Programmieren lernen, wo mit sollte ich anfangen?

Beitrag von „Max“ vom 8. September 2021, 19:43

Hallo,

Ich bin Max und ich bin noch nicht volljärig, womit kann ich anfangen zu programmieren, ich denke so an HTML und dann CSS und später JavaScript. Womit kann man soetwas am besten lernen. Und wie fange ich am besten an? Ich möchte ein paar Webseite probieren zu programmieren, und Spiele versuchen zu programmieren

---

Beitrag von „Elektrohorst“ vom 8. September 2021, 20:19

Hallo Max, ich empfehle Dir mit html anzufangen und dann nach und nach in die Bereiche Css und Javascript vorzuarbeiten. Gratis gibt es viele Kure und Tutorials, zb. selfhtml.org. Es gibt auch Kurse, die etwas kosten, die aber auch sehr gut sein können.

Beitrag von „bounty96“ vom 8. September 2021, 21:03

W3Schools - währe mein Mittel der Wahl um mit Webentwicklung anzufangen.

Und ist mein Mittel der Wahl um immer wieder mal was nachzuschen wenn ich etwas nicht auswendig weis.

Beitrag von „grt“ vom 8. September 2021, 21:04

grmpfffff....

also erstmal weder html noch css ist "programmieren" - html ist eine "auszeichnungssprache" das ist struktur, so wie z.b. die config.plist von unserem bootloader. css heisst "cascading stylesheet" und hat auch absolut nix mit programmieren zu tun.

vereinfacht gesagt strukturiert html eine webseite, und css fabriziert die gestaltung dazu. das ganze ist "statisch" - da bewegt sich nicht wirklich viel. interaktivität erreichst du z.b. mit javascript oder auch php, beides scriptsprachen, die interpretiert werden (macht der browser oder der server). in jedem fall bleibst du im dunstkreis des browsers, der webseiten stecken.

ist aber durchaus spannend (zumindest, wenn man mit javascript browserspielchen oder -anwendungen zusammenstrickt, php kann ich nicht leiden..), aber "richtig" programmieren ist noch mal was anderes. da hättest du mit sprachen wie C, C++ oder C# zu tun oder java, oder python - komplexer, als javascript, aber auch um längen vielseitiger und nicht ausschliesslich ans web gebunden.

einfaches C (eine art abgespeckter "C-dialekt") lässt sich prima mit microcontrollern (arduino, ESP8266, ESP32), einem steckbrettchen und ein paar bauteilchen wie widerständen, leds und tipptastern entdecken. ist auch für schülertaschengeld erschwinglich, und die grundprinzipien des programmierens kriegst du dabei durchaus ziemlich gut mit. und C ist fast schon eine der "ursprachen" der programmierung.

kannst mich gerne auch deswegen anpingen (pn)

Beitrag von „Max“ vom 8. September 2021, 22:16

Danke für diese viele Tipps

Elektrohorst Danke für den Tipp

bounty96 Auch ein Danke für die Tipps.

grt Ok, also wäre HTML und CSS und JavaScript eher was, wenn man Webseiten „programmiert“? Arduino und Raspberry PI 3B habe ich, arduino wahr glaub ich ein Uno. Sachen zum löten habe ich auch, mein Vater hat mir sein altes Lötgerär gegeben, und mit Elektronik und sowas kennt er sich gut aus.

Beitrag von „grt“ vom 8. September 2021, 22:58

löten musst du gar nicht. ein steckbrett und eine (passende) handvoll bauteilchen reicht aus. uno ist ok, ein arduino nano wär auch ausreichend. der raspi und seine pins ist schon fortgeschrittener, und für den anfang eher overkill. aber interessant, um das einrichten und administrieren von linux per ssh/terminal übers netzwerk zu lernen. aber das wär noch ein anderes thema.

und ja, das zusammenspiel von html, css und javascript ist für webseiten gedacht.

du hattest ja mal auf die webseite geguckt, die ich im profil hab, das ist html, css, ein kleines bisschen php und (unfertiges) javascript. die fischchen die da herumgeistern, oder das gummibären sortieren ist mit javascript realisiert. nicht ganz fehlerfrei geb ich ja gerne zu, und auch noch nicht fertigprogrammiert das meiste...

ich schreib dir morgen mal eine pn mit mehr infos.

Beitrag von „Max“ vom 8. September 2021, 23:01

Ich denke das es am Anfang nicht wichtig ist, den besten Code zu schreiben, sondern Projekte auch irgendwann zu Ende zu bringen, und nicht immer ein neues Anfangen, und das alte nicht fertig zu machen

---

Beitrag von „grt“ vom 8. September 2021, 23:12

damit hast du auf jeden fall recht. und die projekte sollten auch zu dem jeweiligen lernstand passen - also nicht gleich eine künstliche intelligenz entwickeln wollen, bevor man überhaupt weiss, was eine variable ist 😉

wie gesagt, ick meld mir morgen im lauf des tages.

Beitrag von „ozw00d“ vom 9. September 2021, 07:29

sehr gut ist auch Codecadamy, bringt einen schritt für schritt weiter.

Beitrag von „krokol“ vom 9. September 2021, 12:38

Okay, da ich lange in dem Bereich Dozent war, nutze ich mal meinen 100. Beitrag hier und gebe auch mal meinen Senf hinzu. Wie andere schon erwähnt haben, sind HTML und CSS kein "programmieren", sondern beschreiben die Struktur und das Aussehen der Inhalte einer Webseite. Die Interaktivität kommt dann durch JavaScript (JS), PHP usw., mit denen diese Elemente auf der Seite nach bestimmten Regeln verändert werden.

Wichtig ist aber die Frage, was und warum du programmieren lernen möchtest und was am Ende herauskommen soll. HTML, CSS, JS und diverse "Frameworks": Damit kannst du zB Anwendungen und einfache Spiele entwickeln, die du dann leicht im Web veröffentlichen kannst. ABER du musst dazu erst mal viele unterschiedliche Teile lernen. Beim Raspberry ist es eher umgekehrt: Die Anwendungen, die man da entwickelt, sieht man oft gar nicht: die lesen irgendwelche Sensordaten aus und verarbeiten sie weiter, schalten die Beleuchtung an, wenn der Dämmerungssensor "dunkel" anzeigt etc. Und dann gibt es noch die Anwendungsentwicklung zB für Windows, macOS oder Linux oder die App-Entwicklung für iOS und Android. Alles unterschiedliche Bereiche, in denen oft unterschiedliche Werkzeuge und Programmiersprachen genutzt werden.

Zunächst geht es beim Programmieren aber erst mal darum, die grundlegenden Elemente zu beherrschen: Variablen, Schleifenkonstrukte, Bedingungen, Ein-/Ausgabeanweisungen etc. Beherrscht man das mal richtig, dann kann man sich auch leichter in neuere Programmiersprachen und Werkzeuge einarbeiten. Denn in der Industrie wird oft mit ganz anderen Programmierwerkzeugen gearbeitet.

Für den Einstieg würde ich bei verfügbarem Mac/iPad das kostenlose "**Swift Playgrounds**" empfehlen - da klickt man die Anweisungen zusammen und muss verschiedene Aufgaben/Szenarien lösen, indem man eine vorhandene Spielfigur auf einem vorhandenen Spielfeld dazu anweist, Diamanten einzusammeln und Schalter zu betätigen. Vorteil: kostet nix, kein Vorwissen erforderlich, wenig Frust durch Syntaxfehler (also Tippfehler, die dein Programm für den Computer unlesbar machen). **Swift** ist die Programmiersprache, die Apple auch für die App-Entwicklung unter iOS und macOS empfiehlt und theoretisch kannst du von

Swift Playgrounds zu Xcode wechseln und dann in die App-Entwicklung einsteigen (die Hürden bleiben aber erst mal hoch).

Anschließend würde ich mir **Thonny** anschauen: Das ist eine kostenlose, interaktive Entwicklungsumgebung für **Python**. Diese Programmiersprache wird auch viel auf dem Raspberry benutzt und mit Thonny hat man so was ähnliches wie den Basic-Interpreter, den die älteren von uns noch vom C64 kennen: Thonny öffnen und lostippen, dann auf "Ausführen" klicken, sich wundern, Fehlermeldung analysieren, Code verbessern und von vorne. Toll daran ist, dass du dir dein System nicht vermurkst, denn wenn du Python normal installierst, musst du auf Kommandozeile arbeiten und kannst dir dein System zerschießen, wenn du Versionen und Erweiterungspakete installierst, die mit den systemeigenen nicht problemlos zusammenspielen. Für Python gibt es zahllose Kurse und Beispiele im Web, vieles auch kostenlos. Und die Einsatzgebiete reichen von der Entwicklung eines einfachen "Hello World" über Kommandozeilen-Tools bis zu Neuronen Netzen, Sprach- & Signalverarbeitung oder eben auch dem Verarbeiten von Sensordaten zB eines Raspberry Pi.

Hast du dich mal mit Swift Playgrounds und Thonny/Python beschäftigt, wirst du schon viel besser wissen, was du zukünftig programmieren willst - und dann suchst du dir die richtige Programmiersprache, die richtigen Werkzeuge und die richtigen Libraries/Frameworks (quasi Sets von Lego-Bausteinen für Programmierer, die bestimmte Zwecke erfüllen) für den Job aus.

Hope that helps - Geld würde ich am Anfang keines ausgeben, gerade nicht für Kurse. Das lohnt sich erst, wenn du weißt, in welche Richtung du weitergehen willst.

Beitrag von „Si Vis Pacem“ vom 9. September 2021, 15:19

Meine 2¢

Harvards CS50's Introduction to Computer Science An introduction to the intellectual enterprises of computer science and the art of programming.

> Zitat

> ## About this course
>
> Harvard University's introduction to the intellectual enterprises of computer science and the art of programming for majors and non-majors alike, with or without prior programming experience. An entry-level course taught by David J. Malan, **CS50x** teaches students how to think algorithmically and solve problems efficiently. Topics include abstraction, algorithms, data structures, encapsulation, resource management, security, software engineering, and web development. Languages include C, Python, SQL, and JavaScript plus CSS and HTML. Problem sets inspired by real-world domains of biology, cryptography, finance, forensics, and gaming. The on-campus version of **CS50x** , CS50, is Harvard's largest course.

Beitrag von „MacPeet“ vom 9. September 2021, 17:46

Ich finde es nicht ganz fair, HTML, CSS, JS und PHP als "ist kein Programmieren" abzuwerten, nur weil es Browser-Anwendungen sind.

Es gibt ja schließlich auch dafür Experten, die nur reine Browser-Anwendungen schreiben, ohne jemals eine Codezeile Basic oder C oder sonst was geschrieben zu haben und damit noch gut Geld machen.

Als ich damals noch Gewerbe hatte und Software entwickelte, die ich natürlich auch verkaufen wollte und habe, musste ich mich zwangsläufig mit dieser Sache beschäftigen und ich empfand dies nicht weniger Aufwendig, als die Software selbst zu entwickeln.

Ok, HTML und CSS ist die Grundlage, aber auch die muss man erst einmal verstehen und richtig einsetzen, wenn's gut werden soll.

Ich habe damals mittels PHP meine ganze Kaufabwicklung automatisiert, Datenbank angebunden, mit automatischer Rechnungserstellung als PDF und automatischen eMail-Versand an den Kunden und an mich in Kopie und dies alles im Browser.

Ohne dies hätte ich wohl sehr viel Arbeit gehabt, um je auch nur eine Software zu verkaufen, aber mit diesem zusätzlichen Aufwand lief alles von selbst.

Und dabei war ich ganz sicher noch unter 10% von dem was diese Sprachen tatsächlich noch alles können.

Für mich war auch dies tatsächlich Programmieren, wie schon die Entwicklung der Software selbst.

Mein Hinweis an den TE ist folgender, wenn man so ganz bei Null anfängt und nicht gerade Informatik oder Webdesign studiert hat.

Für jede Sprache, egal ob nun Web- oder App-Entwicklung gibt es viele Hilfeseiten, von denen hier ja schon einige genannt wurden.

Oft findet man dort viele Code-Beispiele, welche helfen, die Grundlagen besser zu verstehen.

Man muss sehr viel lesen und auch dafür tatsächliches Interesse mitbringen, wenn man bei Null anfangen will, aber es geht, wenn man nur will.

Im Fall der Code-Beispiele einfach mal eine Sache ändern und das Ergebnis anschauen, dann... aha, ok, dies hat meine Änderung nun bewirkt.

Was man mal selbst gemacht hat bleibt einfacher hängen und so kommen immer mehr Erkenntnisse dazu.

Von Null lernen ist ein langer Weg mit viel Ausdauer und Geduld.

Ich finde Dein Vorhaben aus Post#1 erst einmal prima, wenn echtes Interesse dahinter steht, denn ich selbst habe mal so angefangen.

Allerdings fang mal klein an! Spiele sind da eine ganz andere Hausnummer, denke ich. Richtig gute Spiele mit vernünftiger Grafik und auch etwas Anspruch werden nicht von Einzelpersonen entwickelt.

Das ist mitunter ein richtig langer Weg zum Erfolg.

Müll diesbezüglich gibt's schon genug im Appstore.

Beitrag von „Max“ vom 9. September 2021, 18:34

MacPeet ich bin 14, da werde ich wohl das noch nicht studiert haben

---

Beitrag von „Obst-Terminator“ vom 9. September 2021, 18:50

B4tMaxt kann keiner wissen das du erst 14 bist. „Nicht volljährig“ ist ne große Range.

Beitrag von „MacPeet“ vom 9. September 2021, 19:05

B4tMaxt

...prima, dann sind meine Hinweise doch gut, durch Eigeninteresse diese Dinge zu lernen und später dann mit gemachten Erfahrung vielleicht mit Fachrichtung ins Studium gehen.

Beste Voraussetzungen dafür, wenn man sich schon früh damit beschäftigt hat.

Beitrag von „grt“ vom 9. September 2021, 20:03

> Zitat von MacPeet
>
> HTML und CSS ist die Grundlage, aber auch die muss man erst einmal verstehen und richtig einsetzen, wenn's gut werden soll

das ist vollkommen richtig, und abwerten wollte ich ganz bestimmt nix. sorry, wenns so rübergekommen sein sollte. aber HTML und CSS sind keine "programmiersprachen" und eine statische webseite aus den beiden ist kein "programm". das meinte ich.

webprogrammierung wär dann mit php, jscript etc.. scriptsprachen, was aber nicht heisst, dass sie jetzt irgendwie schlechter sind, oder weniger "wert", oder weniger wissen und verständnis des programmierers benötigen.

Beitrag von „Elektrohorst“ vom 9. September 2021, 20:12

In dem Alter macht man normalerweise in der Informatik AG, für die interessierten scratch: https://scratch.mit.edu/studios/3579538/

und geht danach auf html, oder in der Oberstufe vielleicht sogar zu java.

Beitrag von „grt“ vom 9. September 2021, 20:37

naja... "in dem alter" ist schon ganz schön pauschal finde ich. und die verlinkten projekte sind auch noch recht "kindlich".

so, wie ich den max hier im forum erlebt habe, ist er interessiert, engagiert, und hat ein offensichtlich gutes verständnis für it-krams, ich finde durchaus überdurchschnittlich in bezug auf sein alter. und wenns eine brauchbare informatik-ag in der schule gäbe, könnte ich mir vorstellen, dass er da längst teilnehmen würde. scheint aber wohl nicht der fall zu sein.

Beitrag von „Moorviper“ vom 9. September 2021, 23:41

Ich würde mir

https://de.aliexpress.com/item...22%3A%2266468842330%22%7D

oder so ein Arduino Set holen

https://de.aliexpress.com/item...22%3A%2266499224078%22%7D

Bei der Arduinoprogrammierung fängt man schön bei low level c an.

Es gibt online vile how to's welche sich mit solchen starter kits befassen.

Beitrag von „mhaeuser“ vom 10. September 2021, 09:01

krokol Ich habe bis auf die üblichen Verdächtigen "Java" und "Arduino" keine besonderen Gegenvorschläge im Sinn, aber warum gerade Swift und Python - Sprachen, die dafür bekannt sind, ohne vernünftige Begründung alles anders als in anderen Sprachen zu machen? Ich habe schon gesehen, wie Python-Einsteiger C schreiben und da kommt mir die Galle hoch.

Beitrag von „Aluveitie“ vom 10. September 2021, 09:30

Das eine ist die Sprache, das andere die Leute die eine Sprache anzieht. Was gerade bei Python viele sind, die sich nicht unbedingt für das Handwerk Programmierung interessieren...

JavaScript oder PHP sind OK um mit Webdesign mal die Grundkonzepte zu lernen und dann mit einer anderen Sprache wie C, C++ oder Java erst richtig einzutauchen.

Beitrag von „Elektrohorst“ vom 10. September 2021, 10:49

Ich wollte mit der Scratch Empfehlung keines Falls das Talent von Max irgendwie in Frage stellen, ich wollte lediglich zeigen, wo heute in der Schule Informatik anfängt. Und da machen wirklich alle Lehrer, die ich kenne, zu Anfang Scratch. Danach wird gerne der Lego EV3 Roboter angeboten. (Der Arduino ist selbstverständlich deutlich billiger und komplexer). Da hört sich alles sehr "kindlich" an, ist aber weitaus komplexer als es scheint aber die Kids haben so schneller ein Erfolgserlebnis. Deshalb wird dann auch schnell auf html übergegangen, einfach weil es schnell begreifbar ist und man schneller ein Ergebnis hat.

Fängt man gleich, ohne Hilfe, mit Java oder C++ an, ist der Frust bei den Kids schnell vorprogrammiert und der Spaß am programmieren verfliegt. Das alles ist aus der Perspektive eines Vaters mit einem Sohn im gleichen Alter geschrieben.

Beitrag von „grt“ vom 10. September 2021, 11:22

> Zitat von Elektrohorst
>
> Fängt man gleich, ohne Hilfe, mit Java oder C++ an, ist der Frust bei den Kids schnell vorprogrammiert

das kommt darauf an, wie man das gestaltet. terminalanwendungen bis zum irrewerden ist natürlich nicht besonders attraktiv (kann man aber auch spannend rüberbringen, bei entsprechendem pädagogischem engagement), aber ein steckbrettchen, zu anfang eine kleine einführung, was bauteilchen wie leds und widerstände etc. so machen, und dann per arduino eine oder mehrere leds morsecodes, lauflichter oder ampeln produzieren lassen, tipptasterchen dazu, vielleicht noch ein poti (aber auf keinen fall ein "starterset" mit ohne ende sinnlosen und qualitativ indiskutablen chinasensoren...) ... das kann richtig spass machen, und "nebenbei" gibts die essentiellen grundlagen der programmierung dazu, worauf dann, wenn das interesse weiter besteht aufgebaut werden kann.

Beitrag von „Elektrohorst“ vom 10. September 2021, 11:57

Genau das war das Konzept des Adventskalenders bei Conrad Electronic, vor ungefähr drei Jahren. Es gab einen Arduino und jeden Tag neue Teile und eine neue Programmieraufgabe. Mein Sohn war damals 14 und ziemlich überfordert. Aber so hab ich mich auch noch mal mit der Arduino Programmierung beschäftigen müssen, was ich auch gerne gemacht habe. Ich will damit nur sagen, hat man jemanden, der einem helfen kann, ist es deutlich leichter. Ich glaube, Conrad hat den Kalender wieder aus dem Sortiment genommen. Bin mir aber nicht ganz sicher, falls nicht, wünsch ich mir einen dieses Jahr.

Beitrag von „grt“ vom 10. September 2021, 12:21

> Zitat von Elektrohorst
>
> jeden Tag neue Teile und eine neue Programmieraufgabe

lass mich raten... das waren genau die erwähnten chinasensoren - abstandsmesser, der hausnummern misst, servomotoren, die schon im ruhezustand zuviel strom ziehen, höchst ungenaue winkel produzieren etc. - ich kenn das zeugs, und kenn auch die hübschen beispiele

aus den bibliotheken.

destawegen sagte ich auch "eine handvoll einfache bauteilchen und ein steckbrettchen". damit kann man nämlich einfache prograämmchen realisieren, den funktionsumfang des arduinos kennenlernen - digitale/analoge pins, als ein- oder ausgang, wie geht man damit um - schreiben/schalten oder abfragen/überwachen.. da reichen eine led, ein vorwiderstand und ein tipptaster, und für analoge pins noch ein 10K poti aus, und alles kombiniert - zack kann man auch gleich noch bedingungen im programm lernen - abfrage eines ereignisses, und reaktionen darauf programmieren. lässt sich später prima auf komplexere sachen anwenden und erweitern.

und programmieren ist nun mal nicht klick&bunt, sondern fieser kryptischer code in texteditoren, mit gemeinen spitzfindigen syntaxvorgaben, hyperpingeligen und kleinkarierten compilern, die dauernd herummeckern etc... wer wirklich interessiert ist, der findet sich damit ab, und findet das noch nicht mal schlimm 😉

Beitrag von „loeffla“ vom 10. September 2021, 12:43

> Zitat von grt
>
> in jedem fall bleibst du im dunstkreis des browsers, der webseiten stecken.

JavaScript an sich als auch das JavaScript "Eco-System" ist weitaus mehr als bloß ein Dunstkreis aus Browsern und Websites.

> Zitat von grt
>
> ist aber durchaus spannend (zumindest, wenn man mit javascript browserspielchen oder -anwendungen zusammenstrickt, php kann ich nicht leiden..), aber "richtig" programmieren ist noch mal was anderes.

Nichts für ungut und es keineswegs böse meine, bezüglich JavaScript scheint dein Wissen jedoch auf einem Stand vor ~10 Jahren stehengeblieben zu sein. JavaScript ist längst nicht mehr das "Kleinkind", welches nur im Sandkasten (Browser) mit Schippe und Förmchen spielt. Dank Runtimes, Frameworks und Bibliotheken wie Node.js, Angular, React, Vue (samt Nuxt) oder Electron auch außerhalb des Browsers zunehmend an Beliebtheit gewinnt. Egal ob Editoren wie Atom oder VS Code, ob Messenger wie Discord oder die Desktop-App von Spotify - in diesen Anwendungen mischt JavaScript mit.

> Zitat von grt
>
> java, oder python - komplexer, als javascript, aber auch um längen vielseitiger und nicht ausschliesslich ans web gebunden.

Dieser Aussage hätte ich damals noch zu gestimmt, sie ist aber inzwischen obsolet. Mittlerweile hat JavaScript mehr auf dem Kasten, als z.B. die "Manipulationen" des DOM's. Angefangen bei Client- oder Server-seitigem Gedöns im Webbereich, bis hin zu nativen und Plattform-übergreifende Applikationen. An Komplexität mangelt es bei JavaScript auch nicht, da die Einsatzmöglichkeiten längst deutlich vielseitiger sind.

LG loeffla

Beitrag von „Aluveitie“ vom 10. September 2021, 13:02

Dass JavaScript mittlerweile auch für vieles mehr verwendet werden kann heisst nicht, dass das eine gute Idee ist. JavaScript ist nun halt mal der Standard im Web, aber für mehr würde ich diese Sprache nie verwenden wollen (wenn überhaupt).

JavaScript wurde schnell in 2 Wochen entwickelt, und das zeigt sich überall. PHP ist da genau so schlimm. Wieso dann eine schlecht designte Sprache noch dort verwenden, wo man gar nicht dazu gezwungen ist? Ohne statisches Typensystem ist jedes Grossprojekt eine tickende Zeitbombe in der Wartung. Das zeigt sich auch darin, dass die meisten neuen Scriptsprachen welche JavaScript ersetzen wollen statisch sind (TypeScript, Dart).

Beitrag von „grt“ vom 10. September 2021, 13:07

> Zitat von loeffla
>
> bezüglich JavaScript scheint dein Wissen jedoch auf einem Stand vor ~10 Jahren

> stehengeblieben zu sein

nö... kann man so nicht sagen. in der letzten zeit hab ich nix mehr mit web gemacht, das ist richtig, mir ist aber durchaus bewusst, das jscript nicht nur dort angewendet werden kann, und von den erwähnten frameworks hab ich auch schon läuten hören, das ein oder andere sogar auch mal ausprobiert, allerdings nicht weiter vertieft...

hier geht es aber um den einstieg in programmieren allgemein, vorkenntnisse sind so gut wie nicht vorhanden. will heissen, von "spezialformen", frameworks etc. weit weit entfernt. um sich damit erfolgreich auseinandersetzen zu können, brauchts grundlagen (in bezug auf die frameworks z.b. javascript an sich beherrschen).

Beitrag von „krokol“ vom 10. September 2021, 13:12

mhaeuser Ganz einfach: Weil wir hier in einem Hackintosh-Forum sind und ich versucht habe, Möglichkeiten aufzuzeigen, die mit einem Mac kostenlos und unkompliziert auch von einem kompletten Anfänger genutzt werden können. Ich habe auch nicht Swift empfohlen, sondern Swift Playgrounds, weil man da nicht erst komplexe Syntax und APIs lernen muss, sondern rasch Erfolgserlebnisse hat. Ebenso Python nicht per homebrew, sondern Thonny - auch da wegen der einfachen Nutzung.

Wenn man mal genügend Erfahrung gesammelt hat, kann man dann (abhängig von den eigenen Interessen) die passenden Sprachen, Frameworks und Werkzeuge wählen. Thread heißt aber "PROGRAMMIEREN LERNEN, WO MIT SOLLTE ICH ANFANGEN?" und nicht "Best ever Universal Programming Language wanted". Aber viele "alte Hasen" haben ihre ersten Schritte mit Commodore Basic oder Turbo Pascal oder gar Cobol und Fortran gemacht - sicher aus heutiger Sicht keine optimalen Sprachen, aber gerade die ersten beiden waren insofern "anfängerfreundlich" als man rasch Resultate sah.

Zweifellos würde mein Fazit aber anders aussehen, sollte ich konkret für ein Projekt eine sinnvolle Toolauswahl empfehlen, wenn das programmiertechnische Grundwissen in einer oder mehreren Sprachen vorhanden sind.

Beitrag von „grt“ vom 10. September 2021, 13:22

hier wird immer wieder von "erfolgserlebnis" gesprochen, und in dem zusammenhang davon abgeraten sich mit

> Zitat von krokol
>
> komplexe Syntax

etc. zu befassen. zum einen ist "komplexe syntax" nun mal ein charakteristikum von programmiersprachen, damit MUSS man sich auseinandersetzen, wenn man programmieren ernsthaft lernen will, und es könnte genauso eine böse enttäuschung sein, wenn sich erst mit klick&bunt die ersten erfolgserlebnisse "erkauft" wurden, und dann der "absturz" in form von syntax lernen müssen folgt.

und warum ist es denn eigentlich kein erfolgserlebnis, wenn man das ergebnis eines selbst- (unter anleitung/ nach tutorial) geschriebenen codes - das funktionieren desselben im terminal - beobachten kann? weils nicht bunt ist und nicht wackelt? 😉

Beitrag von „loeffla“ vom 10. September 2021, 13:50

> Zitat von Aluveitie
>
> JavaScript ist nun halt mal der Standard im Web, aber für mehr würde ich diese Sprache nie verwenden wollen (wenn überhaupt).

Ob, wann und wofür man selber JavaScript einsetzt und ob das letzten Endes eine gute Entscheidung wäre, ist ein Thema für sich. Mir ging hier vorrangig um die Aussage von grt , das JavaScript nur im Dunstkreis von Browsern und Websites eingesetzt wird. Weil dem nicht so ist, hier kurz ein paar Beispiele aufgezählt habe.

> Zitat von Aluveitie
>
> JavaScript wurde schnell in 2 Wochen entwickelt, und das zeigt sich überall. PHP ist da genau so schlimm. Wieso dann eine schlecht designte Sprache noch dort verwenden, wo man gar nicht dazu gezwungen ist? Ohne statisches Typensystem ist jedes Grossprojekt eine tickende Zeitbombe in der Wartung. Das zeigt sich auch darin, dass die meisten neuen Scriptsprachen welche JavaScript ersetzen wollen statisch sind

> (TypeScript, Dart).

Ich stimme dir in Teilen durchaus zu und hier JavaScript nicht als Eier-legende Wohlmilchsau anpreise. Ganz im Gegenteil - die Wahl der "richtigen" Sprache ist vorrangig vom Projekt selbst und dem eigenen Know-how abhängig. Natürlich gibt es Programmiersprachen, die Probleme mit sich bringen (können) oder selber erst mit dem Anspruch entstehen, etwas besser zu machen als andere Sprachen davor.

> Zitat von grt
>
> nö... kann man so nicht sagen. in der letzten zeit hab ich nix mehr mit web gemacht, das ist richtig, mir ist aber durchaus bewusst, das jscript nicht nur dort angewendet werden kann

Dass dir auch andere Einsatzmöglichkeiten bekannt sind, konnte ich aus deinem vorigen Post und der Aussagen so nicht herauslesen. Daher meine Kritik, das "JS" mehr als im Dunstkreis von Browser und Websites Verwendung findet. Meine Einwende also kurz geäußert habe, da der aktuelle Stand von JavaScript (ECMAScript) doch etwas anders aussieht.

Beitrag von „krokol“ vom 10. September 2021, 14:05

grt Ich gehe davon aus, dass du wenig Erfahrung in der Erwachsenenbildung hast? Es geht darum, Anfänger erst mal bei der Stange zu halten. Warum nutzen Leute Legosteine, wo man mit einer vernünftig ausgestatteten Werkstatt doch alles selbst und viel besser bauen und auch noch selbst konstruieren kann? (Gibt sogar Fernsehsendungen dazu: Lego Masters). Am Anfang hat man weder die Werkzeuge noch die Erfahrung. Später kaufe ich vielleicht eine Fräse, eine Drehbank oder einen 3D-Drucker und finde dann Lego eher primitiv. Aber eben erst später. Natürlich erlernt man irgendwann die vollständige Syntax, wenn das entsprechende Interesse geweckt und ggf ein eigenes Projekt vorhanden ist. Aber eben auch erst später.

Zumal: wir reden bei Syntax doch nicht nur von den "paar" Anweisungen der Programmiersprache selbst (damit kriegst du doch heute nicht mal mehr eine Bildschirmausgabe hin, erst recht keinen Touch-Input), sondern auch die der tausenden API-Calls, Frameworks etc. Willst du jemanden vergraulen, dann zwingst du ihn, erst mal vier Wochen die Reference-Manuals der Sprache und der verwendeten Frameworks zu lesen, bevor er auch nur eine Zeile programmieren darf. Danach darf er als Neuling im kommerziellen Umfeld eine (!) Funktion in einem riesigen Programmkomplex überarbeiten (!), die ihm dann

von der QS vier Mal zurückgeschmissen wird, bis sie akzeptiert wird. Wenn er dafür Geld bekommt, wird er das akzeptieren, aber dann ist er kein Neuling im Programmieren, höchsten in der Sprache, der Entwicklungsumgebung und/oder dem Projekt.

Wenn es sich aber um einen Jugendlichen handelt, der das als Hobby macht? Da solltest du dafür sorgen, dass der schon nach ner Stunde, besser nach ein paar Minuten, die ersten kleinen Erfolge sieht. Und da gibt es keine QS oder einen anderen Coder im Team, die Feedback geben, sondern nur ein einsames "Error" wegen vergessener oder überzähliger Semikolons, Klammern, Gleichheitszeichen, Parameteranzahlen, falschen Typen o.ä. Wenn man dann auch noch mit Anwendungen aus einem Tutorial beginnt, die man abwandeln möchte, und die aus 200+ Zeilen bestehen, wird das schnell frustrierend.

Nur weil jemand mit Lego oder Fischertechnik begonnen hat, heisst das nicht, dass er nicht heute Bandsägen, Schweißgeräte und Stehbohrmaschinen nutzt.

Beitrag von „grt“ vom 10. September 2021, 14:25

> Zitat von krokol
>
> Ich gehe davon aus, dass du wenig Erfahrung in der Erwachsenenbildung hast?

wie kommst du darauf?

> Zitat von krokol
>
> Da solltest du dafür sorgen, dass der schon nach ner Stunde, besser nach ein paar Minuten, die ersten kleinen Erfolge sieht.

geht super mit dem erwähnten arduino. und hat sowohl code, als auch bunt. und ich werde niemanden dazu zwingen erstmal trocken zu fuss code zu lesen, aber ebensowenig werde ich mit hübsch&bunt einen ponyhof vorgaukeln, und verschweigen, dass auch im ponyhof der alltag mehrheitlich aus mistschippen besteht...

und ja, ich hab erfahrung in der erwachsenenbildung, und versuche dort einen gesunden mittelweg aus trockenem formel- und syntaxkram und "anfassenkönnen" der ergebnisse dessen zu finden.

Beitrag von „Elektrohorst“ vom 10. September 2021, 16:46

Mit 14 ist man nicht erwachsen, man ist mit 14 ein Teenager und geht in die 8te oder 9te Klasse. Bei Mathe fängt man übrigens auch nicht mit Trigonometrie an, weil sie so schön ist, sondern mit banalen Zahlen.

Beitrag von „MacPeet“ vom 10. September 2021, 17:58

Naja, eigentlich sehr banal gesagt. Wissensdurst ist doch keine Frage des Alters.

Der TE zeigt sich hier im Forum doch sehr interessiert und für sein Alter doch recht kompetent in einigen Sachen und mit seinem Interesse dazu hat er meine volle Unterstützung.

Ich finde es gut, dass er Interesse dafür zeigt. Es gibt viel zu viele Jugendliche, die Stundenlang (24/7) vor irgendwelchen Games sitzen, was letztendlich nach hinten raus nix bringt.

In dem Alter habe ich lieber an Mädels rumgefummelt, was mehr Sinn machte und auch erste, aber damals noch geringe Erfahrungen mit Computern gemacht.

Die Jugendlichen heute sind uns ohnehin weit voraus, im Vergleich zu damals.

Es gibt ja auch Leute, welche in Deutsch eine Null sind/waren und trotzdem auf anderen Gebieten besondere Talente zeigen.

Einige Nullen in der Schule sind heute Firmen-Chef's größerer Firmen, nur weil sie in einer Fachrichtung Talent gezeigt haben und Ideen hatten.

Ideen kann man auch mit 14 haben.

Hier im Thread ging es ums Programmieren lernen, womit man doch nicht früh genug anfangen kann, wenn man dafür Interesse zeigt.

Was draus wird... ? ...kann ich nicht sagen, liegt ja an jedem selbst.

Es ist auch erst einmal egal, welche Sprache man sich ansieht aus Interesse, es wird ohnehin nicht bei nur einer Sprache bleiben, denke ich.

Beitrag von „Max“ vom 10. September 2021, 18:07

Ich werde wohl erstmal mit HTML anfangen, mein Plan ist also erstmal HTML und dann CSS und dann kann ich ja mal gucken.

Beitrag von „Elektrohorst“ vom 10. September 2021, 18:16

Gute Entscheidung. Viel Erfolg!

Beitrag von „grt“ vom 10. September 2021, 18:41

> Zitat von Elektrohorst
>
> Bei Mathe fängt man übrigens auch nicht mit Trigonometrie an

das - aufs programmieren bezogen - hab ich auch nie befürwortet.

guck dir mal das "blink" beispiel aus der arduino-ide an. ist supereinfach, trotzdem code, und (je nach led) auch noch bunt. und lässt sich schritt für schritt erweitern, und zwar so, dass variablen vorkommen, und auch der sinn davon selbsterklärend ist, funktionen, funktionsaufrufe und schleifen kann man unterbringen, parameterübergabe, evtl. sogar rückgabewerte, und all das in einfachem und übersichtlichem code, mit direkter rückmeldung vom steckbrettchen. das würde ich durchaus als

> Zitat von Elektrohorst
>
> mit banalen Zahlen

anfangen beschreiben.

Beitrag von „Raptortosh“ vom 10. September 2021, 19:00

> Zitat von MacPeet
>
> Naja, eigentlich sehr banal gesagt. Wissensdurst
>
> Hier im Thread ging es ums Programmieren lernen, womit man doch nicht früh genug anfangen kann, wenn man dafür Interesse zeigt.

True. Kenne einen der schon sehr früh damit angefangen hat, und jetzt in der Schule (bin 16) deutliche Vorteile gegenüber uns anderen hat, und vieles schon viel früher kann als andere, und auch Sachen kann die "wir anderen" vermutlich nie so lernen werden (außer man bringt es sich selbst bei, wozu aber viele kein Interesse haben(gibt viele mit der Einstellung "nur das nötigste" wobei für einige davon das noch zu viel ist..)). War schon am Anfang so, und ist es auch jetzt noch, dass der Sachen kennt, die andere noch nie gehört haben.

Würde definitiv dazu raten früh mit programmieren anzufangen, egal ob du mal was in die Richtung machen willst, oder nicht. B4tMaxt

---

Beitrag von „Elektrohorst“ vom 10. September 2021, 19:44

grt Arduino ist super, keine Frage.

Beitrag von „Sascha_77“ vom 10. September 2021, 20:01

> Zitat von B4tMaxt
>
> Ich werde wohl erstmal mit HTML anfangen, mein Plan ist also erstmal HTML und dann CSS und dann kann ich ja mal gucken.

Ich (Baujahr 1977) habe damals mit einem HTML4 Buch (das liegt hier sogar noch irgendwo rum) angefangen und meine erste Webseite auf einem Amiga erstellt mit nem nackten Text Editor. Wenn ich mir die Seite heute anschaue ist die natürlich furchtbar im Quadrat aber

damals (so 1998 rum) war ich total stolz drauf im Netz eine Präsenz zu haben. Irgendwann später habe ich dann mit Bash angefangen zu Scripten und habe da bis heute eine Art "fetisch". Wenn irgendeine Problemstellung anliegt überlege ich als erstes wie man das automatisieren kann und tüftel solange rum bis es klappt. Wiederum später kam dann Swift dazu mithilfe dessen ich schon einige Apps in Xcode gebaut habe.

Von daher ist das was du vorhast ein guter Start und der richtige Weg in meinen Augen.

Beitrag von „Max“ vom 10. September 2021, 20:47

Was für ein Buch würdet ihr empfehlen für html/css

Beitrag von „Sascha_77“ vom 10. September 2021, 20:52

Also wenn Du es in "physischer" Form lernen willst und nicht bei z.b. W3Schools online kann ich das Buch von bhv empfehlen. Ich hatte zwar nur das für HTML 4 aber ich denke das 5er hier ist genauso gut beschrieben. Und die Beschreibung ist wirklich sehr verständlich formuliert. Ebenfalls auch gut bebildert. Sprich das ideale Anfängerbuch.

https://www.amazon.de/HTML-bhv…omas-Kobert/dp/3826681878

Die haben bei Amazon wohl nur ein gutes Gebrauchtes aber für 4,13 Euro inkl. Versand ist das ja ziemlich schüler-budget-freundlich. 😉

Wegen CSS kann ich dir aber keine Empfehlung geben da ich mich damals nur mit HTML beschäftigt habe. Aber vllt. gibts aus dem bhv Verlag da auch was.

EDIT:

Wie wärs damit?

https://www.amazon.de/Cascadin...eets-bhv-Co/dp/382669368X

Beitrag von „Max“ vom 10. September 2021, 21:02

Und wenn ich das gut durch arbeite sollte ich dann HTML können

---

Beitrag von „grt“ vom 10. September 2021, 21:07

puh... bücherempfehlungen bzgl. html/css kann ich nicht geben, aber vielleicht wär ja der ein oder andere user mit entsprechenden kenntnissen hier bereit, dir beim durcharbeiten der werke ein wenig unter die arme zu greifen, in dem er/sie fragen beantwortet und/oder deine übungsprojekte mit dir durchgeht.

ich denke, dass anleitung und austausch verflixt wichtig ist, wenn man sich in ein neues fachgebiet einarbeiten möchte. allein im stillen kämmerlein fachbücher verspeisen kann, aber muss nicht lehrreich sein. austausch mit rückmeldung ist echt nicht zu unterschätzen.

und

> Zitat von B4tMaxt
>
> sollte ich dann HTML können

auslernen tut man nie, "können" ist ein dehnbarer begriff, und jede technik entwickelt sich weiter. können jetzt muss nicht können in 5 jahren bedeuten... und bücher lesen ersetzt auf keinen fall die praxis. anwenden (üben) musst du auf jeden fall auch

---

Beitrag von „Elektrohorst“ vom 10. September 2021, 22:50

Du kannst es kostenlos erstmal damit versuchen:

https://wiki.selfhtml.org/wiki/HTML/Tutorials/Einstieg

Als nächstes brauchst Du einen Code Editor.

Ich würde Atom: https://atom.io empfehlen.

Oder Brackets: https://brackets.io/?lang=de

Du schreibst Deinen ersten Code in dem Editor, speicherst ihn als

(irgendeinName.html) und öffnest diese Datei dann in Deinem InternetBrowser.

Wenn Du das geschafft hast, hast Du freie Bahn und kannst Deinen Code beliebig erweitern.

Falls das nicht klappt, einfach hier melden.

Und kauf bitte keinen Webspace, das brauchst Du erst, wenn Dein Projekt fertig ist. Nicht vorher.