Въпрос 1 – ip

IP адрес е уникален номер, който се присвоява на всеки устройство, което има достъп до Интернет. Има две основни версии на IP адреса: IPv4 и IPv6.

IPv4 адресите са от типа „xxx.xxx.xxx.xxx“, където xxx е число между 0 и 255. Например, „192.168.0.1“ е валиден IPv4 адрес. IPv4 адресите са ограничени до около 4 милиарда, което е основната причина за интродукцията на IPv6.

IPv6 адресите са от типа „xxxx:xxxx:xxxx:xxxx:xxxx:xxxx:xxxx:xxxx“, където xxxx е шестнадесетично число. Например, „2001:0db8:85a3:0000:0000:8a2e:0370:7334“ е валиден IPv6 адрес. IPv6 адресите са много по-големи от IPv4 адресите и те имат много повече адреси, което означава, че има достатъчно адреси за всички устройства и интернет услуги.

Въпрос 2 - TCP

TCP (Transmission Control Protocol) е един от основните протоколи за мрежово взаимодействие в Интернет. Той е създаден за да обезопаси и контролира предаването на данни между различни устройства в мрежата.

TCP се използва за създаване на стабилна и надеждна връзка между два устройства, като използва механизми за контрол на потоците и контрол на грешките. Той е основен протокол за много от приложенията в Интернет, като например интернет браузъри, e-mail клиенти, файлови мениджъри и други.

Въпрос 3 - UDP

UDP (User Datagram Protocol) е друг популярен протокол за мрежово взаимодействие. Той също като TCP, се използва за изпращане и получаване на данни между различни устройства в мрежата.

UDP е по-малък и по-прост протокол от TCP. В отличие от TCP, UDP няма механизми за контрол на потоците и контрол на грешките. Това означава, че UDP не гарантира, че данните ще бъдат доставени успешно и не гарантира, че данните ще бъдат получени в правилна последователност.

UDP се използва в приложения, където е нужна по-висока ефективност и по-малко контрол в сравнение с TCP. Например, в мрежовите игри, голосово комуникация и видео транслации използват UDP, защото това позволява да се избегнат забавяния и да се поддържа по-висока качество на видеото и звука.

Въпрос 4 – GET

GET е метод, който се използва в HTTP (Hypertext Transfer Protocol), който се използва за комуникация между веб сървъри и клиенти. GET метода се използва за да се извлечат данни от сървъра и да се покажат на клиента.

GET заявката може да съдържа допълнителна информация в формата на параметри, които могат да бъдат използвани от сървъра за да се извлекат интересуващите данни. GET заявката е към данни и не прави промени в сървърната база данни, затова тя е безопасна и може да се кешира.

GET заявката се използва когато искаме да вземем информация от сървъра, например когато искаме да видим страница в интернет браузъра или да изтеглим файл. GET метода е използван в повечето браузъри по подразбиране за търсене на страница.

Въпрос 5 – POST

POST е метод, който се използва в HTTP (Hypertext Transfer Protocol) и се използва за изпращане на данни от клиента към сървъра. POST заявката се използва, когато искаме да изпратим данни към сървъра и да правим промени в сървърната база данни.

В момента на изпращане, POST заявката съдържа данните към сървъра в тялото на заявката, като те са обикновено във формат на форма или данни в JSON формат. POST заявката може да има резултат, като ново създаден документ или обновен ресурс, и може да се използва за създаване, обновяване или изтриване на данни от сървъра.

POST методът се използва за изпращане на данни към сървъра при изпълнение на различни действия, като например изпращане на формуляр, качване на файл или изпращане на данни.

Въпрос 6 – PUT

Put е команда, която се използва в различни контексти, като програмиране, командна линия или системно администриране.

В програмиране, put често се използва като метод за добавяне на елемент в колекция данни, като например масив, списък или дърво. Например в Ruby Hash има метод put , който добавя нов ключ-стойност към хеш-таблица.

В командната линия, put може да се използва за качване на файлове на удален сървър или за съхраняване на данни във файл. Например, командата scp (secure copy) използва put като субкоманда за качване на файлове към удален сървър.

В системно администриране, команда put може да се използва като средство за конфигуриране на устройства, като сървъри или маршрутизатори, като използва интерфей.

Въпрос 7 – DELETE

„Delete“ е команда, която се използва за изтриване на файлове, папки или данни в различни компютърни програми и операционни системи. В зависимост от контекста, „delete“ може да има различни действия и да изисква различни начини за извършване на операцията.

Например, в Windows може да се използва командата „delete“ за изтриване на файл или папка, като изберете файла или папката и натиснете клавиша Delete на клавиатурата, или да използвате контекстното меню, като десен клик на мишката върху файла или папката и избирате „Delete“

В програмните езици за програмиране, функцията „delete“ се използва за изтриване на обект или данни, които са били динамично заделени в паметта. Това може да включва изтриване на масиви, списъци, дървета и други структури от данни.

В други езици като Python, специална функция или метод не е нужен за изтриване на динамично заделени обекти от паметта, вместо това системата за управление на паметта автоматично освобождава паметта след като обектът пестят своите референции.

Въпрос 8 - PATCH

PATCH е един от HTTP методите, които се използват за модифициране на ресурси на уеб сървъра. Той се използва, когато искате да промените някакви части от данните на ресурса, без да променяте цялата информация. Това е по-ефективно от използването на метода PUT, който изисква полното изпращане на всички данни за ресурса.

PATCH заявката се състои от заглавие „PATCH“ и следва го следва URL на ресурса който се модифицира, и в тялото има разликата на данните които се иска да се модифицира. В зависимост от реализацията на сървъра и клиента и може да се използва някой от стандартните формати за данни като JSON или XML.

PATCH методът е включен в HTTP/1.1 спецификацията.

Въпрос 9 - клиент сървърна комуникация

Клиент-сървърната комуникация е модел на комуникация, в който два компютъра (клиент и сървър) обменят информация. Клиентът изпраща заявки до сървъра, а сървърът отговаря с данни.

Например, когато използвате браузъра си да отворите веб страница, браузърът Ви се представя като клиент и изпраща заявка до сървъра, на който се намира тази страница, да й я достави. Сървърът обработва заявката и изпраща HTML, CSS и JavaScript кода, който се използва за да се покаже страницата в браузъра Ви.

Този модел може да се използва в множество други приложения и системи, като електронна поща, файлови сървъри, бази данни и много други.

Въпрос 10 – css

CSS (Cascading Style Sheets) е технология за оформление и оформляване на документи, която се използва заедно с HTML или XML. С помощта на CSS може да се определят стиловете за различните елементи на документа, като цветове, размери, отстояния и много други. Това позволява на изгледа на документа да се сепарира от неговата структура и съдържание, така че е по-лесно да се променя или поддържа.

CSS позволява да се приложат стилове за групи от елементи, като например всички елементи с определен клас или ID, така че промените да се изпълняват навсякъде, където се използва този клас или ID. Това е една от най-силните страни на CSS.

CSS3 добавя много нови свойства и функционалности, които позволяват да се създават по-интересни и динамични дизайни. Някои от тези свойства включват:

Въпрос 11 – семантични страници

„Семантични страници“ е термин, който се използва за описване на страниците в Интернет, които са оптимизирани за да се достигне добър качествен ранжиране в търсачките и да се обезпечи по-добър потребителски опит.

Семантичната оптимизация е процес по който се улеснява за търсачките да разберат съдържанието на страницата и да го свържат с заявените ключови думи. Това се постига чрез структуриране на съдържанието, използване на мета-тагове и други техники.

Семантично оптимизираната страница трябва да съдържа информация, която е полезна и релевантна за потребителите, и да се представя така, че да се лесно индексира от търсачките.

Въпрос 12 – адаптивно оформление на страници

Създаването на адаптивно оформление на страници означава, че страницата се приспособява към различните размери на екрана и ориентации на устройството, което се използва за преглед. Това означава, че потребителите на телефони и таблети могат да видят една и съща страница, но с приложения, които са оптимизирани за тяхната конкретна ориентация и размер на екрана.

Адаптивното оформление може да се използва за решаване на различни проблеми, свързани с различните размери на екрана и ориентации на устройството. Някой от тях са:

Поддържане на видимостта на важна информация: Адаптивното оформление помага да се поддържа видимостта на ключовата информация, като се избягва затрудняване на четенето или навлизането на контента на малки екрани.

Оптимизиране на интерфейса за мобилни устройства: Адаптивното оформление помага да се създаде интерфейс, който е удобен за използване на мобилни устройства, като се избягва малките бутони или елементи, които са трудно да се натискат.

Въпрос 13 – java script

JavaScript е език за програмиране, който се използва за създаване на динамични страници и интерактивни приложения. Един от основните концепти в JavaScript е работата с обекти.

В JavaScript, обектите са колекции от ключ-стойност пари. Ключовете са стрингове, а стойностите могат да бъдат практически всякакъв тип данни – числа, стрингове, масиви, други обекти, и т.н.

В JavaScript съществуват два основни типа обекти: прототипно ориентирани и класове ориентирани обекти.

Прототипно ориентираните обекти са основани на идеята за наследяване. Всеки обект има прототип, който може да наследява свойства и методи. Това позволява да се използват сходни кодове и да се избегнат дублирания.

Класове ориентираните обекти, които са включени в ECMAScript 6, позволяват да създавате обекти с определена структура и действия, като използвате класове.

JavaScript има много вградени обекти, като Array, Date, Math и String, които могат да бъдат използвани за различни потребности.

Въпрос 14 – Принципи и манипулиране на DOM

DOM (Document Object Model) е интерфейс, който представя документа (обикновено HTML или XML) като дърво от обекти. Това дърво може да бъде манипулирано чрез JavaScript, за да се променят или добавят елементи към документа.

Принципите на DOM са следните:

Всеки елемент в документа е обект. Елементите могат да бъдат достъпвани и манипулирани чрез JavaScript.

Всеки елемент може да има деца, родител и съседи. Това означава, че може да се навигира по дървото на документа, за да се достъпят и манипулират елементи.

Всеки елемент може да има собствени атрибути и стилове. Те могат да бъдат достъпвани и манипулирани чрез JavaScript.

Събития, като кликване на бутон или зареждане на страницата, могат да бъдат слушани и обработвани чрез JavaScript.

DOM (Document Object Model) е стандартна модел за описване на документите в HTML и XML. Тя представя документа като дърво от елементи, където всеки елемент е възел в дървото. Всеки елемент има свойства и методи, които могат да бъдат използвани за манипулиране на елементите в документа.

Някой от принципите и методите за манипулиране на DOM са:

Document.getElementById() – този метод може да се използва за достъп до елемент по ID.

Document.getElementsByTagName() – този метод може да се използва за достъп до група от елементи с определен таг.

Element.innerHTML – това свойство може да се използва за четене или промяна на съдържанието на елемент.

Element.appendChild(child) и element.removeChild(child) – тези методи може да се използват за добавяне или премахване на дете към елемент

Element.classList – това свойство дава достъп до класовете, които са зададени за елемента. С помощта на classList може да се добавя, премахва или тогава да се проверява дали елемент има конкретен клас.

Element.addEventListener(event, function) – този метод може да се използва за добавяне на слушател на събития към елемент. Примерни събития са „click“, „mouseover“ и „keydown“.

Element.querySelector() и element.querySelectorAll() – тези методи може да се използва за търсене на елементи в документа искайки по css-селектор. querySelector връща първия елемент, който се съответства на критерия, докато querySelectorAll връща колекция от елементи.

Да, както и да се използват за извличане на информация от документа и добавяне или премахване на елементи или събития. Но това може да доведе до появата на проблеми с производителността, ако не се извършват правилно. Това е добре да се избягва честотното манипулиране на DOM и да се използва както се може по-оптимизиран код, като използването на DocumentFragment, requestAnimationFrame и кеширане на елементите.

JavaScript библиотеки, като jQuery, могат да дадат интуитивен и лесен начин за манипулиране на DOM, като скриват много от детайлите и комплексността на манипулациите.

В цялостно, манипулирането на DOM е критична част на разработката на динамични интерактивни веб страници и приложения, и е добре да се има достатъчно знания и практика за да се използва умело.