

El día de ayer, estuve trabajando en 2 cursos complementarios, como Frontend Beginner. Los cursos eran: Introduction to Dev Tools, v3 y Modern Search Engine Optimization (SEO) . En el primer curso me sentí bien hacker, porque me metí a varias páginas web, a alterar casi todos los elementos desde la pestaña Elements de las DevTools. Estuve trabajando con la pestaña Console, Source, Network y conocí para qué sirven las pestañas Performance, Memory, Application y Security. Me gustó mucho poder conocer qué son las DevTools, para que están, cómo se utilizan y cómo se pueden implementar en el día con día para tener una mejor experiencia como desarrollador. Algo que me gustó bastante de las DevTools fue una herramienta llamada Lighthouse ya que si uno como desarrollador quiere probar su aplicación web y darse cuenta de si esta bien desarrollada o no, esta herramienta da un test sobre todo lo que una aplicación web debería tener y pienso que realmente si ayuda demasiado para poder arreglar problemas que tenga la app y poder entregar un buen resultado. Luego de ello, pasé al 2 curso. En ese curso aprendí de igual manera muchas cosas interesantes, sobre algo llamado SEO (Search Engine Optimization) . En este curso, obtuve bastantes sitios web para poder utilizarlos en mi path como desarrollador o en cualquier situación requerida. Aprendí un montón sobre Metadatos: Facebook OpenGraph y Twitter OpenGraph. Luego habían otras partes como: Structured Data, Traditional SEO Strategies, Mobile Optimizations y AMP Projects . El día de ayer lo disfruté bastante. Pensar que ya estoy en mi 5ta semana y que estoy muy concentrado y activado. Pienso que al ritmo en el que voy, puedo seguir dando más de mí y todo el conocimiento que voy adquiriendo, me irá dando más experiencia y madurez.

El día de ayer inicié el nuevo stack de cursos llamado: Frontend Professional. Me está gustando mucho este nuevo stack de cursos porque van orientados 100% a :js: y tecnologías relacionadas con el lenguaje. Ayer empecé con el primer curso llamado: JavaScript: The Hard Parts, v2 y lo que me gustó bastante de la persona que está impartiendo el curso, es que en vez de que las partes difíciles de :js: sean así, las explica y las hace ver mucho más fácil. El día de ayer estuve trabajando en las partes del curso: JavaScript Principles, Functions and Callbacks y Closures. El día de hoy estaré trabajando en los 24 ejercicios de Callbacks y 24 ejercicios de Closures. Ayer estuve on fire con los 7 primeros ejercicios de Callbacks. Me siento super contento porque al fin estoy poniendo en práctica todo el conocimiento y nuevas metodologías que había estado aprendiendo en todos los cursos que había visto. Así que a darle con todo.

El día de hoy estuve trabajando en el curso: JavaScript: The Hard Parts, v2 . Ha estado muy entretenido este curso porque la persona encargada del curso, explica muy bien, he podido comprender y realizar estructuras funcionales de algoritmos con :js: , me he sentido más confiado de lo que hago cuando programo y siento que eso es lo que necesitaba. He estado aprendiendo demasiado, pero no lo había estado poniendo tanto en práctica para decirme: "Si, lo puedes hacer". Por la mañana estuve trabajando en unos ejercicios de Callbacks pero luego decidí mejor, dejarlos porque, también necesito terminar el curso y los demás cursos que vienen. Luego de ello me hice 5 ejercicios de Closure y 5 ejercicios de Asynchronous . Este curso me ha despertado muchas cosas técnicas que ya sabía en teoría y que ahora estoy poniendo en práctica. El día de ayer hice las partes de: Asynchronous JavaScript, Promises y esto a punto de terminar la última llamada: Classes & Prototypes . El día de hoy terminaré esa parte y así poder iniciar el segundo curso de :js: llamado: Deep JavaScript Foundations, v3 . Este es uno de los más largos del Frontend Professional , pero lo que me gusta, es que el que enseña ese curso, explica demasiado bien y ya con las bases que adquirí y sigo adquiriendo, voy con grandes expectativas para poder desarrollarlo de la mejor manera y seguir aprendiendo más cosas de este asombroso lenguaje de programación.

El día de ayer estuve trabajando en las últimas partes del curso: JavaScript: The Hard Parts, v2 . Algo que se me olvidó comentar y que fué super chivo, fue el conocer el poder que tiene la palabra reservada `fetch` . Conocí como trabaja `:js:` , desde lo que es el global memory, local memory, el trabajo asíncrono que tiene el lenguaje con el Navegador, y las herramientas con las que ambos se comunican y ayudan entre sí. Pude finalizar la parte de: Classes & Prototypes , estuve viendo y entiendo como se hacen los prototypes en `:js:` con solo 2 funciones y objetos y entendí como funciona la `this` keyword y la `new` keyword. Luego de ello, finalicé el curso, y gracias a ello obtuve un mente super despejada y clara de como funciona `:js:` . Luego de ello, estuve trabajando en unos ejercicios de Callbacks. Lo bueno es que voy tomando ejercicios, ejemplos y algoritmos de cursos pasados, voy entendiendo como funcionan y en que situación me sirven y los voy implementando para ir realizando los ejercicios que tengo pendientes. Después inicié el 2º curso del stack de Frontend Masters llamado: Deep JavaScript Foundations, v3 . Esta bastante chivo la verdad, ahorita ya pase la parte introductoria, estuve trabajando con uno de los pilares de `:js:` : Types y finalicé la parte de Coercion. Me gusta este curso porque es `:js:` pero en profundidad y me ayuda bastante a poder entender porque sucede esto, encuentro la logica de aquello, se como crear lo otro y me voy sintiendo tipo transformer, me voy autocompletando mi conocimiento. El día de hoy seguiré trabajando con las partes de: Philosophy of Coercion, Equality, Static Typing y trataré de seguir haciendo más ejercicios, ya sea de Closure o Asynchronous . A darle con todo.

El día viernes estuve trabajando en el curso: Deep JavaScript Foundations, v3 . Logré abarcar las partes de Philosophy of Coercion, Equality, Static Typing, Scope, Scope & Function Expressions y Advanced Scope. En Philosophy of Coercion estuve viendo las intentional coercion, implicit coercion, e hice consciencia de que muchas veces no me preocupaba por lo que programaba en :js: , solo lo hacía porque yo pensaba que estaba viendo hacerlo así y ahora he ido cambiando e implementando poco a poco, buenas prácticas con el lenguaje. Al igual manera con Equality , logre entender de manera más profunda, para que sirve == y === , cuando los debo ocupar y porque están estos dos tipos de comparación en :js: . Luego, con Static Typing hice unos codigos en TypeScript , fue mi primera vez programando con ese lenguaje, pero siento que quien enseñaba el curso, en vez de enseñarlo de una manera amplia, en cuanto a las tecnologías: TypeScript y Flow , se enojaba porque pensaba que al desarrollador de :js: lo vuelve ignorante del codigo que escribe y/o lo vuelve más perezoso en cuanto a entender porque funciona esta parte del código o si hay un bug, no sabe realmente como resolverlo, pero yo pienso que si están las herramientas, como para ahorrar tiempo ayudando al desarrollador a encontrar rapidamente los errores que ha cometido, pienso que esta bien. Luego estuve trabajando con Scope . En scope me siento más tranquilo y confiado porque le entiendo y voy conociendo situaciones donde se ocupan los block scope, nested scope la declaración de variables var, let y const, y para que sirven cada una de ellas. Luego estuve viendo scope and function expressions , repasé las naming function expressions, arrow functions y estuve estudiando y entendiendo como pasar una funcion normal a una arrow function paso por paso para no perderme. Después en la parte de Advanced Scope ví las diferencias entre Lexical y Dynamic Scope , el IIFE Patern , el uso de let y var con scopes y ya por ultimo estuve trabajando con algo que nunca había trabajado, llamado Hoisting . Hice unos ejercicios de hoisting y conocí que con var si se puede hacer hoisting pero con let y const no se puede porque :js: dá un error llamado TDZ error que significa Temporal Dead Zone error y esto es así porque const y let no permiten asignación de valores a sus variables antes de su declaración. El día de hoy estaré trabajando con Closure, Objects y Prototypes . Espero terminar el curso: Deep JavaScript Foundations, v3 y así tambien empezar el tercer curso del stack de cursos Frontend Professional llamado: JavaScript: The Recent Parts . Así que a darle con todo.