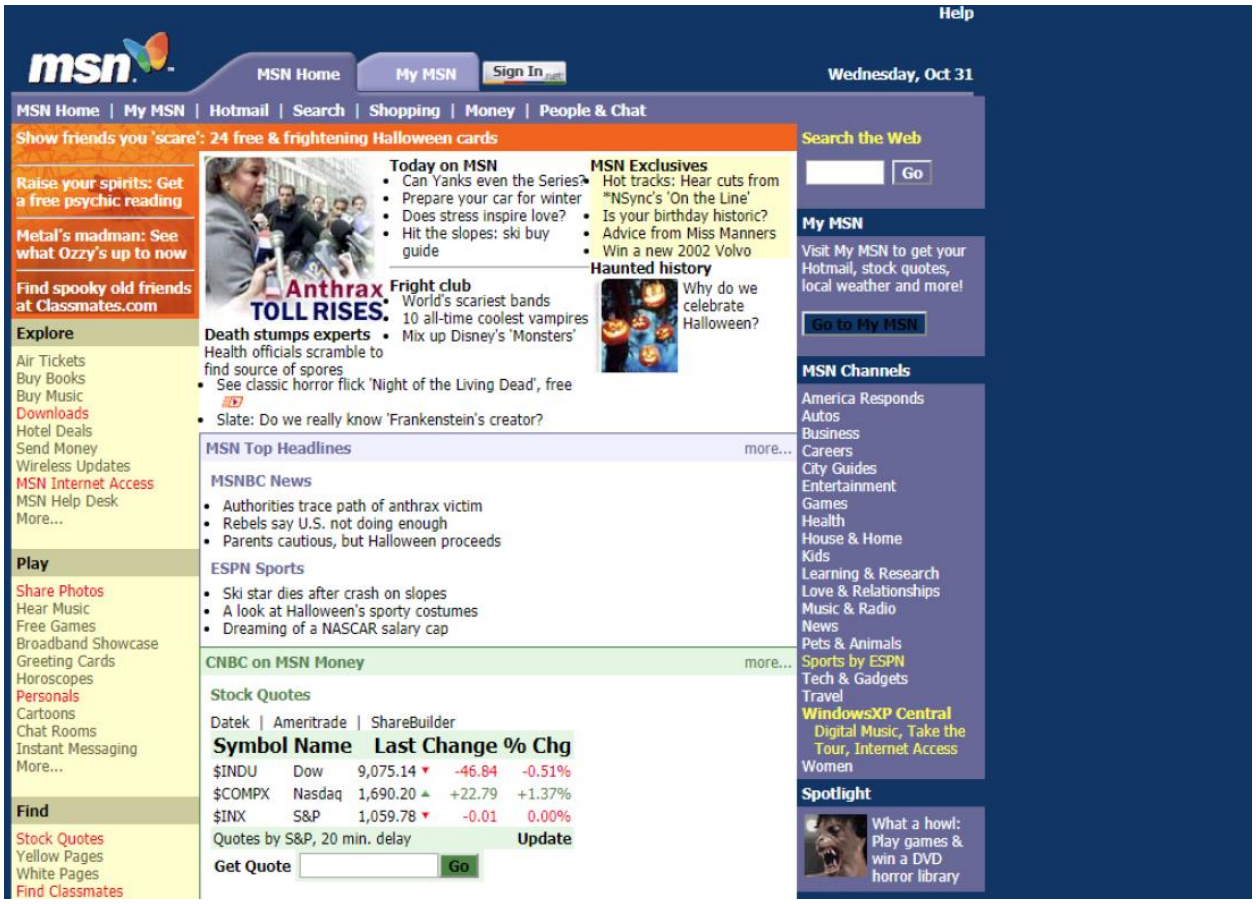
- Flash came with serious shortcomings, however, including not being searchable.

- Flash also caused slow loading times, as users had to run an application in the web browser. When the web finally acquired support for animations and video, Flash died rather quickly.

EARLY 2000S: CSS

- The early 2000s saw the influx of CSS, enabling the separation of content and design.

- Content could be created independently of design, and vice versa, making websites easier to maintain with less code and complexity.



Help

MSN Home | My MSN | Sign In

Wednesday, Oct 31

MSN Home | My MSN | Hotmail | Search | Shopping | Money | People & Chat

Show friends you 'scare': 24 free & frightening Halloween cards

Raise your spirits: Get a free psychic reading

Metal's madman: See what Ozzy's up to now

Find spooky old friends at Classmates.com

Explore

Air Tickets
Buy Books
Buy Music
Downloads
Hotel Deals
Send Money
Wireless Updates
MSN Internet Access
MSN Help Desk
More...

Play

Share Photos
Hear Music
Free Games
Broadband Showcase
Greeting Cards
Horoscopes
Personals
Cartoons
Chat Rooms
Instant Messaging
More...

Find

Stock Quotes
Yellow Pages
White Pages
Find Classmates

Anthrax TOLL RISES.

Death stumps experts
Health officials scramble to find source of spores

- See classic horror flick 'Night of the Living Dead', free
- Slate: Do we really know 'Frankenstein's creator?

Today on MSN

- Can Yanks even the Series?
- Prepare your car for winter
- Does stress inspire love?
- Hit the slopes: ski buy guide

Fright club

- World's scariest bands
- 10 all-time coolest vampires
- Mix up Disney's 'Monsters'

MSN Exclusives

- Hot tracks: Hear cuts from *NSync's 'On the Line'
- Is your birthday historic?
- Advice from Miss Manners
- Win a new 2002 Volvo

Haunted history

Why do we celebrate Halloween?

MSN Top Headlines more...

MSNBC News

- Authorities trace path of anthrax victim
- Rebels say U.S. not doing enough
- Parents cautious, but Halloween proceeds

ESPN Sports

- Ski star dies after crash on slopes
- A look at Halloween's sporty costumes
- Dreaming of a NASCAR salary cap

CNBC on MSN Money more...

Stock Quotes

Datek | Ameritrade | ShareBuilder

Symbol	Name	Last	Change	% Chg
$INDU	Dow	9,075.14 ▼	-46.84	-0.51%
$COMPX	Nasdaq	1,690.20 ▲	+22.79	+1.37%
$INX	S&P	1,059.78 ▼	-0.01	0.00%

Quotes by S&P, 20 min. delay Update

Get Quote Go

Search the Web

Go

My MSN

Visit My MSN to get your Hotmail, stock quotes, local weather and more!

Go to My MSN

MSN Channels

America Responds
Autos
Business
Careers
City Guides
Entertainment
Games
Health
House & Home
Kids
Learning & Research
Love & Relationships
Music & Radio
News
Pets & Animals
Sports by ESPN
Tech & Gadgets
Travel
WindowsXP Central
Digital Music, Take the Tour, Internet Access
Women

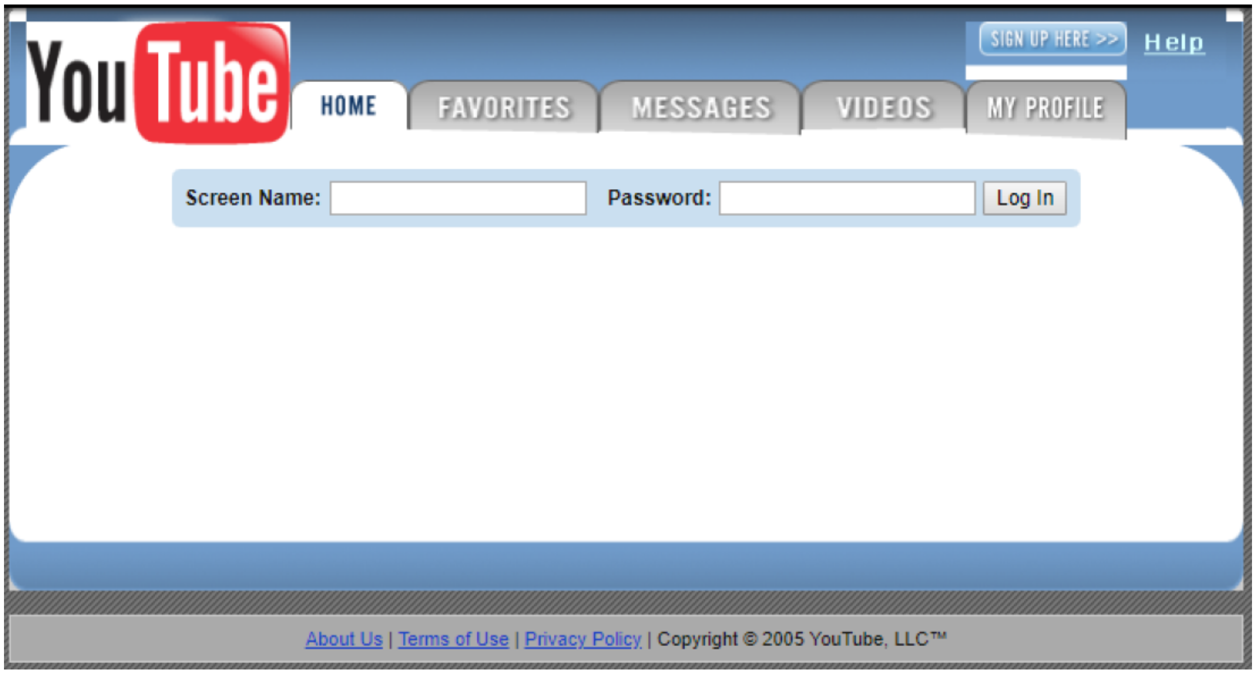
Spotlight

What a howl: Play games & win a DVD horror library

MID-2000S: WEB 2.0

- While the term “web 2.0” might be unclear, it may be used to refer to the growth of multimedia applications, interactive content, and the introduction of social media platforms. All of these features started to gain prominence from around 2005 and onwards, and marked the definitive death of Flash.

- “Web 2.0” introduced more JavaScript functionality in the web browser, as well as Ajax technology—which downloads and refreshes parts of a website. Now the web went from being static HTML to function more like the web we’re familiar with today. Greater attention were paid to typography, whitespace, color palette, user experience, and SEO. 2005 also saw the launch of YouTube:

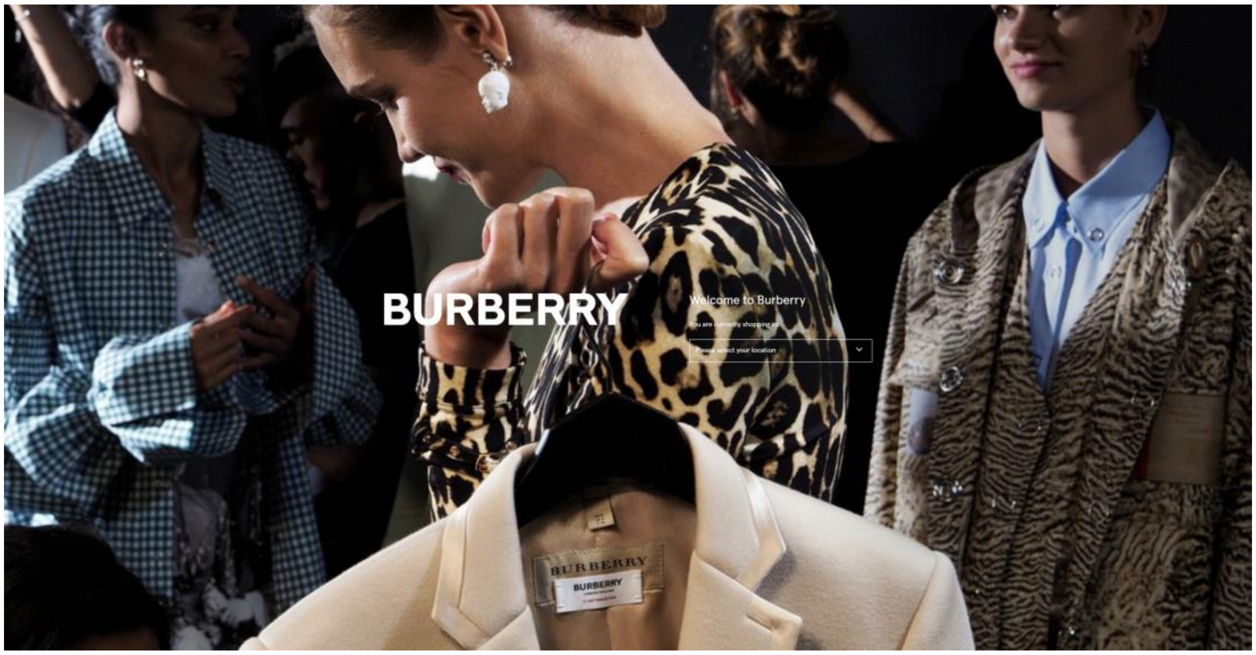


About Us | Terms of Use | Privacy Policy | Copyright © 2005 YouTube, LLC™

WEBSITES TODAY

- Further innovations in technologies like HTML5, CSS3, and JavaScript have enabled more advanced and complex websites than before—further emphasized by the addition of mobile and apps.

- Similarly, by analyzing the best manufacturing websites, one can observe how increased awareness about UX, content strategies, and SEO has influenced modern web designs, leading to minimalist graphics, flat design, blended typography, and more.



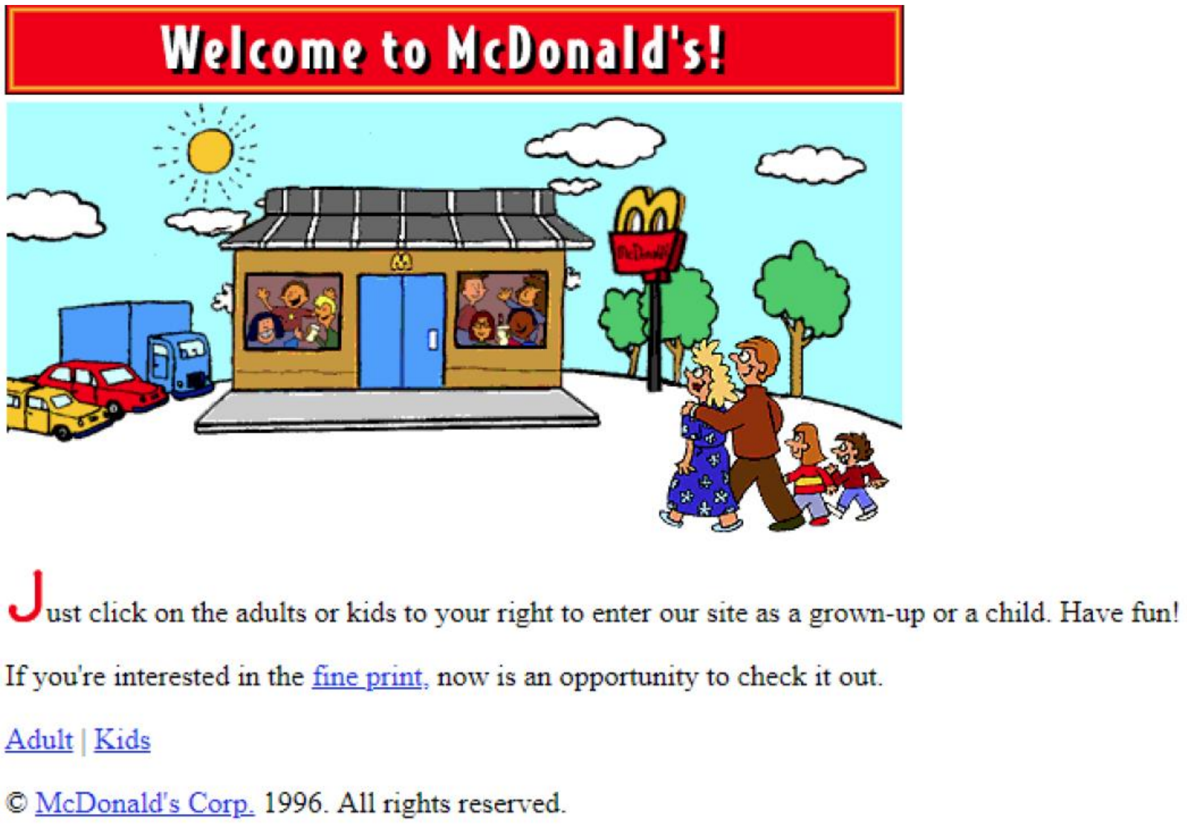
- The history of websites and web design is an oscillation between content and presentation, where content was first the dominant part, before a period where designers went bananas with Flash and tables, and finally the trend turned back to leverage content again.

- This fiery relationship is still going strong, with the addition of multichannel complexity, structured content, headless CMS, and powerful front-end frameworks like Next.js to the mix.

COMPARISONS: THEN AND NOW

- Nothing is more striking than a “before and after” comparison. To really get a grip of how much the web has evolved, let’s take a look at some famous brands and their respective websites at different times.

- McDonalds.com in 1996:



Welcome to McDonald's!

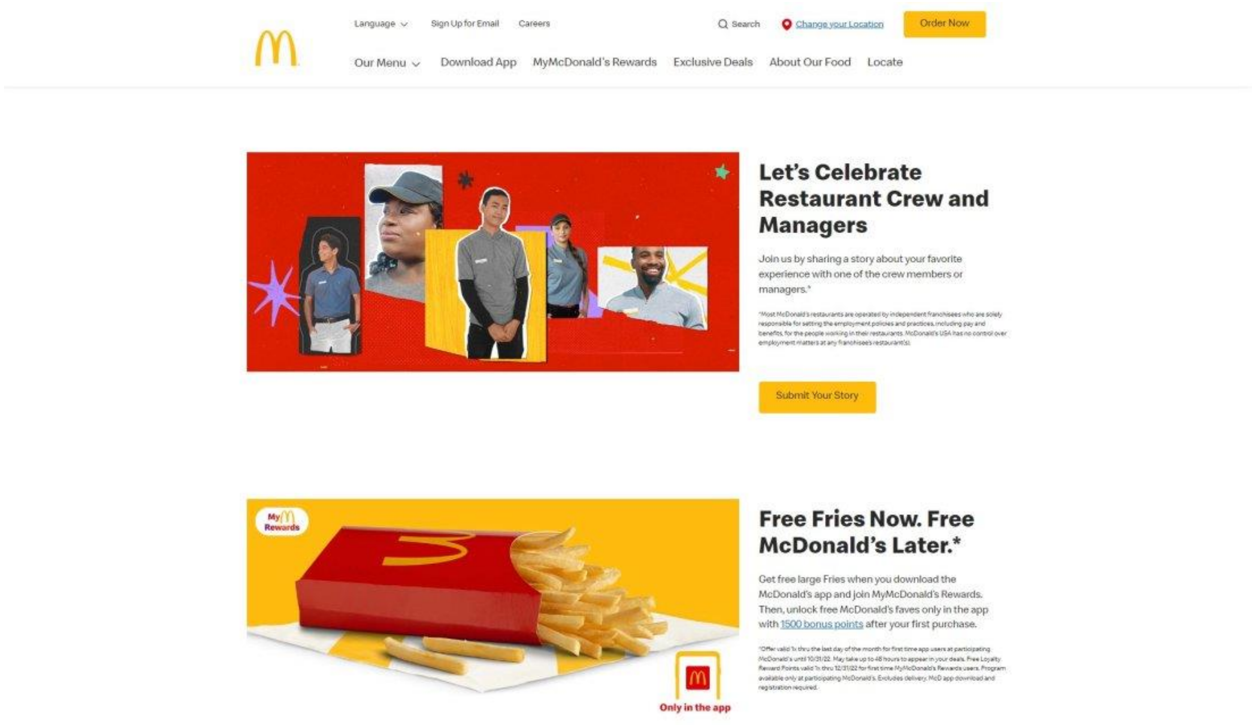
Just click on the adults or kids to your right to enter our site as a grown-up or a child. Have fun!

If you're interested in the fine print, now is an opportunity to check it out.

Adult | Kids

© McDonald's Corp. 1996. All rights reserved.

- McDonald’s in 2022:



Let’s Celebrate Restaurant Crew and Managers

Join us by sharing a story about your favorite experience with one of the crew members or managers.*

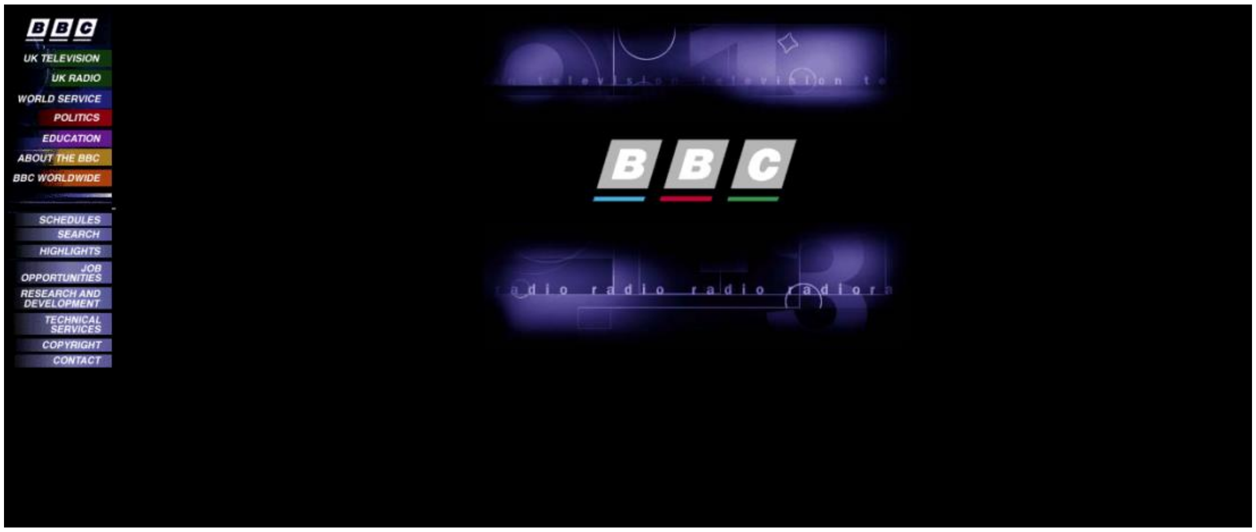
*Most McDonald's restaurants are operated by independent franchisees who are solely responsible for setting the employment policies and practices, including pay and benefits, for the people working in their restaurants. McDonald's USA has no control over employment matters at any franchisee's restaurant(s).

Free Fries Now. Free McDonald’s Later.*

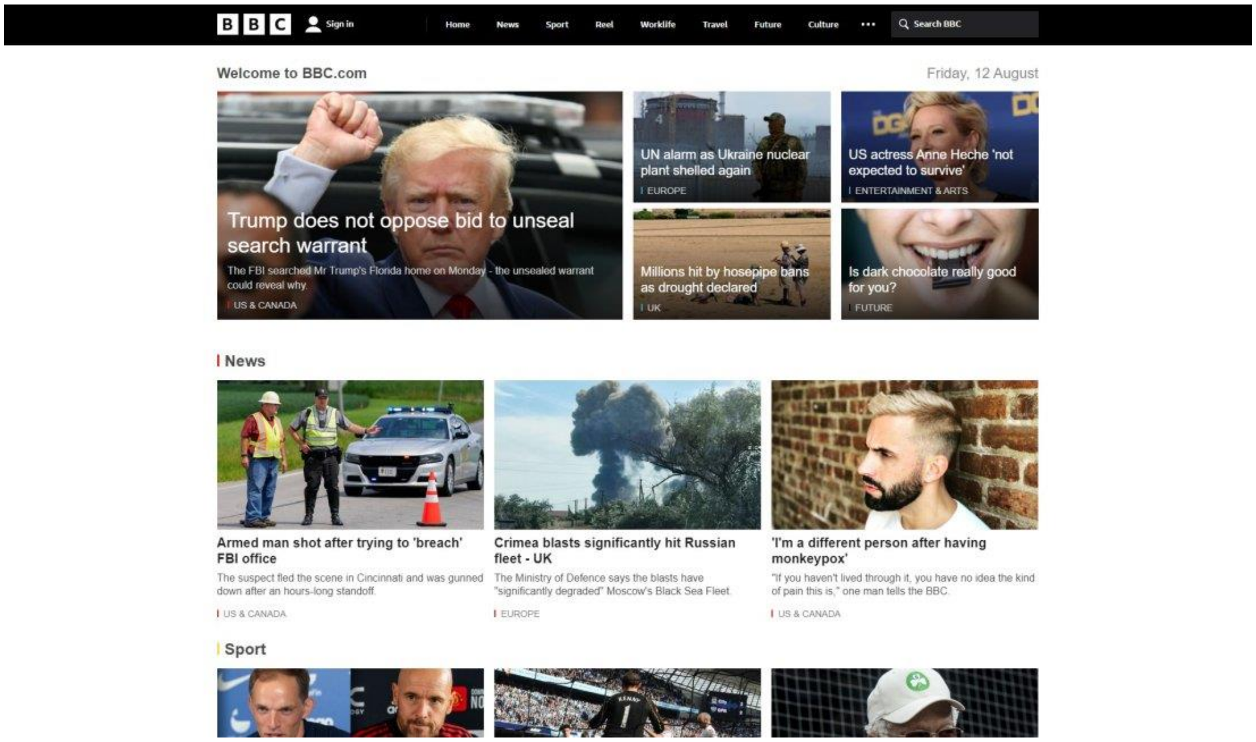
Get free large Fries when you download the McDonald's app and join MyMcDonald's Rewards. Then, unlock free McDonald's faves only in the app with 1500 bonus points after your first purchase.

*Offer valid 1x thru the last day of the month for first time app users at participating McDonald's until 10/31/22. May take up to 48 hours to appear in your deals. Free Loyalty Reward Points valid 1x thru 12/31/22 for first time MyMcDonald's Rewards users. Program available only at participating McDonald's. Excludes delivery. McD app download and registration required.

- BBC in 1997:



- BBC in 2022:



Welcome to BBC.com

Friday, 12 August

Trump does not oppose bid to unseal search warrant

The FBI searched Mr Trump's Florida home on Monday - the unsealed warrant could reveal why.

US & CANADA

UN alarm as Ukraine nuclear plant shelled again

EUROPE

US actress Anne Heche 'not expected to survive'

ENTERTAINMENT & ARTS

Millions hit by hosepipe bans as drought declared

UK

Is dark chocolate really good for you?

FUTURE

News

Armed man shot after trying to 'breach' FBI office

The suspect fled the scene in Cincinnati and was gunned down after an hours-long standoff.

US & CANADA

Crimea blasts significantly hit Russian fleet - UK

The Ministry of Defence says the blasts have "significantly degraded" Moscow's Black Sea Fleet.

EUROPE

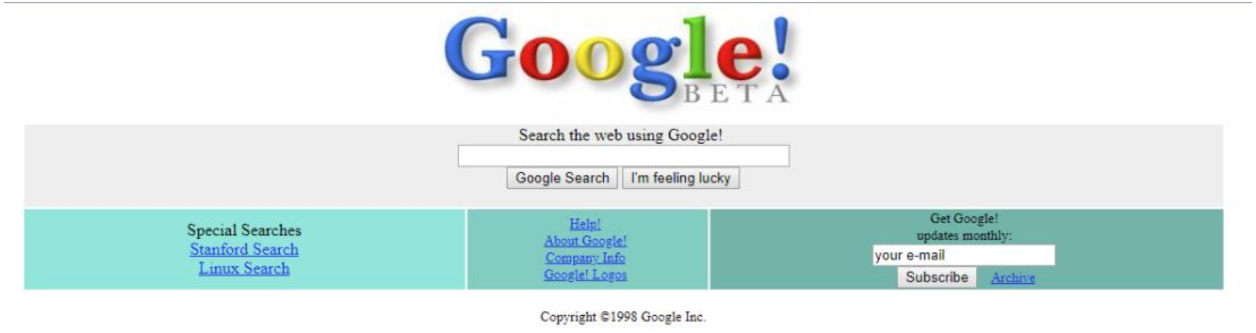
'I'm a different person after having monkeypox'

"If you haven't lived through it, you have no idea the kind of pain this is," one man tells the BBC.

US & CANADA

Sport

- Google in 1998:



Search the web using Google!

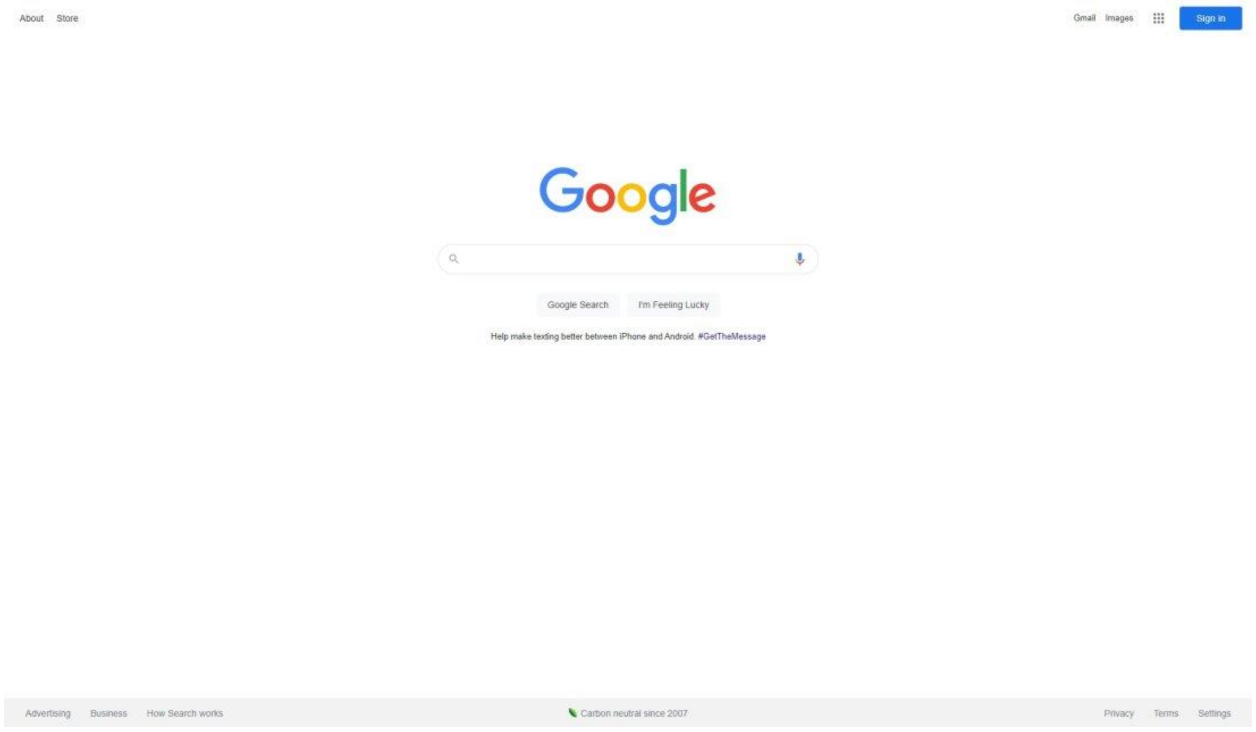
Special Searches
Stanford Search
Linux Search

Help!
About Google!
Company Info
Google! Logos

Get Google!
updates monthly:
your e-mail
Archive

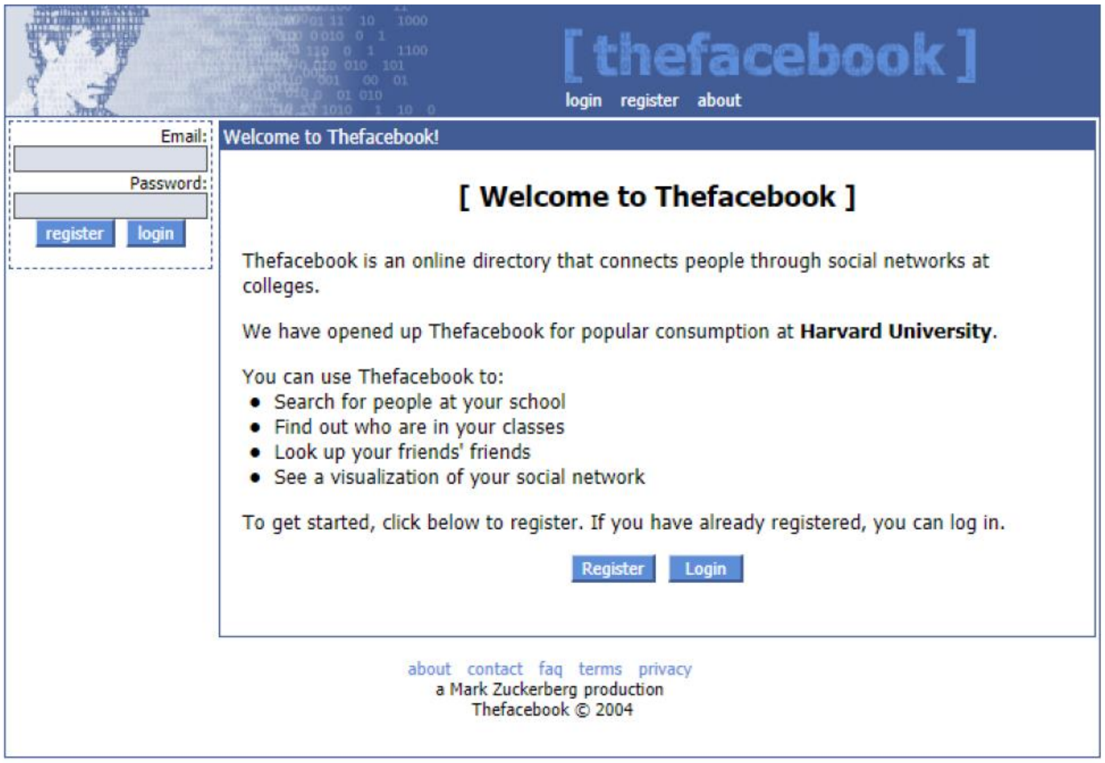
Copyright ©1998 Google Inc.

- Google in 2019:



Help make texting better between iPhone and Android. #GetTheMessage

- Facebook in 2004:



Welcome to Thefacebook!

[Welcome to Thefacebook]

Thefacebook is an online directory that connects people through social networks at colleges.

We have opened up Thefacebook for popular consumption at **Harvard University**.

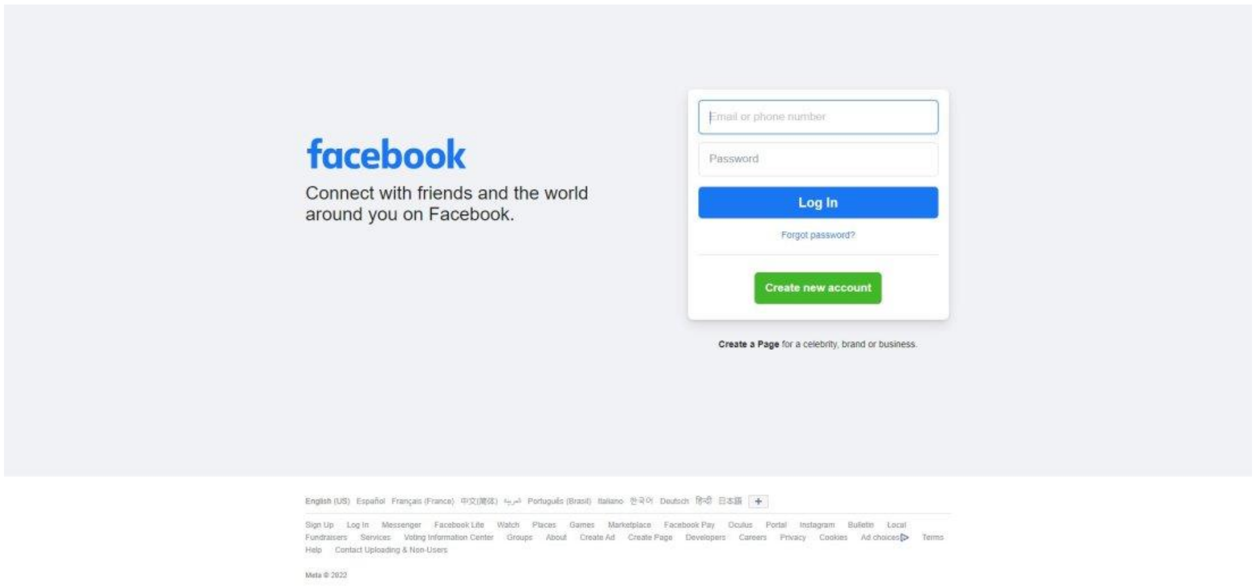
You can use Thefacebook to:
- Search for people at your school
- Find out who are in your classes
- Look up your friends' friends
- See a visualization of your social network

To get started, click below to register. If you have already registered, you can log in.

Register Login

about contact faq terms privacy
a Mark Zuckerberg production
Thefacebook © 2004

- Facebook in 2019:



facebook

Connect with friends and the world around you on Facebook.

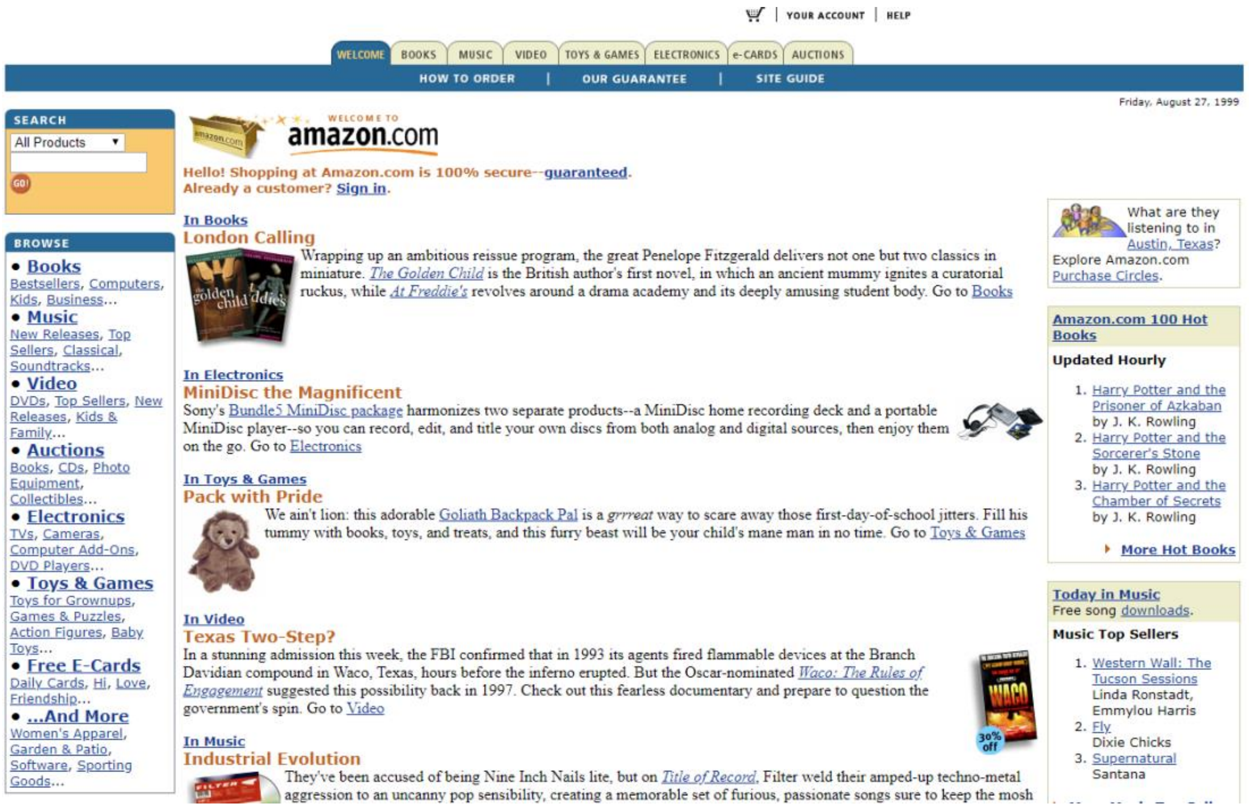
Create a Page for a celebrity, brand or business.

English (US) Español Français (France) 中文(简体) العربية Português (Brasil) Italiano 한국어 Deutsch हिन्दी 日本語 +

Sign Up Log In Messenger Facebook Lite Watch Places Games Marketplace Facebook Pay Oculus Portal Instagram Bulletin Local Fundraisers Services Voting Information Center Groups About Create Ad Create Page Developers Careers Privacy Cookies Ad choices Terms Help Contact Uploading & Non-Users

Meta © 2022

- Amazon in 1999:



Friday, August 27, 1999

Hello! Shopping at Amazon.com is 100% secure--guaranteed.
Already a customer? Sign in.

BROWSE

- **Books** Bestsellers, Computers, Kids, Business...
- **Music** New Releases, Top Sellers, Classical, Soundtracks...
- **Video** DVDs, Top Sellers, New Releases, Kids & Family...
- **Auctions** Books, CDs, Photo Equipment, Collectibles...
- **Electronics** TVs, Cameras, Computer Add-Ons, DVD Players...
- **Toys & Games** Toys for Grownups, Games & Puzzles, Action Figures, Baby Toys...
- **Free E-Cards** Daily Cards, Hi, Love, Friendship...
- **...And More** Women's Apparel, Garden & Patio, Software, Sporting Goods...

In Books
London Calling
Wrapping up an ambitious reissue program, the great Penelope Fitzgerald delivers not one but two classics in miniature. *The Golden Child* is the British author's first novel, in which an ancient mummy ignites a curatorial ruckus, while *At Freddie's* revolves around a drama academy and its deeply amusing student body. Go to Books

In Electronics
MiniDisc the Magnificent
Sony's Bundle5 MiniDisc package harmonizes two separate products--a MiniDisc home recording deck and a portable MiniDisc player--so you can record, edit, and title your own discs from both analog and digital sources, then enjoy them on the go. Go to Electronics

In Toys & Games
Pack with Pride
We ain't lion: this adorable Goliath Backpack Pal is a *grrreat* way to scare away those first-day-of-school jitters. Fill his tummy with books, toys, and treats, and this furry beast will be your child's mane man in no time. Go to Toys & Games

In Video
Texas Two-Step?
In a stunning admission this week, the FBI confirmed that in 1993 its agents fired flammable devices at the Branch Davidian compound in Waco, Texas, hours before the inferno erupted. But the Oscar-nominated *Waco: The Rules of Engagement* suggested this possibility back in 1997. Check out this fearless documentary and prepare to question the government's spin. Go to Video

30% off

In Music
Industrial Evolution
They've been accused of being Nine Inch Nails lite, but on *Title of Record*, Filter weld their amped-up techno-metal aggression to an uncanny pop sensibility, creating a memorable set of furious, passionate songs sure to keep the mosh

What are they listening to in Austin, Texas? Explore Amazon.com Purchase Circles.

Amazon.com 100 Hot Books
Updated Hourly

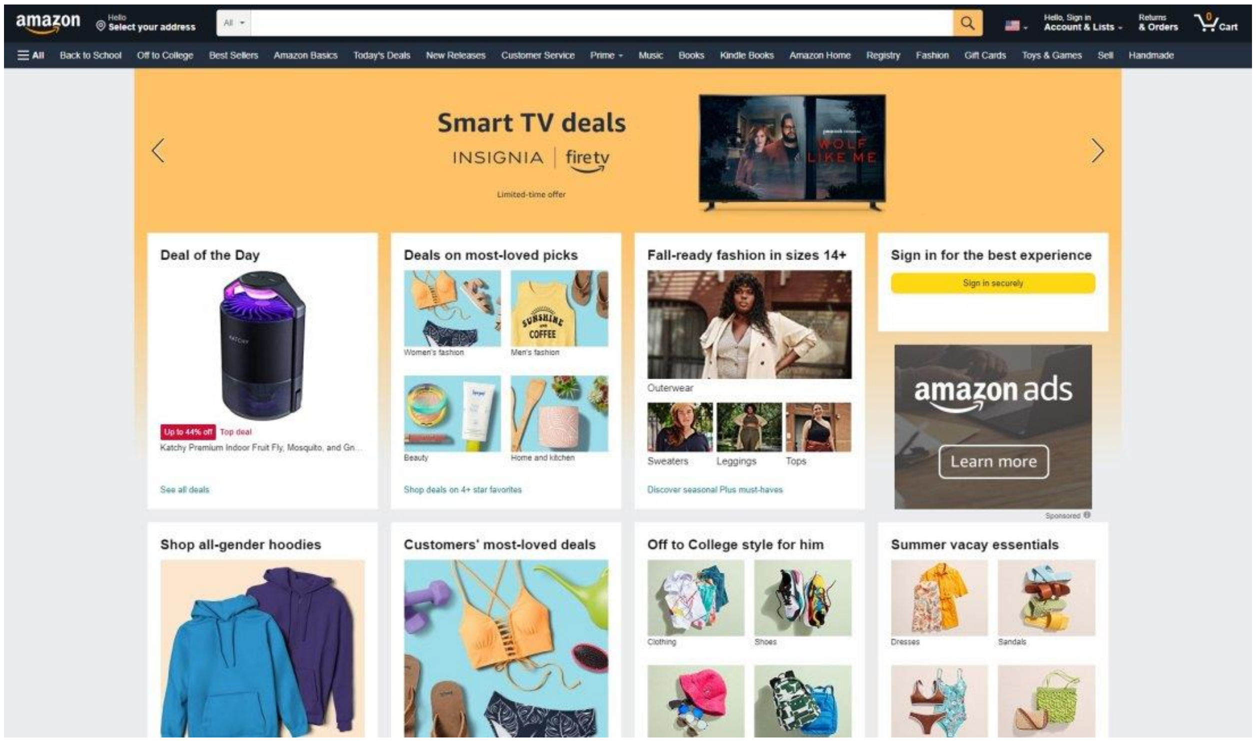
1. Harry Potter and the Prisoner of Azkaban by J. K. Rowling
2. Harry Potter and the Sorcerer's Stone by J. K. Rowling
3. Harry Potter and the Chamber of Secrets by J. K. Rowling

More Hot Books

Today in Music
Free song downloads.
Music Top Sellers

1. Western Wall: The Tucson Sessions Linda Ronstadt, Emmylou Harris
2. Fly Dixie Chicks
3. Supernatural Santana

- Amazon in 2019:



WHERE ARE WE HEADED NEXT?

The World Wide Web has changed dramatically since the first web page went online in 1991. From primitive text documents through flashy color explosions to the modern websites of today, the website has finally found a balance between content and design, and is now ready for the days to come.

Although it’s impossible to predict the future, the trend is fairly clear for where we’re headed next: The ascent of mobile, progressive web apps, omnichannel demand, and headless CMS and powerful front-end frameworks tells us a story of digital experiences with rich functionality in any channel, whether it be smart phones, tablets, wearables, IoT, digital signage, or the classical website.

We’ve had an exciting 30 years of web development—and it will be exciting to see what the next 30 years will bring.

Now let's get to know the role of React in the development of the web, but let's first know what React is:

- React JS, also known as React or React.js, is an open-source JavaScript library that serves as a front-end user interface (UI) framework for building interactive and dynamic web applications. Created and maintained by Facebook, React has gained widespread popularity among developers due to its efficiency, flexibility, and performance. It helps the developers to create reusable UI components that efficiently update and render in response to changes in data, providing a smooth user experience.

- The primary intent behind using React JS is to build responsive and high-performance web applications with a focus on a component-based architecture. Its key features include the use of a virtual DOM (Document Object Model), one-way data binding, and a declarative syntax, enabling developers to efficiently manage and manipulate the UI without directly interacting with the real DOM. This results in faster rendering and improved performance.

- React JS works by breaking the UI into reusable components. Each component encapsulates specific functionalities, allowing developers to build complex applications with ease. When data changes within a component, React efficiently updates only the affected parts of the UI, thanks to its virtual DOM. This results in minimized reflows and repaints, leading to enhanced performance.

- JavaScript plays a crucial role in enabling React JS to function seamlessly. React primarily uses JavaScript’s ECMAScript 6 (ES6) syntax, which includes modern features like arrow functions, classes, and modules. JavaScript’s powerful ecosystem provides access to numerous libraries and packages, further enhancing React’s capabilities and facilitating the development process.

Why Use React for Web Development?

- As an integral component of the JavaScript language, the utilization of React confers numerous benefits. Notably, React, along with other JavaScript technologies, has been adopted by prominent global enterprises in the development of market-leading products, with notable instances being Instagram, Reddit, and Facebook.

Improved UI Performance with React JS

- React’s efficient rendering process makes web applications faster and more responsive. It uses a Virtual DOM (Document Object Model) that creates a lightweight copy of the actual DOM. When data changes occur, React calculates the minimal changes required to update the Virtual DOM and efficiently applies those changes to the real DOM. This approach reduces unnecessary re-rendering and leads to a smoother user experience.

- For example, imagine you have a to-do list with several items. With React’s Virtual DOM, only the specific item that you modify will be updated on the screen rather than re-rendering the entire list, resulting in faster and more efficient updates.

Component-Based Architecture for Web Development

- It follows a component-based architecture, which means building web applications by combining small, self-contained components. Each component can manage its state and behavior independently, making the application easier to understand, develop, and maintain. This modular approach fosters code reusability, simplifies debugging, and promotes scalability.

Reusable UI Components with React

- As React components are self-contained, they can be reused across multiple pages or even projects. Developers can create a library of custom UI components that adhere to a consistent design language, ensuring a unified and professional look for the entire application.

Enhanced Code Maintainability using React JS

- With its declarative nature and component-based architecture, React fosters code maintainability. Declarative code allows developers to specify “what” they want to achieve, leaving React to handle “how” to accomplish it. This approach results in more readable and organized code, making it easier for the developers to collaborate and understand each other’s work.

Streamlined Collaborative Development with React

- React’s component-based structure enables effective collaboration among developers. Different team members can work on separate components independently, without interfering with others’ work. This promotes a more efficient and collaborative development process.

Cross-Platform Capabilities with React Native

- In addition to web development, React Native allows developers to build responsive mobile applications for both iOS and Android platforms using the same codebase. This cross-platform capability saves significant development time and resources, as a single codebase can be utilized for both platforms.

Seamless Integration with Other Libraries and Frameworks

- React JS can easily integrate with other libraries and frameworks, enhancing its capabilities and extending its functionality. Developers can leverage the vast ecosystem of JavaScript tools to optimize the development process and enhance the application.

Extensive React Ecosystem and Third-Party Tools for Web Development

- React JS benefits from a wide range of third-party tools and libraries available in its ecosystem. These tools provide solutions for specific challenges, such as data management, state handling, routing, and styling, among others. Developers can tap into this rich ecosystem to enhance productivity and focus on building unique features.

The Benefits of Using a Web Application Framework

- Web applications can be accessed through a web browser such as Google Chrome, Safari or Mozilla Firefox. A Web server, application server, and a database are needed to operate a web app. Usually Web servers manage the requests that come from a client, while the application server completes the requested task. Inorder to store any needed information, a database can be used. Web apps are usually written in JavaScript, HTML5, or Cascading Style Sheets which help to build an application's front-end. Server side programming is the program that runs on a server dealing with the generation of content of a web page. Languages such as Python, Java, and Ruby are commonly used in server-side programming.

- Benefits of Web apps include:

- Ease the development process

- Eases Debugging and Application Maintenance

- Improves Database Proficiency

- Code Length reduction

- Security reinforcement

Difference Between Framework Vs Library

What Is A Framework?

- The key difference between a framework and a library is that a framework's code does not contain "completed" functions. On the other hand, frameworks are program scaffolds that supply the blueprint instead of a finished product. As a result, a framework gives the fundamental structure while indicating the required customization from the coder. The framework defines the workflow of a software application, informs the developer of what he needs, and invokes the developer's code when necessary.

Components of a framework

- When examining the components of a framework like Sencha Ext JS, it's crucial to understand its key elements. First and foremost, frameworks like Ext JS offer a wide range of pre-built UI components, such as buttons, grids, forms, and charts, that can be easily integrated into applications. Moreover, these frameworks provide layout managers, which assist in organizing and positioning components within the user interface. Additionally, event-handling mechanisms enable developers to respond to user interactions and trigger specific actions.

- Furthermore, data models and stores are vital components that facilitate data management and communication with back-end servers. Another important aspect is the presence of robust styling and theming capabilities, allowing developers to customize the appearance of their applications. Lastly, frameworks often include modules for handling various tasks, such as routing, validation, and internationalization, streamlining the development process.

Why Do We Use Frameworks?

- Software development is a difficult process. This includes a lot of steps, like creating, designing, and testing. When it comes to software development projects, developers must be concerned with syntax, declarations, garbage collection, exceptions, and other considerations. Software frameworks facilitate development by offering a centralized platform from which programmers can control all or a portion of the process of software development.

Apart from that, there are other benefits to using a framework for web development:

- Reduces code length

- Enable code reuse

- Ease debugging and application monitoring

- Simplify database connectivity

- Improve security

- Although Framework can help you develop applications in many ways, you still have to follow the best practices of developing web applications to get the best out of them.

Examples Of Frameworks

Here are some of the most well-known frameworks:

- Ext JS

- Examples of framework vs library

- Sencha Ext JS is a JavaScript framework for building data-driven, cross-platform web applications. It is feature-rich and provides a robust set of tools for developers to create rich, interactive user interfaces. Sencha Ext JS also includes a comprehensive library of pre-built components, which can be easily integrated into your application. In addition, Sencha Ext JS supports both traditional web technologies and newer HTML5 standards. As a result, it is an ideal choice for developing modern web applications that need to support a wide range of devices and browsers.

- Sencha Ext JS is a powerful framework that can help you build sophisticated web applications quickly and easily.

- Learn more about Ext JS.

- Angular

- Angular is a front-end framework for developing single-page applications. It’s a lively framework capable of creating full client-side applications or any software providing generic functionality. There’s a lot to do and learn in Angular. Angular 1.x utilized JavaScript, while later editions switched to Typescript which is a superset of JavaScript.

- The main disadvantages of Angular are its size in comparison to other frameworks. While it is not SEO-friendly by default, you can optimize it for SEO. Although Google created Angular, now so many big tech companies are using it, including Microsoft and Paypal.

- Django

- Django is a Python-based framework based on the Model-View Template design pattern. It’s a free and open-source framework built targeting rapid web development. Well-known companies like YouTube, Google, and Instagram use Django to build their apps.

- Django has more than 10,000 packages or code libraries, allowing you to build virtually any functionality in your web application. These packages include APIs, database support, user authentication, CMS, and security features.

- Express

Today, Express is swiftly becoming one of the most popular top frameworks for web development. Accenture, IBM, Uber, and many other firms use it. It is also interoperable with other frameworks like Sails, Kraken, and Loopback. Express takes pride in being a lightweight, quick, and unbiased framework.

Express provides some essential framework functionality without obscuring Node's features and takes advantage of the asynchronous Node.js' robust performance. It's also highly adaptable, supporting both web apps and REST APIs. From a web developer's perspective, the biggest disadvantage of Express is there is no established manner of developing features. Finally, when you build apps with JS, you need to have a good JS test framework. Read this article to have a good guide to the JS test framework.

- Rails

- Rails is a popular Ruby-based Model-View-Controller framework liked by many developers. Airbnb, Hulu, and GitHub are some popular companies that use Rails. Rails is a novice-friendly framework that aids beginners in quickly getting started with web development. There are many valuable Rails libraries that let you add new functionalities to your application more efficiently. The Rails ecosystem is trustworthy and welcoming, and there are several courses, screencasts, and tools available to help you become a Rails expert in no time.

- Spring

- Spring is a Model-View-Controller framework written in Java. Popular companies such as TicketMaster, Wix, and BillGuard use Spring. Spring has numerous sibling projects that improve its performance and allow you to swiftly scale your business. The main con is that the learning curve of Spring can be steep, especially if you are unfamiliar with Java.

- If you are not aware, you can build feature-rich HTML5 applications using Java and GWT with Sencha GXT.

- What Is A Library?

- A library is a group of pre-written codes that make jobs easier to complete. A library is a collection of pre-defined methods and classes that developers can use to ease their work and accelerate development. As a result, developers do not need to write code to achieve specific features. Most programming languages include standard libraries, but developers can create their own customized libraries.

- Why Do We Need A Library?

- Libraries, in the context of software development, are a collection of functions readily available to you. As a software engineer, you can leverage libraries during a typical development process, regardless of whether you're working with a framework vs library. Libraries, just like frameworks, play a crucial role in increasing the efficiency of developing new applications. They provide reusable code components, predefined functions, and important constants that can be utilized to simplify and accelerate the development process. Additionally, some libraries, such as machine learning libraries, offer specialized functions and algorithms that can be used without the need for in-depth knowledge in that specific domain. Therefore, both the framework vs library are valuable resources that support developers in building applications more effectively.

- Examples Of Libraries

- React

- React is a powerful front-end library that has often been compared to frameworks due to its widespread usage and popularity. It was among the first to embrace the component-based paradigm, which has since been adopted by other frameworks like Angular and Vue. With React's virtual DOM, developers can benefit from faster DOM operations, making their applications more efficient. Notably, React was created by Facebook and has gained traction among numerous renowned companies.

- When it comes to creating web applications with React, there are many proven ways to create web applications with React. Also, React's influence extends beyond the web, as it has inspired React Native, a JavaScript framework highly regarded as one of the best JavaScript libraries for developing mobile applications.

- Redux

- Redux is a JavaScript open-source library for maintaining and standardizing application state. We can use it for designing user interfaces using libraries like React or Angular. Redux is a lightweight library with a straightforward, constrained API to serve as a trustworthy container for the application state. It works in the same way as a reduction function, which is a functional programming idea.

- Three.js

- Next, we discuss Three.js in our framework vs library guide. Three.js is a cross-browser JavaScript toolkit and application programming interface (API) that uses WebGL to produce and show animated 3D computer graphics. Furthermore, Three.js enables the production of the graphics processing unit (GPU)-accelerated 3D animations as part of a website without the use of proprietary browser plugins. This is now possible because of the introduction of WebGL, a low-level graphics API designed exclusively for the web. High-level frameworks like Three.js or GLGE, SceneJS, PhiloGL, and others enable the creation of elaborate 3D computer animations for browser display without the effort necessary for a conventional standalone application or plugin.

- Lodash

- Lodash is a JS utility library that simplifies working with integers, arrays, texts, and objects. It employs a functional programming style, and aids in the creation of manageable and concise JavaScript code. Lodash makes common chores such as math computations, throttling, decorating, debouncing, and constraining easier for developers. Moreover, it simplifies string functions, such as camel case, trimming, and upper case.

- jQuery

- jQuery is currently one of the most popular libraries among front-end developers. It is a tiny, free, open-source toolkit written in JavaScript programming language. jQuery includes many helpful features for web development, including AJAX, easy DOM manipulation, event handling, animation effects, and so on. There are many other libraries created using jQuery, such as the jQuery date picker library.

- What Are The Technical Differences Between Library vs. Framework?

- The technical difference between a framework and a library is defined by a concept known as inversion of control. When you use a library, you control the application flow, including when and where to contact the library. When you use a framework, the framework itself controls the flow.

- There are some other distinctions between libraries and frameworks as well. Let’s have a look at a couple of the most important ones:

Library Framework

- A set of assistance modules, objects, classes, functions, pre-written code, and so on.

- Includes a variety of APIs, compilers, support applications, libraries, and so on.

- Can be easily substituted by another library.   Are tough to replace.

- When we call a method from a library, we are in control.

- Inversion of control, i.e. the framework calls us.

- Since developing a library needs less code, performance and load time are improved.

- The construction of a framework necessitates large amounts of code, which reduces performance and increases load time.

- Libraries can be simply linked into existing programs to add specific functionality.

- It is tough to incorporate a framework seamlessly into an existing project.

What are Advtanges of Frameworks over Libraries?

- When considering the advantages of frameworks over libraries, several key benefits become apparent. Firstly, frameworks offer a structured and cohesive approach to development, providing predefined rules and guidelines that streamline the coding process. Also, frameworks come bundled with a set of ready-to-use components, saving developers significant time and effort in building common functionalities from scratch.

- Furthermore, frameworks typically provide a well-defined architecture that promotes code organization, maintainability, and scalability. Moreover, frameworks often incorporate tools and utilities that enhance productivity, such as debugging and testing utilities. Another advantage is the presence of a robust ecosystem around frameworks, including extensive documentation, community support, and third-party integrations. Lastly, frameworks often enforce best practices and coding standards, leading to higher code quality and improved overall application performance.

Ready to Dive Into One of the Best JavaScript Frameworks?

- Both the framework vs library is precoded support programs to develop complex software applications. However, libraries target a specific functionality, while a framework tries to provide everything required to develop a complete application. So when you develop a software application, you will need many libraries, but often one or two frameworks. Popular examples of frameworks are Ext JS, Angular, Django, Spring, and Rails, which offer a comprehensive set of tools and components for application development. On the other hand, popular examples of libraries are React and jQuery, which focus on specific tasks or functionalities and can be used in conjunction with frameworks. The choice between a framework vs library depends on the specific requirements and scope of the project at hand.

Unveiling the Internals of ReactJS: A Deep Dive into its Inner Workings

1. The Reconciliation Process

1.1. Virtual DOM Overview

At the heart of React's efficiency lies its Virtual DOM. This virtual representation of the actual DOM allows React to compute and apply updates more optimally. We'll explore how the Virtual DOM is constructed and why it plays a crucial role in React's performance.

1.2. Reconciliation Algorithm

When a component's state changes, React employs a process called reconciliation to determine the minimum number of updates required to sync the Virtual DOM with the real DOM. We'll delve into the details of how React performs this diffing algorithm to optimize rendering.

2. Component Lifecycle and Rendering

2.1. Component Initialization

When a component is created, React goes through a series of lifecycle methods, including constructor, render, and componentDidMount. We'll examine the order in which these methods are invoked and their respective purposes.

2.2. Updating Lifecycle

As components receive new props or state changes, they go through an update lifecycle. This involves methods such as shouldComponentUpdate, componentDidUpdate, and getDerivedStateFromProps. We'll explore how these methods contribute to efficient updates and how React determines whether a component should re-render.

3. Fiber Architecture

3.1. Introduction to Fiber

React’s Fiber architecture is a reimagining of its internal algorithm for handling component updates. It enables more granular control over the rendering process, improving performance and allowing for better interruption and resumption of rendering tasks.

3.2. Concurrent Mode

Concurrent Mode, built on top of the Fiber architecture, introduces the concept of scheduling rendering tasks based on priority. This paves the way for more responsive user interfaces and smoother interactions.

4. Hooks and their Implementation

4.1. State Hooks: useState

We’ll explore how React implements the useState hook internally to manage component state and provide a way for functional components to hold local state.

4.2. Effect Hooks: useEffect

The useEffect hook facilitates side effects like data fetching and DOM manipulation. We’ll look into how React schedules and manages these effects to avoid unnecessary computations.

5. Context and Reusable State

5.1. Context API Under the Hood

The Context API enables components to share state without the need for prop drilling. We’ll dissect how React manages context behind the scenes and maintains its efficient updating mechanisms.

6. Performance Optimization Strategies

6.1. Memoization with React.memo

React.memo is used to prevent unnecessary re-rendering of components. We’ll examine how this mechanism works and its implications for performance.

6.2. Memoization with useMemo

The useMemo hook is employed to memoize expensive computations. We’ll delve into how React caches values and optimizes re-computations using this hook.

What is the difference between server-side and client-side?

WHAT IS SERVER-SIDE?

- Server-side refers to processes that are carried out on the web server, where the website or web application is hosted. These processes are typically executed by the server before the website or web application is delivered to the user's device, and they can include tasks such as retrieving data from a database, rendering a web page, or handling user input.

WHAT IS CLIENT-SIDE?

- Client-side, on the other hand, refers to processes that are carried out on the user's device, typically in the user's web browser. These processes are executed after the website or web application has been delivered to the user's device, and they can include tasks such as rendering and displaying a web page, handling user interactions, or running JavaScript code.

WHAT ARE THE DIFFERENCES?

- One of the main differences between server-side and client-side processes is the amount of control and access to resources that each has. Server-side processes have access to the server's resources, such as its CPU, memory, and storage, as well as any databases or other servers that the web application uses. Client-side processes, on the other hand, have access only to the resources of the user's device, such as its CPU, memory, and storage.

- Another difference between server-side and client-side processes is the level of trust that can be placed in them. Because server-side processes are executed on the web server, they are typically more secure and less vulnerable to tampering or malicious attacks. Client-side processes, on the other hand, are executed on the user's device, which means that they are potentially less secure and more susceptible to tampering or attacks.

In summary, the key benefits of client-side rendering technologies include:

- Reduced server-side workload.

- Improved separation of concerns.

- Reduced server-side costs.

- Easier code deployments.

- A better client experience.

What Is Server-side Rendering And How Does It Improve Site Speed?

- Server-side rendering (SSR) addresses the performance and search engine optimization issues of single-page JavaScript applications. In contrast to client-side rendering, it generates static content on the server before sending it over to the user's browser.

An overview of single-page applications

Single-page applications (SPAs) are a web app architecture that appeared as an alternative to traditional websites and multi-page applications. SPAs, also known as client-side apps, became possible with the introduction of asynchronous JavaScript (AJAX), which makes it possible to update smaller parts of the user interface without reloading the full page.

Modern-day SPAs are often built with frontend UI frameworks such as React, Vue, and Angular. They consist of reusable JavaScript components fully rendered on the client side.

The main goal of this architecture is to make web apps similar to native mobile and desktop applications in terms of interactivity. As SPAs only have a single HTML page that fetches data from the server asynchronously, users can see updates instantly, without having to wait for the whole page to refresh.

How does client-side rendering work?

Client-side rendering (CSR) is the default rendering method for single-page applications.

In web development, rendering means the process of converting application code into interactive web pages. The page HTML is generated by a JavaScript engine. With client-side rendering, this is always done on the frontend. The browser then takes the generated HTML to visually render the page.

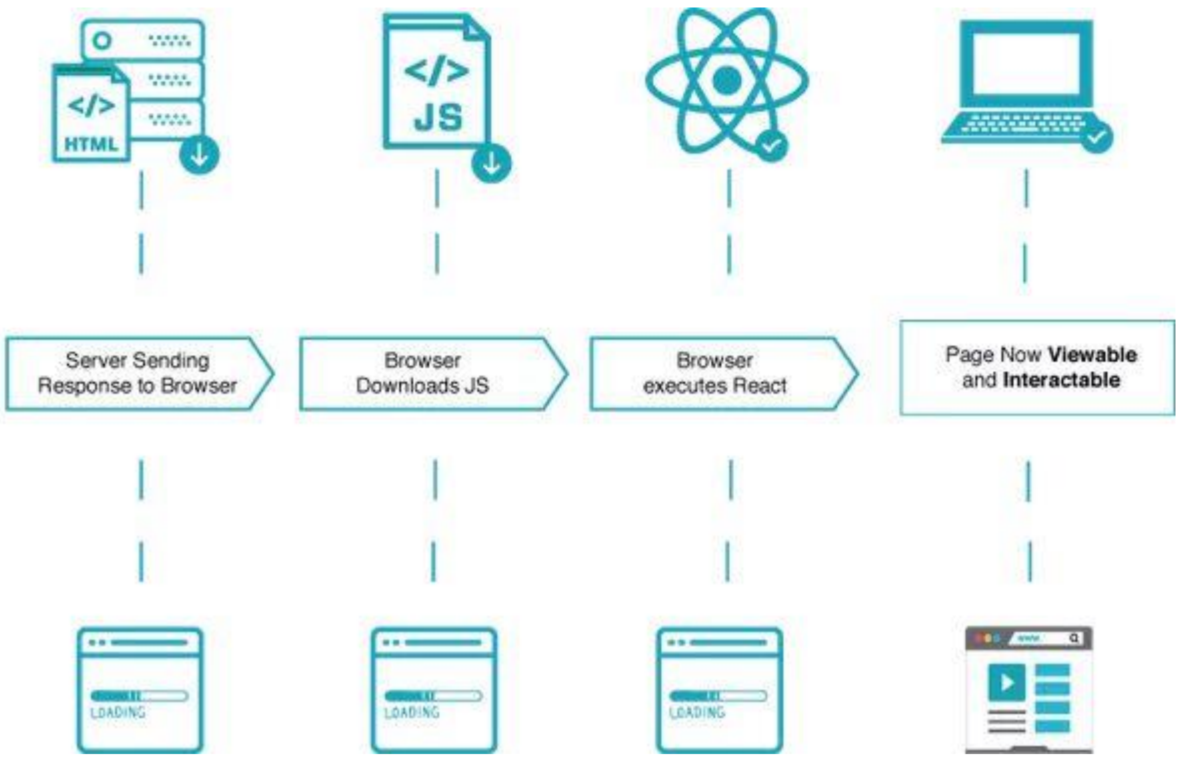
If you use client-side rendering, it's the user's browser that generates the entire app, including the user interface (UI), data, and functionality. No server is involved in the process, except to store the client-side code and data and transfer it to the browser.

As the following code example shows, in CSR apps, the HTML file only contains a blank root (often also named app) element and a script tag. The root element is populated by the browser that downloads and processes the JavaScript bundle to render all the other elements:

```
<!DOCTYPE html>

<html lang="en">

  <head>

    <meta charset="UTF-8">

    <title>CSR</title>

  </head>

  <body>

    <div id="root"><!-- blank --></div>

    <script src="/bundle.js"></script>

  </body>

</html>
```

Since the browser needs to download and run the whole application code before the content appears on the screen, the first page load is usually slow with client-side rendering (server-side rendering splits this process between the client and server).

As a result, users see a blank screen or loading spinner for a relatively long time. This leads to a poorer user experience and higher bounce rates (see Google’s discussion of how page load time impacts bounce rates).



What is server-side rendering (SSR)?

Server-side rendering, also known as universal or isomorphic rendering, is an alternative rendering method for single-page applications. SSR generates the static HTML markup on the server so that the browser gets a fully rendered HTML page. This is done by using a backend runtime such as Node.js that can run the JavaScript code to build the UI components.

Here's an example HTML file, containing a simple newsletter signup form, that the browser could receive with server-side rendering. All HTML elements inside the root element were rendered on the server:

```
<!DOCTYPE html>

<html lang="en">

  <head>

    <meta charset="UTF-8">

    <title>SSR</title>
  </head>

  <body>

    <div id="root">

      <div class="container">

        <h2>Stay Updated</h2>

        <form method="post">

          <input type="email" name="email"

          placeholder="Enter your email" required>

          <button type="submit">Subscribe</button>

        </form>

      </div>

    </div>

    <script src="/bundle.js"></script>

  </body>

</html>
```

As the browser doesn’t have to render the HTML, static content appears on the page faster with server-side rendering. However, the browser still needs to download and process the JavaScript file to add interactivity to the HTML elements. As a result, users will need to wait more before they can interact with the app, e.g. click buttons or fill input fields.

Faster-loading static content versus less time between content visibility and interactivity is a trade-off between server-side and client-side rendering — but more on that later.

Steps in the server-side rendering process

An SSR app processes the same JavaScript code on both the client and server side — this is why it’s also called universal rendering.

In brief, server-side rendering consists of the following steps:

Client's HTTP request – When the user enters the URL into the browser's address bar, it establishes an HTTP connection with the server, then sends the server a request for the HTML document.

Data fetching – The server fetches any required data from the database or third-party APIs.

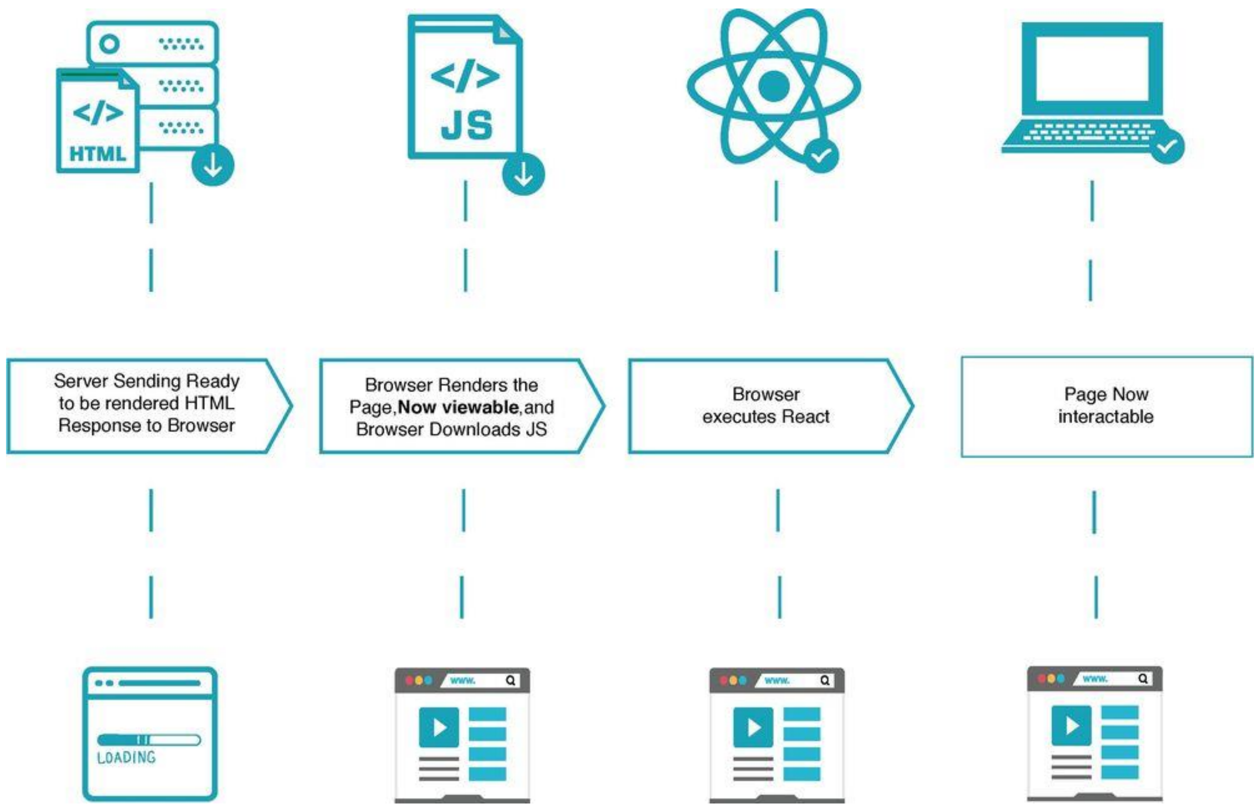
Server-side pre-rendering – The server compiles the JavaScript components into static HTML.

Server's HTTP response – The server sends this HTML document to the client.

Page load and rendering– The client downloads the HTML file and displays the static components on the page.

Hydration – The client downloads the JavaScript file(s) embedded into the HTML, processes the code, and attaches event listeners to the components. This process is also called hydration or rehydration.

Here's how server-side rendering looks from the browser's perspective. Note that the flowchart below starts with Step 4 when the browser gets the server's response:



HTML
JS
Server Sending Ready to be rendered HTML Response to Browser
Browser Renders the Page, Now viewable, and Browser Downloads JS
Browser executes React
Page Now interactable
LOADING

Server-side rendering frameworks and tools

Popular frontend UI frameworks all have their own features that make it possible to write universal JavaScript code that also runs on the server side, respectively:

React uses the ReactDomServer object together with the hydrateRoot() method.

Vue has a createSSRApp() method and a corresponding server-side rendering API.

Angular has its in-house server-side rendering tool called Angular Universal.

Processing server-side JavaScript also needs a backend JavaScript framework that runs on the Node.js server, such as Express.js or Hapi. These backend frameworks handle network requests, render the components on the server, and return the pre-rendered HTML to the browser. You can use them together with any frontend JavaScript framework.

There are also full-stack JavaScript frameworks for creating universal applications, such as Next.js for React or Nuxt.js and Quasar for Vue.

What are the advantages of server-side rendering?

Server-side rendering can make your website load more quickly and make it easier for search engines to index.

How big the positive impact will be depends heavily on how your website is built. Use a site speed testing tool to check if server side rendering is a good way to speed up your website.

Better search engine indexability

These days, search engine bots can easily crawl static HTML, but they still tend to have problems with indexing JavaScript-generated content. Even though Google can now index synchronous JavaScript, JavaScript SEO is a complicated question with several drawbacks such as delays in JavaScript indexing.

As a result, client-side rendering is still considered risky from an SEO perspective. Put simply, if you want to rank high in search engines, server-side rendering is the better choice.

Faster initial page loads

As SSR apps pre-render HTML on the server, it takes less time for the browser to load content onto the screen.

However, note that while first-time visitors do experience faster initial page loads with server-side rendering, caching might change this result for returning users. If the frontend page doesn't load any dynamic data from the server and all code is already cached, the browser only needs to render the page locally with client-side rendering.

Faster Largest Contentful Paint (LCP)

Largest Contentful Paint is one of Google's three Core Web Vitals now included in its search ranking algorithm. It's also the one that's the hardest to pass for both desktop and mobile search.

LCP is a time-based value measured in seconds. A lower value means a better LCP score. As the largest content element (either an image or text block) is part of the static content your server pre-renders, SSR will display it faster on the screen.

Lower Cumulative Layout Shift (CLS)

Cumulative Layout Shift is another Core Web Vitals score tracked by Google. It measures the amount of unexpected change in the dimension and position of your content elements after the first page render.

With server-side rendering, the browser doesn't have to go over the rendering process step by step, which typically results in fewer random layout shifts and, therefore, better CLS scores.

Fewer issues with social media indexing

Similar to search engine bots, social media crawlers also have issues with indexing JavaScript content. For example, Facebook's Open Graph Protocol and Twitter Cards don't support client-side rendering. So, if social media is important for your marketing strategy, server-side rendering can be the better choice.

Better for accessibility

As the server sends pre-rendered content to the browser, SSR apps are more suitable for people who use older devices with less powerful CPUs.

Server-side rendering is also a frequent recommendation for SPA accessibility as assistive technologies such as screen readers can’t always parse client-side JavaScript.

Are there disadvantages to server side rendering?

Despite its numerous advantages, there are still some cases when SSR might not be worth the effort. It can increase implementation and hosting costs, and in some cases leads to a worse user experience if not implemented carefully.

Increased complexity

SSR increases complexity, which may or may not be worth it for you. You’ll have to write universal code that runs both on the server and client, take care of more complicated dependency management and caching, set up and maintain a server environment, find developers with the proper skillset, and more.

Obviously, this more complex architecture will be more expensive, harder to maintain and debug, and more prone to errors.

Potentially higher First Input Delay (FID)

First Input Delay is the third metric of Google’s Core Web Vitals. It’s also the one where server-side rendering might lead to web performance issues. FID is a time-based value measured in milliseconds. It shows how long it takes for the browser to respond to the user’s first interaction.

With server-side rendering, the browser displays static content faster (which leads to a better LCP), but it still needs time to hydrate the application. As a result, the app looks ready for interaction while the code is still being processed in the background. If the user tries to interact with the app during this period of time, there will be a delay in the browser’s response.

The extent of the first input delay depends on many things, including your app's complexity, whether there are many interactive elements, the page weight, and others. For many SSR apps, first input delay won't be an issue.

However, if you experience higher FID, you still don't have to give up on server-side rendering, as there are ways to mitigate it.

For example, your UI can indicate to users that the app is not yet ready for input (e.g. you can hide or disable the buttons) so that they won't try to interact with it too early and, therefore, produce a high FID score. Alternatively, you can eliminate long-running blocking tasks by splitting up rendering into smaller chunks.

Less efficient caching

With client-side rendering, you can speed up your app by taking full advantage of browser caching. The initial page HTML is the same for all pages, so you can cache it and load it from a content delivery network (CDN) along with the JavaScript code.

With server-side rendering, the page HTML is different for each page, so it's harder to cache this on a CDN. Users who load a page that hasn't been cached on the CDN will experience a longer page load time.

Compatibility issues

There are several third-party libraries and tools that are not compatible with server-side rendering.

For example, at DebugBear, we recently started implementing server-side rendering for some of our components. Our frontend is written in TypeScript and imports CSS code for each UI component, which is then compiled by Webpack and served as a single JavaScript file.

However, on the backend, we use the standard TypeScript compiler rather than Webpack, so we had to switch from SCSS includes to Emotion, a CSS-in-JS library, to render these components on the server.

Even though nowadays compatibility issues are less of a problem than used to be, you still need to choose your dependencies carefully if you want to use server-side rendering.

Higher costs

As client-side apps don’t need a server, you can deploy them to a free or cheap static storage service such as Netlifly or Amazon S3. However, you’ll need to pay for a server or at least a “serverless” backend to deploy an SSR application, which means higher running costs.

Larger HTML size

SSR apps come with a larger HTML size because of the embedded hydration state.

This is not really an argument against SSR, just something to keep in mind as a potential risk if it’s implemented poorly. You can test your app for HTML bloat and other issues with our free HTML size analyzer tool.

React's Rendering Mechanism: Explaining the concept of rendering in React

React's process of describing a user interface based on the application's current state and props.

What is the DOM?

The Document Object Model (DOM) is a programming interface for web documents. It represents the page so that programs can change the document structure, style, and content. The DOM represents the document as nodes and objects; that way, programming languages can interact with the page.

A web page is a document that can be either displayed in the browser window or as the HTML source. In both cases, it is the same document but the Document Object Model (DOM) representation allows it to be manipulated. As an object-oriented representation of the web page, it can be modified with a scripting language such as JavaScript.

For example, the DOM specifies that the querySelectorAll method in this code snippet must return a list of all the <p> elements in the document:

JS

Copy to Clipboard

```
const paragraphs = document.querySelectorAll("p");

// paragraphs[0] is the first <p> element

// paragraphs[1] is the second <p> element, etc.

alert(paragraphs[0].nodeName);
```

All of the properties, methods, and events available for manipulating and creating web pages are organized into objects. For example, the document object that represents the document itself, any table objects that implement the HTMLTableElement DOM interface for accessing HTML tables, and so forth, are all objects.

The DOM is built using multiple APIs that work together. The core DOM defines the entities describing any document and the objects within it. This is expanded upon as needed by other APIs that add new features and capabilities to the DOM. For example, the HTML DOM API adds support for representing HTML documents to the core DOM, and the SVG API adds support for representing SVG documents.

The Virtual DOM: Understanding the concept of the virtual DOM in React

React uses Virtual DOM exists which is like a lightweight copy of the actual DOM(a virtual representation of the DOM)

Pros and Cons of Virtual DOM

- It is clear that the performance provided by the Virtual DOM is amazing. Not only that, below are some advantages of the Virtual DOM:

Speed and performance boost

- Lightweight

- It is simple and clear

- Amazing diffing algorithm

- It can be used on other frameworks not just react

- Well, anything which has an advantage will have a disadvantage too, let us consider the drawbacks of virtual DOM:

- Higher memory usage problems as the diffing algorithms need to keep comparing the elements to know which components needs to be updated or changed.

- It is not easily integrated into many other frameworks.

- You can’t use it or target template engines.

- Even at the cons mentioned above, virtual DOM is always the go-to because of the boost in performance and speed it offers.

Performance Optimization Techniques

React's Virtual DOM provides a more efficient way to update the real DOM, but there are additional optimization techniques that can be employed to further improve your application's performance. Some of these techniques include:

PureComponent and shouldComponentUpdate

PureComponent:

React.PureComponent is a base class for class components that automatically implements the shouldComponentUpdate lifecycle method. It performs a shallow comparison of the component's props and state, skipping the re-render if there are no changes. This can improve performance by preventing unnecessary re-renders.

shouldComponentUpdate:

You can manually implement the shouldComponentUpdate lifecycle method in a class component to determine whether a re-render is necessary. By providing a custom comparison logic, you can prevent unnecessary re-renders and optimize your component's performance.

useMemo and useCallback Hooks

useMemo:

The useMemo hook can be used in functional components to memoize expensive calculations, ensuring that they are only recomputed when necessary. By providing an array of dependencies, useMemo will only recompute the value when one of the dependencies changes.

```
const expensiveValue = useMemo(() => {
  // Perform expensive calculation here
}, [dependency1, dependency2]);
```

useCallback:

The useCallback hook is useful for memoizing functions in functional components, preventing unnecessary re-renders of child components that depend on the function as a prop. Like useMemo, useCallback accepts an array of dependencies to determine when the function should be recreated.

```
const memoizedCallback = useCallback(() => {
  // Your callback function here
}, [dependency1, dependency2]);
```

React.memo for Functional Components

React.memo is a higher-order component that can be used to optimize the rendering behavior of functional components. It performs a shallow comparison of the component's props, similar to PureComponent, and skips the re-render if there are no changes. This can help prevent unnecessary re-renders and improve performance.

```
const MyComponent = React.memo(function MyComponent(props) {
  // Your component implementation here
});
```

Using Keys for List Elements

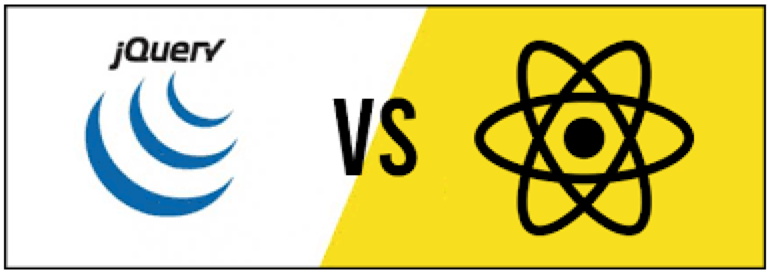
When rendering lists in React, it is important to provide a unique key for each item in the list. This helps React to identify which items have changed, been added, or removed, enabling efficient updates and preventing unnecessary re-renders.

```
{items.map((item) => (
  <ListItem key={item.id} data={item} />
```

))}

By applying these performance optimization techniques in combination with React's Virtual DOM, you can build highly efficient and performant web applications that deliver a seamless user experience.

React vs. jQuery: Highlighting the differences between React and jQuery



JQuery: Main Characteristics, Pros And Cons

[One of the most popular programmer tools is the small but feature-rich jQuery (jQ) library written in JavaScript (JS). Simply put, jQuery is a set of useful functions that make it easy to manipulate HTML elements and CSS styles, create AJAX requests, handle events (for example, mouse clicks), control animation – in short, everything that is called "interactivity".

What are the undoubted advantages of jQuery?

Cross-browser compatibility. The jQ syntax is supported by all modern web browsers, and you can be sure that the stylish slider you create in jQuery and CSS will be seen by every site visitor, no matter what browser they use.

Code Compactness. What you have to write in JavaScript in separate functions is implemented in jQ literally in a couple of code lines. Moreover, such code has harmony, logic, and clarity, so programming in jQuery is a pleasure.

A convenient work with events and visual effects. Need to create a tooltip on mouseover? Start animation on click? Create the effect of falling snow right after page load? Request the user's data? All this can be easily implemented in jQuery.

Clear documentation. On the official resource, you can get acquainted not only with the library's capabilities, but also with all its functions, conveniently divided into categories. Read and implement.

The thousands of ready-made plugins are perhaps the main advantage. Don't want to create a photo gallery from scratch? There are tons of ready-made options on the net! Need a tooltip? Slide show? HTML form validation? jQuery does it all, you just need to find a plugin.

Let's be objective and try to highlight several disadvantages of the jQ library:

Performance. Experts claim that code in other frameworks (such as React) is faster than jQuery. But there is a problem – today it is almost impossible to verify this thesis, which is associated, firstly, with an increase in the speed of the library itself, and secondly, with an increase in the performance of computers through which users access the network. Moreover, the speed of the Internet is increasing every year.

Library size. jQuery weighs about 19 KB, which theoretically can affect the speed of loading a web page, especially on older computers and ones with an unstable network connection. However, here we return to the above-mentioned thing – most modern providers guarantee an acceptable Internet speed, so the overwhelming majority of users will not see any problems when loading a page with jQuery code.

React: Differences And Advantages

The quality of user interfaces directly affects the success of an application. Therefore, front-end development is one of the most dynamically developing areas of programming. It is difficult to keep up with all the many new trends and tools in this area. With the adoption of the ES6 standard, developers are beginning to move away from imperative jQuery towards declarative, component-based ways to structure applications. In other words, the JavaScript language is evolving and new development tools appear following these changes.

React is one of the main players in this area at the moment. React is a free JavaScript library developed by Facebook. The source code of the library was published in 2013, and since then the technology has grown in popularity. Now it is used by such large companies as Instagram, Netflix, Yahoo, Dropbox, and many others.

The library allows you to manage the virtual DOM, component architecture, and their states. Many React apps rely on third-party libraries built and maintained by the programming community. The development of React applications requires a higher level of expertise, so it should be noted that mastering the library is more difficult and the threshold for entering this technology is quite high.

So what are the benefits of React?

Based on simple programming languages.

Extremely flexible application.

Using the DOM.

The possibility to withstand heavy loads.

Excellent cooperation with SEO. Search bots can browse the site more easily, so user interaction with your resource is improved.

An open data library.

Low weight of the database

Does React development have any downsides? Of course, there are shortcomings to any approach. And for React, they lie in the very architecture of the applications being developed. Simply put, applications are heavier.

Since React is an intermediate in the ecosystem of software frameworks, an application developed in React will be slightly larger than an application built using the native libraries from Apple or Google. That is, if you create, for example, an App for a loyalty program or for booking tables in a cafe, then it will take about 25-30 megabytes of the phone memory. However, in most cases, this has little or no effect on how users evaluate the app, given the capacity of flashcards and the speed of broadband Internet access.

The decrease in productivity is observed. Since any actions of a React application go through intermediate libraries, their speed is slightly reduced. However, if your mobile program doesn’t carry out any calculations, then no one will notice the increase in processor load up to 2% instead of 1%. Difficulties can arise when the size of the application is large enough, and we’re talking about a really complex software environment that takes hundreds of megabytes in the phone's memory.

The same goes for performance. The disadvantages of having an additional framework will only affect engineering or computational applications that code something, do some calculations or process large amounts of data right on a mobile device. In the other 95% of cases, React is the optimal way to run the prototype as quickly as possible and get started with the application.

For Which Projects Is React A Better Choice Than JQuery?

ReactJS is a smart choice when building large applications and complex user interfaces. The library allows you to display large amounts of dynamic content that changes during user viewing. Its declarativity simplifies and speeds up the process of product development and release.

With this development tool, you can build a site, application, or specific function from flexible reusable components (tab bars, lists, etc.) without rewriting the code. Thus, it allows developers to create quality applications, with a flexible architecture that can be adapted to any further changes.

There is a trend among developers to opt for React. According to the State of JS, this library has the highest level of satisfaction and more than 70% of developers use it in their work. Since now most large projects are created using this JS library, we also use it in development.

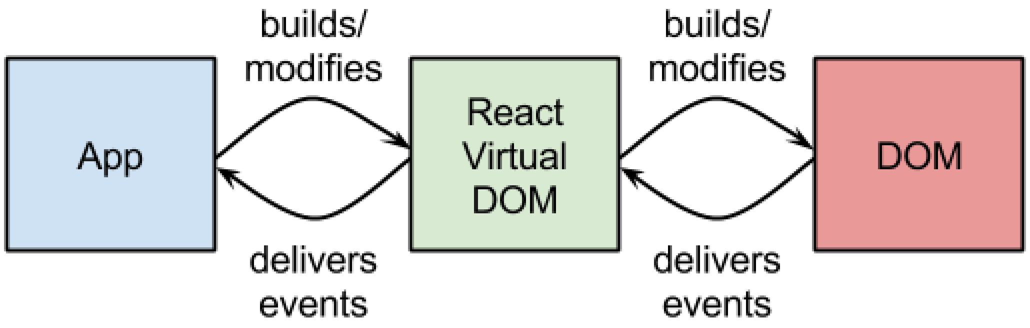
React for Multi-Platforms: Exploring the capabilities of React in developing applications for various platforms

Here are 7 major reasons why some of the giants like Facebook, Instagram, and Whatsapp are relying on ReactJs. Businesses can choose React.js it to build interactive, unique, creative, user-friendly solutions. For startups, small and large enterprises, it is the most suitable, right, and manageable solution.

Well, the below-mentioned advantages of ReactJS can also help your small, moderate, and large enterprises in building leading web applications.

1. Reactjs is extremely efficient

Facebook's React.js creates its own DOM (virtual) where your components live. This approach gives your developer high flexibility and amazing performance gains because React.js calculates what change is needed to be made in the DOM in advance and updates the DOM trees accordingly. In this way, React.js avoids costly DOM operations and does update efficiently.



2. The JavaScript Library

The JSX syntax is a nice and healthy blend of JavaScript and HTML. It is used specifically in ReactJS. JSX simplifies the whole process of writing components for the websites and the HTML aspect allows your developers to render functions without concatenating strings.

The major advantage of ReactJS is, that it makes proper use of native APIs and as a result, JavaScript makes the stack work across the platforms.

3. It’s awesome for SEO

One of the major challenges with JavaScript frameworks is, that they are not at all search engine friendly. Though there have been some improvements in this area recently, however, that’s not very helpful.

Surprisingly, React.js stands out, as you are able to run React.js on the server and the virtual DOM will be returned and rendered to the browser as a regular web page. No need for any other tricks!

4. Focused on the User interface

React Native is much more focused on the user interface, unlike MeteorJS, Firebase, and AngularJS. It gives your users a highly responsive interface with the help of JavaScript interactions between the Native environment of the device and React Native. As a result, this increases the application's load time and helps to keep it running very smoothly without any interruptions.

5. It's easy and out of the box

When you start your project with Facebook's React.js, don't forget to install the official Chrome extension of React.js. It makes debugging your app much easier.

After installing the extension, you can have a direct look into the virtual DOM as if you were browsing a regular DOM tree in the element's panel. Quite amazing!

6. Reusable React Components

Another advantage of Facebook’s React.js is it offers the ability to reuse code components of a different level anytime. This is a very meaningful time-saving effect. ReactJS components are isolated and change in one doesn’t affect the others. This allows you to reuse components that do not produce changes in and of themselves to make programming more precise, ergonomic, and comfortable for developers.

7. Raise Productivity

The reason behind using ReactJS or this convert-able framework is its performance and all those features or components that it contributes to making development simple.

Features such as Virtual DOM, component reusability, Backward-Compatibility, and flexibility for other platforms, enable the developers to craft applications that grant excellent UI with intricate workings.

If we talk about the impact of ReactJS on business app productivity, then the apps produced using ReactJS provide transcendent UI, with consumer-grade simplicity. That’s the reason why ReactJS is the most suitable for business app development.

Numerous businesses are confused about technology adoption because there are several technologies in the market, and new technologies are also released day by day. So, which technology is best? It is ReactJS. Yes, ReactJs met all the requirements of many Businesses.

Famous Companies Using ReactJS

Right now, React has stepped beyond the walls of Facebook, and dozens of renowned businesses have all integrated React into their web applications. ReactJS is benefiting various industries that’s why they are preferring to use it.

Instagram

Instagram is one popular social media platform, used lavishly by the modern generation and the corporate world, to showcase videos, photos, and information. The Instagram app includes modern-day features.

Instagram used the ReactJS library in its development and allowed the developers to enjoy its profits. Just because of React, the Instagram web proffers a fast performance and is speedily responsive to various user-driven events.

Instagram features like Geolocations, progressive loading, Google MAPs, image/video delivery, uploading, etc. are applied because of ReactJS.

Netflix

Netflix is a suggested option when it comes to watching the online latest video. The development team at Netflix has leveraged the ReactJS segments. Due to the use of React, Netflix now offering the top performance to the users.

Netflix utilizes React on Gibbon. Netflix app includes multiple features such as Ultra HD 4K content, HDR content, Dolby Vision content, Play Netflix roulette, and so on, and this became possible because of using ReactJS

Yahoo! Mail

Yahoo!'s email client version utilizes ReactJS. The Yahoo! development team finds ReactJS simple to operate, debug and learn. Yahoo Mail offers top rating performance to the end-users, and this became possible because of employing React.

Twitter, WordPress, New York Times, Airbnb, Facebook, Uber, and so on also use ReactJS. The React.js development companies which use React know how to optimize applications for easy interaction usability.

how React functions on platforms other than web browsers

React Cross Browser Compatibility

ReactJS offers code reusability, where a single piece of code for a UI component can be used across different platforms. This support for all the major browsers and platforms reflects React Cross Browser compatibility. Although a lot of the cross-browser compatibility is taken care of by ReactJS itself, Cross Browser Testing is still important as some of the older versions of browsers have few limitations.

However, as React has individual UI components, Cross Browser Testing becomes more accessible, and managing UX consistency across different browsers, platforms, and devices are simple.

The testers can check for the HTML5 and CSS codes of the elements to ensure React browser compatibility where certain features might cause inconsistencies for several browser versions.

Additionally, older versions can be supported by adding polyfills in a ReactJS web app to achieve cross-browser compatibility. These polyfills are third-party JS files that work similarly to JS libraries.

However, polyfills are also capable of providing new functionalities. For example, a polyfill can support ES6-based features in browsers that fundamentally don’t.