|  |
| --- |
| **Exam knowledge** |
|  |  |
|  | - **Git** |
|  | **- git commit**  Ans: Save your changes to the local repository  The git commit command captures a snapshot of the project's currently staged changes. Committed snapshots can be thought of as “safe” versions of a project—Git will never change them unless you explicitly ask it to. |
|  | **- git clone**  Ans: git clone is primarily used to point to an existing repo and make a clone or copy of that repo at in a new directory, at another location. The original repository can be located on the local filesystem or on remote machine accessible supported protocols. The git clone command copies an existing Git repository. |
|  | **- git merge**  Ans: Merging is Git's way of putting a forked history back together again. The git merge command lets you take the independent lines of development created by git branch and integrate them into a single branch. |
|  | **- git push**  Ans: The git push command is used to upload local repository content to a remote repository. Pushing is how you transfer commits from your local repository to a remote repo.  **- git pull**  Ans: The git pull command is used to fetch and download content from a remote repository and immediately update the local repository to match that content |
|  | **- What is a Git conflict?**  Ans: Version control systems are all about managing contributions between multiple distributed authors (usually developers). Sometimes multiple developers may try to edit the same content. If Developer A tries to edit code that Developer B is editing a conflict may occur. To alleviate the occurrence of conflicts developers will work in separate isolated branches. The git merge command's primary responsibility is to combine separate branches and resolve any conflicting edits. |
|  | **- git status**  Ans: The git status command displays the state of the working directory and the staging area. It lets you see which changes have been staged, which haven't, and which files aren't being tracked by Git. Status output does not show you any information regarding the committed project history. |
|  | **- git diff**  Ans: The git diff command displays the differences between files in two commits or between a commit and your current repository. You can see what text has been added to, removed from, and changed in a file. By default, the git diff command displays any uncommitted changes to your repository. |
|  | **- git log**  Ans: The git log command displays a record of the commits in a Git repository. By default, the git log command displays a commit hash, the commit message, and other commit metadata. You can filter the output of git log using various options. |
|  | **- git branch**  Ans: A branch represents an independent line of development. The gitbranch command lets you create, list, rename, and delete branches. It doesn't let you switch between branches or put a forked history back together again. For this reason, git branch is tightly integrated with the git checkout and git merge commands. |
|  | **- What is a branch?**  Ans: A branch in Git is simply a lightweight movable pointer to one of these commits. As you start making commits, you're given a master branch that points to the last commit you made. Every time you commit, the master branch pointer moves forward automatically. Note. The “master” branch in Git is not a special branch. |
|  | **- git checkout**  Ans: The git checkout command lets you navigate between the branches created by git branch |
|  | **- git add .**  Ans: The git add command adds a change in the working directory to the staging area. It tells Git that you want to include updates to a particular file in the next commit. However, git add doesn't really affect the repository in any significant way—changes are not actually recorded until you run git commit. |
|  | **- What is respository?**  Ans: In Git, the repository is like a data structure used by VCS to store metadata for a set of files and directories. It contains the collection of the files as well as the history of changes made to those files. Repository in Git is considered as your project folder. A repository has all the project-related data. |
|  | **- What is a commit?**  Ans: The "commit" command is used to save your changes to the local repository. Note that you have to explicitly tell Git which changes you want to include in a commit before running the "git commit" command. This means that a file won't be automatically included in the next commit just because it was changed. |
|  | **- What is a good commit message?**  Ans: The seven rules of a great Git commit message   * Separate subject from body with a blank line. * Limit the subject line to 50 characters. * Capitalize the subject line. * Do not end the subject line with a period. * Use the imperative mood in the subject line. * Wrap the body at 72 characters. * Use the body to explain what and why vs. how. |
|  | **- JavaScript/HTML/CSS** |
|  | **- document (what is this?)**  A document is the name of a message sends over HTTP. It's the data unit for the web. A document moves on the internet and is understand by the web client (browser mainly). When you serve a file via a web server, it's wrapped inside an http request and becomes a document |
|  | **- What is the browser DOM?**  The Document Object Model (DOM) is a programming interface for HTML and XML documents. The DOM is an object-oriented representation of the web page, which can be modified with a scripting language such as JavaScript. |
|  | **- What is a function()?**  A function in JavaScript is similar to a procedure—a set of statements that performs a task or calculates a value, but for a procedure to qualify as a function, it should take some input and return an output where there is some obvious relationship between the input and the output. |
|  | **- CSS Selectors:**  **.className**, - To select the class  **#myId,** - To select the ID  **body > div** – to select all the div coming under body tag |
|  | <https://www.w3schools.com/cssref/css_selectors.asp> |
|  | **- JS can use CSS selectors with querySelector**  The querySelector() is a method of the Element interface. The querySelector() allows you to find the first element that matches one or more CSS selectors. You can call the querySelector() method on the document or any HTML element. ... If no element matches the CSS selectors, the querySelector() returns null. |
|  | **- What is the meaning of DOMContentLoaded**  The DOMContentLoaded event fires when the initial HTML document has been completely loaded and parsed, without waiting for stylesheets, images, and subframes to finish loading. |
|  | **document.addEventListener('DOMContentLoaded', domEventHandler);** |
|  | **- What is a HTML tag?**  The <html> tag represents the root of an HTML document.  The <html> tag is the container for all other HTML elements (except for the <!DOCTYPE> tag).  Note: You should always include the lang attribute inside the <html> tag, to declare the language of the Web page. This is meant to assist search engines and browsers. |
|  | **- What is HTML tag attribute?**  HTML attributes are special words used inside the opening tag to control the element's behaviour. An attribute either modifies the default functionality of an element type or provides functionality to certain element types unable to function correctly without them. |
|  | - HTML Rules: |
|  | - if a tag opens, it must be closed |
|  | - visible content -> body, meta info -> head |
|  | - CSS: what is flex/grid?  <https://css-tricks.com/snippets/css/a-guide-to-flexbox/>  <https://css-tricks.com/snippets/css/complete-guide-grid/>  **- what is the virtual dom?**  A virtual DOM object has the same properties as a real DOM object, but it lacks the real thing's power to directly change what's on the screen. Manipulating the DOM is slow. Manipulating the virtual DOM is much faster, because nothing gets drawn onscreen. |
|  | **- React** |
|  | **- What is node? What is NPM?**  Ans : Node.js is an open-source, cross-platform, back-end JavaScript runtime environment that runs on the V8 engine and executes JavaScript code outside a web browser.  NPM is two things: first and foremost, it is an online repository for the publishing of open-source Node.js projects; second, it is a command-line utility for interacting with said repository that aids in package installation, version management, and dependency management. |
|  | - **What is React? What is a React components?**  Ans: React is an open-source front-end JavaScript library for building user interfaces or UI components. It is maintained by Facebook and a community of individual developers and companies. React can be used as a base in the development of single-page or mobile applications.  Components are independent and reusable bits of code. They serve the same purpose as JavaScript functions, but work in isolation and return HTML via a render() function. Components come in two types, Class components and Function components  **Types of React Components you should know**  1] Functional Components  2] Class Components  3] Higher-Order Components  4] Dumb Components  5] Smart Components  6] Presentational Components  7] Container components |
|  | **- What is the React State? setState/intiial state**  Ans: The state is an instance of React Component Class can be defined as an object of a set of observable properties that control the behavior of the component. In other words, the State of a component is an object that holds some information that may change over the lifetime of the component. |
|  | **- Rules for a React constructor (call super constructor with props)**  constructor(props) {  super(props);  this.state = {  reptile: props.reptile,  };  } |
|  | **- onClick (event handling)**  <button onClick={activateLasers}>  Activate Lasers  </button> |
|  | **- What are React properties?**  Data within React Components is stored as either properties or state. Properties are the input values to the component. They are used when rendering the component and initializing state (discussed shortly). After instantiating the component, properties should be considered immutable. |
|  | **- How does a basic React class look like?** |

import React from 'react';

class Basic extends React.Component {

render() {

return (

<div></div>

);

}

}

export default Basic;