Viking Code School Test questions & answers

All About Web Developers

1. **What does a Web Developer do?**

   Builds and maintains websites /web apps.

2. **What's the difference between web developers and web designers?**

   Web developer focuses more on the functionality of the website while web designer focuses more on the look and feel of the website.

3. **What's the difference between front end, back end, and full stack developers?**

   Front end developer focuses more on the browser side of the website while back end developer focuses more on the server side of the website.

4. **What's the difference between static and dynamic web pages?**

   Dynamic web pages usually require server side, static don't.

5. **What's the best way to get help when you get stuck?**

   Google & stuckoverflow.com

6. **Where are the closest developer communities to you?**

   Meetups, Hack Nights, Hackatons, Conferences, Hacker Spaces, School Clubs

7. **What online developer communities can you join?**

   Hacker News, Github, Stack Overflow, IRC - chat for programmers, Reddit's /r/programming.

8. **Why is it better to use a developer-friendly text editor like Sublime Text than something oriented more towards word-processing like Microsoft Word?**

   Sublime Text like editors are more focused on coders. Code editors let me view full directories, highlight different bits of syntax in many languages and have efficiency-gaining code snippets.

9. **What are the top 3 characteristics of great web developers?**

   Desire to built, problem-solving mind and persistence in the face of setbacks.

10. **What do companies look for in a junior web developer?**

    Startups.

11. **What is the hiring process like for a developer?**

    Testing my ability to think technically(thought questions, brain teasers, coding exercises), examination of my previous work. I need to demonstrate my ability to pick up things quickly, complete projects and work well with others.

How the Web Works

1. **What is the internet?**

   A global computer network providing a variety of information and communication facilities, consisting of interconnected networks using standardized communication protocols.

2. **How is information broken down and sent?**

   E-mails or web pages are broken down into smaller packets of information, because there is limited room for message transmission along the Internet. Each packet is then sent off to its destination by the best available route. Upon arrival at the destination computer, the packets are recombined into the original message.

3. **What are packets?**

   Parts of a certain size in bytes carying information from sender's IP address to intended reciever's IP address. The packets carry the data in the protocols that the Internet uses: Transmission Control Protocol/Internet Protocol (TCP/IP). A typical packet contains 1,000 or 1,500 bytes.

4. **What is a "client"?**

   A desktop computer or workstation that is capable of obtaining information and applications from a server.

5. **What is a "server"?**

   A computer or device on a network that manages network resources.

6. **What is HTTP and how does an HTTP request work?**

   Hyper Text Transfer Protocol. Used for comunication between servers and clients, making sure everyone speaks the same language. Client will initiate the communication by sending an HTTP Request and the Web Server will respond back by sending an HTTP Response.

7. **What are DNS servers?**

   Domain naming system. Maps domains to IP addresses.

8. **What is HTML and how is it used?**

   Hypertext Markup Language, a standardized system for tagging text files to achieve font, colour, graphic, and hyperlink effects on World Wide Web pages. It creates a structure of the web page.

9. **What is CSS and how is it used?**

   Cascading style sheets used for describing the look and formatting of a document written in a markup language.

10. **What is your browser's Web Inspector (aka Developer Tools) and how can you use it to poke around in a page's HTML?**

    It lets me inspect many things like the HTML, CSS, number of HTTP requests made, loading speed of the web page, debbug javascript.

11. **What happens behind the scenes after you click "search" on google.com?**

    Google navigates web by crawling (folowing links from page to page). Algorithms are looking for clues to better understand what I mean.

The Command Line

1. **What is the command line?**

   An interface for typing commands directly to a computer's operating system.

2. **How do you open it on your computer?**

   Keyboard/mouse shortcut, type in search, double-click on my desktop..., cmd+space-> type

3. **What is Bash?**

   Bash is the shell, or command language interpreter, for the GNU operating system.

4. **How can you navigate into a particular file directory?**

   cd

5. **How can you create a directory?**

   mkdir

6. **How can you destroy a directory or file?**

   rm

7. **How can you rename a directory or file?**

   mv

8. **Why are file permissions important?**

   because they let me decide the level of security for my files. i want

9. **How do you view hidden files in a directory?**

   ls -a

10. **How do you find information about a particular command?**

    man command; “man ls” for example

11. **What is a "Superuser" and how do you execute commands as this user?**

    Sudo allows a permitted user to execute a command as the superuser or another user.

12. **What is Vim?**

    Vim is a highly configurable text editor built to enable efficient text editing. It is an improved version of the vi editor distributed with most UNIX systems.

13. **How do you quit Vim if you get stuck in it?**

    Esc -> :q

14. **What is the bash_profile file and what is it used for?**

    This file is loaded before Terminal loads your shell environment and contains all the startup configuration and preferences for your command line interface.

15. **What is the $PATH variable?**

    $PATH variable has a listing of all the directories Bash should search through whenever you give it a command in order to figure out what exactly the command does.

16. **Why might you need to add onto your $PATH variable?**

    If I’ve just installed something and I’m getting an error messages saying it can't be found, odds are that I need to add that application's location to my $PATH variable.

17. **What are alias commands?**

    An "alias" lets me specify a simple command that I want to type in order to run some other command or batch of commands. Time saver!

Define the following:

The Web

1. **Domain**

   Specifies the location of the web server responding to my request.

2. **IP Address**

   Definitive address of a computer or server on the web.

3. **HTTPS**

   Used for secure communication between clients and servers - secure transactions for example

4. **POP & IMAP & SMTP**

   Email protocols.
   SMTP = Simple Mail Transfer Protocol, SMTP is used when email is delivered from an email client.
   IMAP = Internet Message Access Protocol, it is designed to let users keep their email on the server.
   POP3 = Post Office Protocol, it offers only a download feature.

5. **Web Server**

   Actively recieves and responds to incoming HTTP requests from browsers or apps.

6. **The Cloud**

   Farms of servers managed by giants like Google or Amazon.

7. **Cookies**

   Snippets of text web app can drop into my browser memory. They help servers to remember me.

8. **404 Error**

   Page that user can't access, because it does not exist.

Design

1. **WYSIWYG**

   What you see is what you get. Usually associated with graphical based website editors, that don't require knowledge of the code.

2. **Responsive Design**

   Website that is able to resize based on the device used to see it.

3. **Grid System**

   A way of visually organizing the content on my page.

4. **Above the Fold**

   Everything the user sees without having to scroll down.

5. **Front End / Back End**

   The look and feel of the website and it's content, the browser side / funcitonality of the website, the server side.

Developer Tools and Code

1. **JavaScript**

   Making web pages dynamic. Thanks to JS, things are moving from servers to browsers.

2. **Pseudo-Code**

   It is somewhere between the logic in english sentences and expression in the code.

3. **Ruby Gems**

   A package manager for the Ruby programming language that provides a standard format for distributing Ruby programs and libraries. A tool designed to easily manage the installation of gems, and a server for distributing them.

4. **Frameworks**

   Bundles of premade code that saves web developer’s time when creating web apps.

5. **Ruby on Rails**

   Fastest way to build web apps, framework written in Ruby.

6. **Command Line**

   An interface for typing commands directly to a computer's operating system.

7. **Version Control**

   Version control is a system that records changes to a file or set of files over time so that you can recall specific versions later.

8. **Open Source**

   Denoting software for which the original source code is made freely available and may be redistributed and modified.

9. **Database**

   Place where everything users do is saved.

10. **Schema**

    XML document that specifies the structure of other XML documents.

11. **Script**

    High level code that is translated into machine runnable instructions.

Installations

1. **Why did you need to install Ruby?**

   Ruby is the back end language we'll be using to write our server code.

2. **Why didn't you need to install HTML, CSS and JavaScript?**

   They come with my browser already.

3. **What is RVM and how will it be useful later?**

   Ruby Version Manager is a way of making sure that each Ruby or Rails project on your computer is treated independently of each other one. It allows you to install multiple versions of Ruby and multiple versions of Rails or any other gem on your computer and then you can choose which set to use for a given project.

4. **What is XCode?**

   XCode is Apple's integrated development environment for creating Mac, iPhone and iPad applications.

1. **What is Git?**

   Version-control system.

2. **What does SCM stand for?**

   Software Configuration Management.

3. **What is a VCS?**

   Version control system.

4. **Why is Git useful for a developer?**

   It lets me see my project project changes and if something goes wrong, revert to these changes.

5. **Why is Git useful for a team of developers?**

   Multiple people can work on one file / project, git makes it all sync together.

The Basics

1. **How do you create a new Git repository for a project locally?**

   git init

2. **How do you create it on Github?**

   create new repository

3. **How do you commit changes?**

   git commit -m “message”

4. **What is the difference between staging and committing changes?**

   Staging comes before committing.

Remotes

1. **What is the difference between committing your changes and pushing them to Github?**

   Commiting is local, pushing is sending local commits to Github.

2. **What is the difference between a "remote" and your local repo?**

   remote is an online version of my local repo

3. **How do you add your Github repo as the remote?**

   git remote add origin https://github.com/user/repo.git

4. **How do you check the status of your current repo in git?**

   git status

5. **How do you see the history of your previous commits (from the command line)?**

git log

6. **How can you look through your historical commits on the Github website?**

   I click on commits history.

Branching and Merging

1. **What is a "Merge"?**

   Merging changes between two branches.

2. **What is a "Pull Request"?**

   Pulling changes from my online repository to local.

3. **What is "Forking" a repo?**

   Creating my own branch of someone else's project.

4. **What is "Cloning" a repo?**

   Duplicating my already existing project.

5. **What is "Branching"?**

   Creating of an additional version of an existing project.

6. **Why would you use a branch?**

   To fix an error, to work on an idea I'm not sure about, or to simply make a different version.

7. **How do you create a new branch?**

   git branch branch_name

8. **How do you merge branches back together?**

   git merge branch_name

9. **What is a "Merge Conflict" and when will it occur?**

   Two different versions of one file, where I have to decide which one to keep.

10. **How do you resolve a "Merge Conflict"?**

    Remove the part of the content I don't want to keep and this other stuff that shows me which version is which.