SMART BEAST
Software Engineer

15分でわかる
ソフトウェア開発者になるための
ダイジェストガイド

www.shige-matsuoka.com

2

はじめに

このガイドを作ったきっかけ

飲み会で、初対面の方への自己紹介で、僕がシステムエンジニアであることを話すと、よく以下のような質問をされました。

・ソフトウェア開発者になるためにはどうすればよいでしょうか？
・技術的なスキルを学ぶにはどうすればよいでしょうか？
・フリーランスとして契約で仕事をとってくるのと正社員として給料をもらうのとどっちが良いのでしょうか？
・重要なスキルは何ですか？
・大学、プログラミングスクール、独学　どれが良いでしょうか？
・未経験者は難しいでしょうか？

システムエンジニアやプログラマーなどソフトウェア開発者の仕事が実際の現場で、具体的にどのような事を行っているのか？どうやって技術的なスキルを学んでいったのか？などの生の声って、あまり世間には知られていないのだと、、、

3

はじめに

このガイドを作ったきっかけ

今は、本屋でもプログラミングに関する書籍は多くあります。
１つのプログラミング言語をとっても、何十冊と書籍があります。

しかし、始める方法、成長する方法など「成功を掴むためにしっておく内容を１から１０まで教えてくれる」書籍や情報が、ほとんどありませんでした。

そのため、未経験者、初心者、ソフトウェア開発について学ぶことに興味がある人が、よくわからず、迷っているのだと思い、このガイドにソフトウェア開発者になるための道しるべをまとめました。

全体像を把握せず、無計画で学ぶと、遠回りをし無駄な時間、無駄な出費をすることになります。
私が学び始めた時は、情報も少なく、実際に、私は、遠回りをし無駄な時間、無駄な出費をしていました。
あなたが、そうならないように効率よく、学習できるように、このガイドが役立てれば、嬉しいです。

4 このガイドで学べること

このガイドで次のことが学べます

このガイドでは、次の事が学べます。

- なぜ？プログラミングを学ぶと良いのか？
  2030年までの傾向とメリット
- ソフトウェア開発者の仕事内容
- どのプログラミング言語を学べば良いか？
- ソフトウェア開発者が身につける重要なスキル
- プログラミングは、どのように学べば良いか？
- ソフトウェア開発者の３つの働き方パターン
- システム開発して、ようやくわかった本質

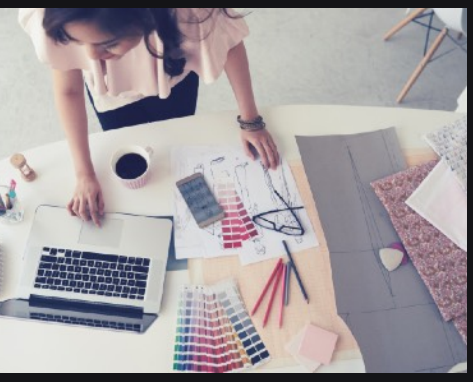
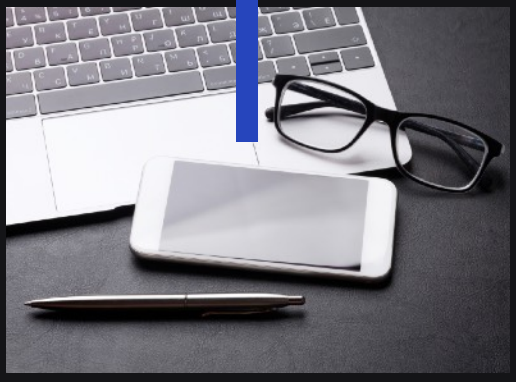
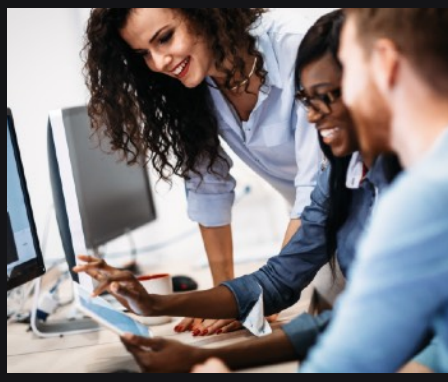
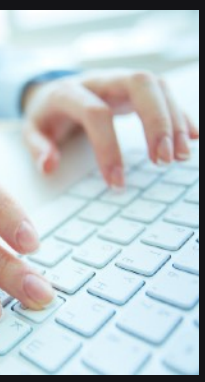
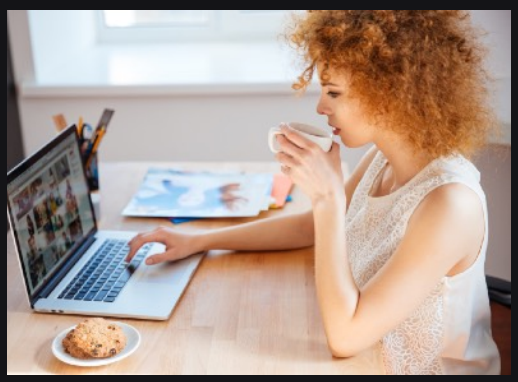
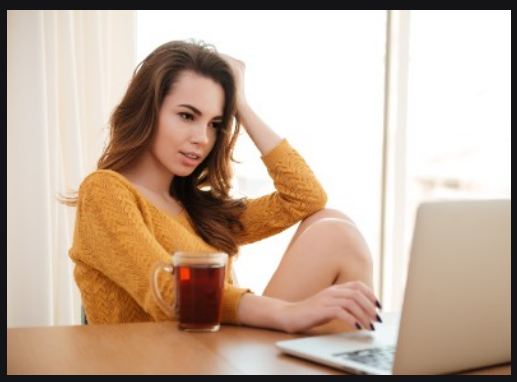
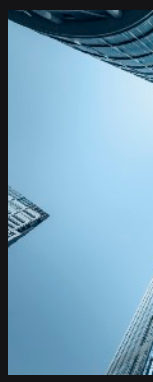
SMART BEAST
Software Engineer

なぜプログラミングを学ぶと良いのか？

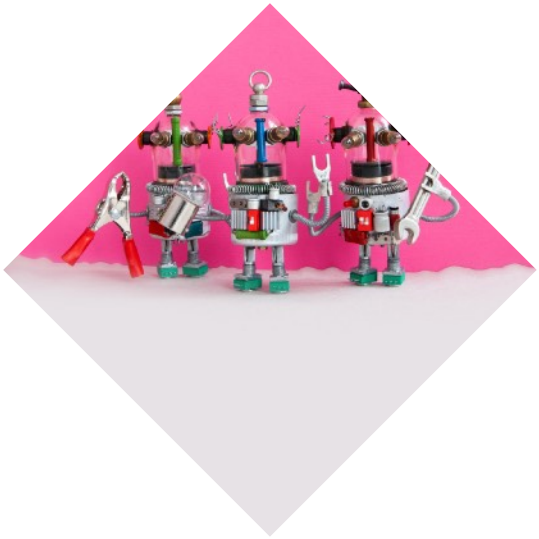
5

AIに使われる人、AIを使う人

この時代の変化にどういう心構えで何をしたかによって、５年先の生活が大きく変わります。

www.shige-matsuoka.com



6 なぜプログラミングを学ぶと良いのか？

AIに使われる人、AIを使う人

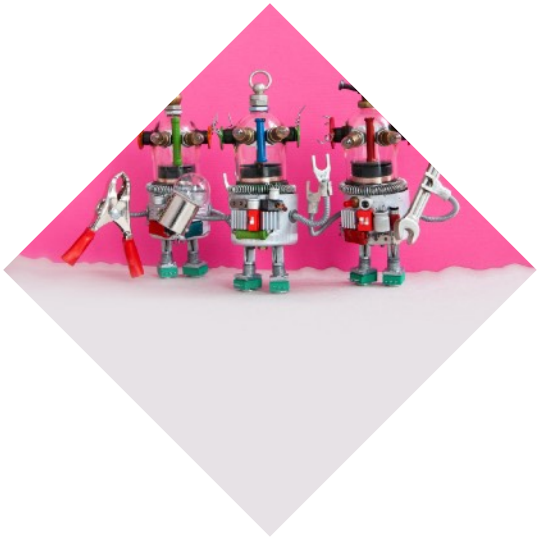
今、大量の計算はコンピューターが、行なっています。

ひと昔前のコンピューターが、まだ導入が進んでいない時は、人間が行なっていましたよね。
例えば、銀行など、手書きの伝票とお金を手動で計算し、１円合わなかったら、 帰れないなんて、話をよく聞きました。

今は、人間が数えたりする事はあると思いますが、ほとんどは、ATMなどコンピューターが人間の変わりに仕事を行なっています。

過去の実績から、今後、AIによって、人間の代わりをコンピューターが行う事は、避けて通れない流れではないでしょうか？

既に、最新のニュースでは、自動車の自動運転だったり、無人コンビニなど、 実用段階まできているようなので、AIが人間の代わりを行っている時代は、遠くありません。



7 なぜプログラミングを学ぶと良いのか？

AIに使われる人、AIを使う人

では、AIが浸透しても、必要な人って、どんな人だと思いますか？

「企画、発想、アート」や「コミュニケーション」、「AIを作ったり管理する人」はAIが発達しても、代わりになる事は難しいと言われています。

私もどんなにAIが発達しても人間の代わりは、絶対できない部分はあり「企画、発想、アート」や「コミュニケーション」は完全には代わりはできないと考えています。

なぜなら、コンピューターの一番、最小のところは、電気がONの１とOFFの０でしか成り立っていないからです。

そのため、いくら人間のように見せていても、完全に人間と同じにはならず、コンピューターが電気がONとOFFの表現である以上、生物ようように新しい生命を生み出すことはできないので、AIを作ったり管理するプログラマーは必ず必要になると思われます。

8 下りのエスカレータを登ろうとしていないですか？

逆はしんどい。

下りのエスカレーターを上まで登ろうとすると、しんどいです。
でも、普通に上りのエスカレーターを上まで上がる場合は、楽チンです。

仕事をしている業界によっては、少子高齢化によって、マーケット（市場）の需要より、供給が多く、下りのエスカレーターとなっているような業界があります。

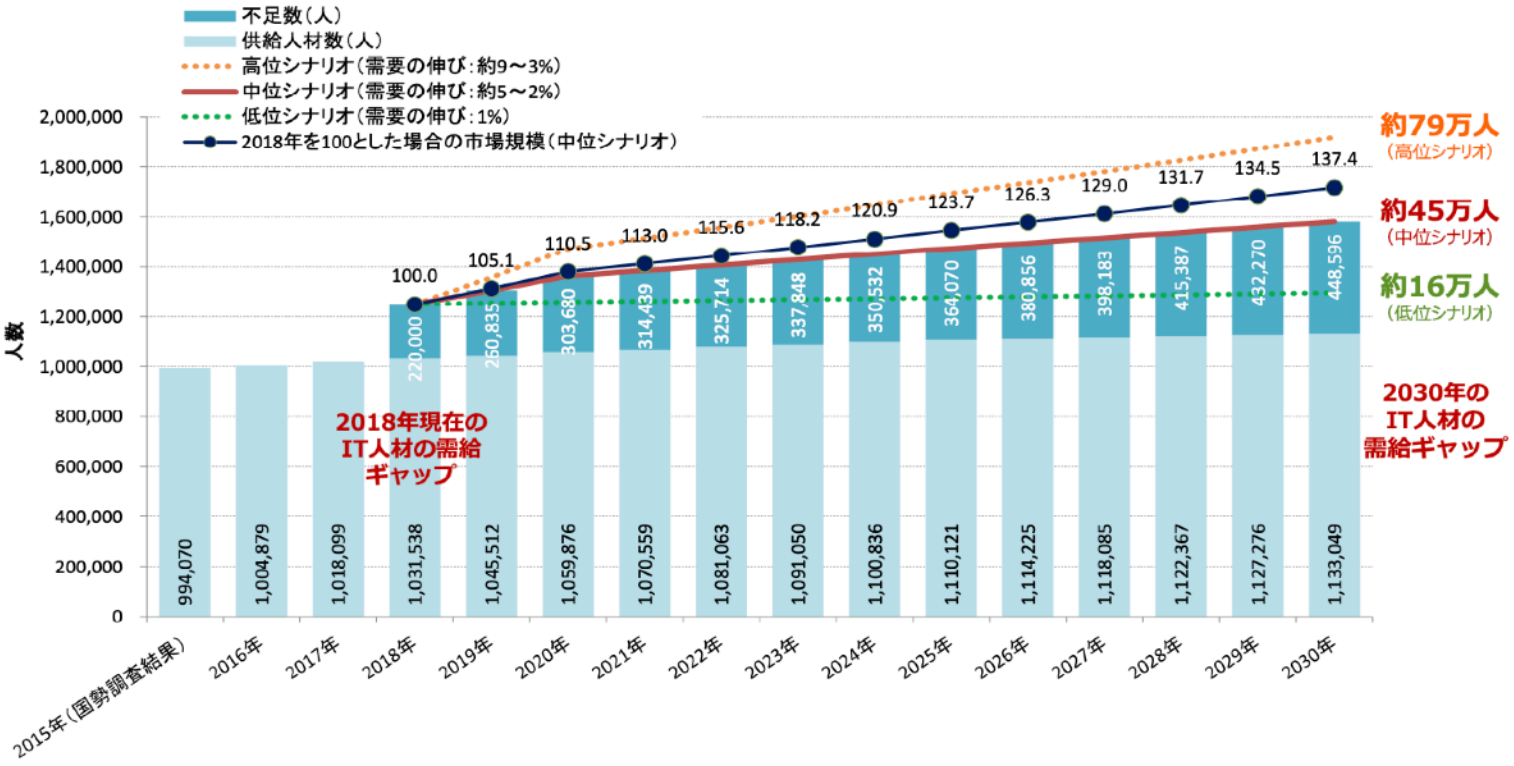
IT業界は既に人材不足です。国内のエンジニアでは、対応できず、中国やベトナムなどのアジアの国へ仕事を出しているのが現状です。

そして、小学校でもプログラミングの授業が始まります。
それほど、IT業界は人材不足となっています。
具体的にどれくらい、人材が不足しているか？について、
経済産業省調査した結果があります。

9 下りのエスカレータを登ろうとしていないですか？

逆はしんどい。

経済産業省調査した結果によると、2030年には、経済の需要の伸び率を中とした場合、約45万人が不足するという予測データとなっています。

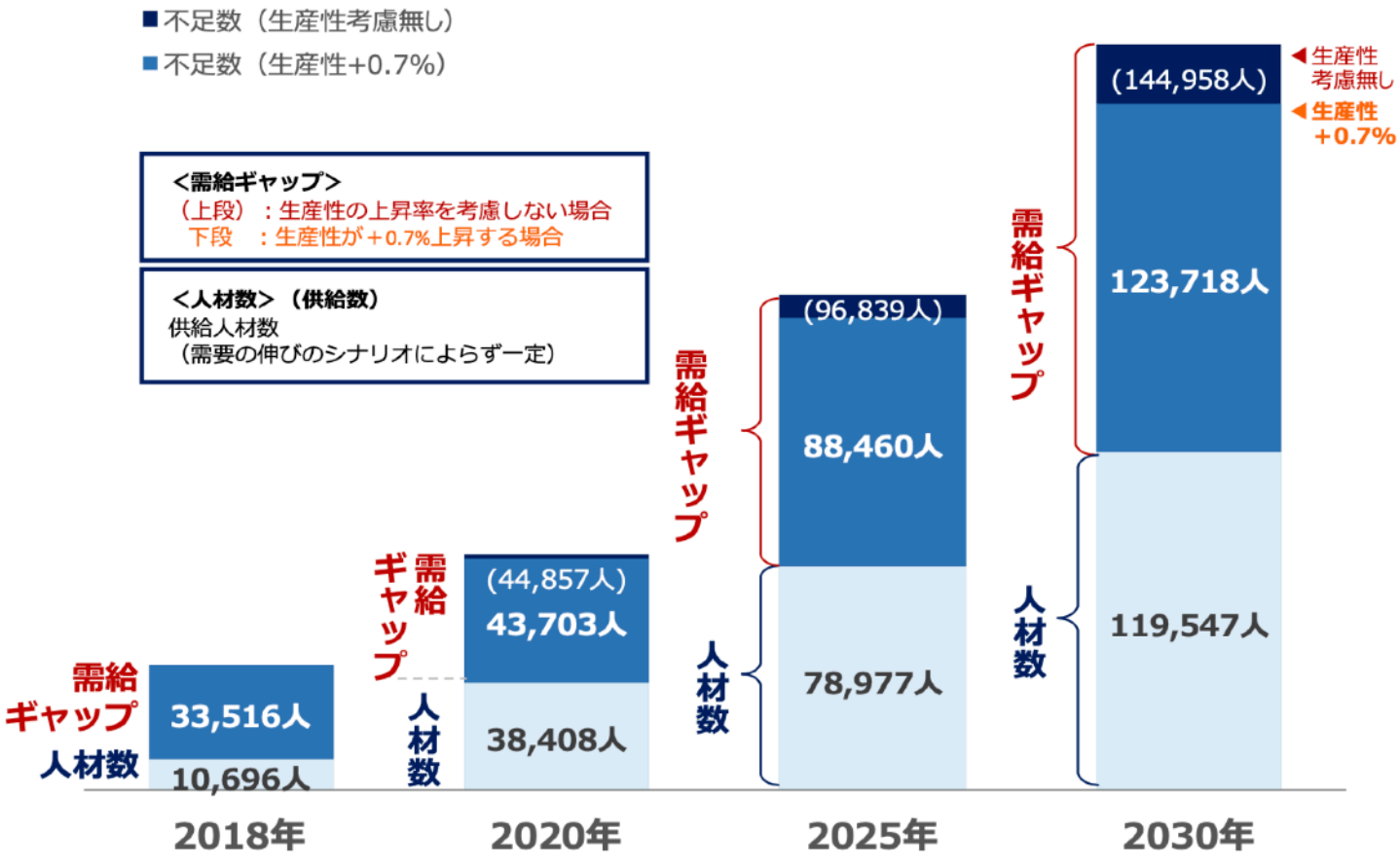


10

下りのエスカレータを登ろうとしていないですか？

逆はしんどい。

さらに、AI（人工知能）の人材は、2030年には、12万人以上不足するという予測データとなっています。
つまり、就職のしやすさや仕事の量では、上りのエスカレーターではないでしょうか。



11

仕事する場所が問われない

ハワイのワイキキビーチで仕事

全部の案件がそうではないですが、案件によっては、リモートで仕事がOKな案件も存在しています。
以前、一緒に仕事していた仲間が大手の外資系企業へ転職していったのですが、そこは、月に５回リモートでの仕事がOKだと言っていました。現在は、新型コロナウィルスの影響で、ほとんどがリモートでの仕事のようです。

場所が問われないという事は、例えば、ハワイのワイキキビーチでココナッツを飲みながら、リモートで仕事をするという事も可能です。

女性の方なら、出産後、小さなお子様がおられると、職場で出向いての仕事より、自宅から仕事できれば、安心して仕事ができるのではないでしょうか？

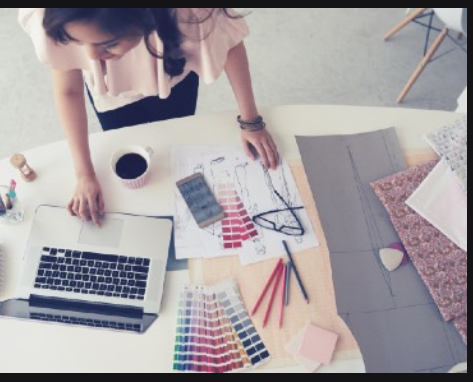
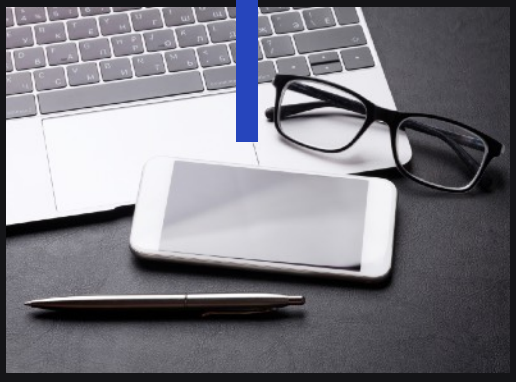
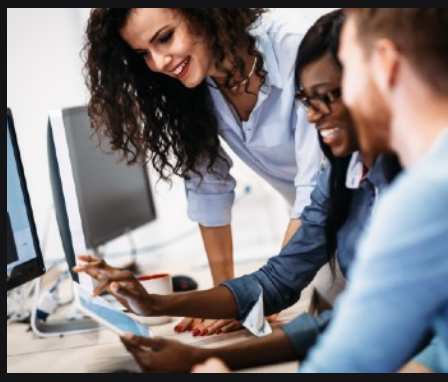
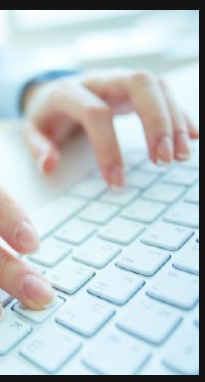
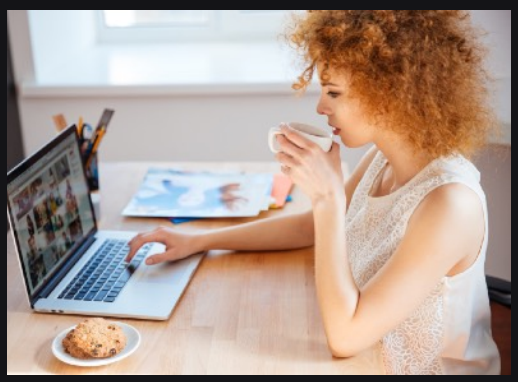
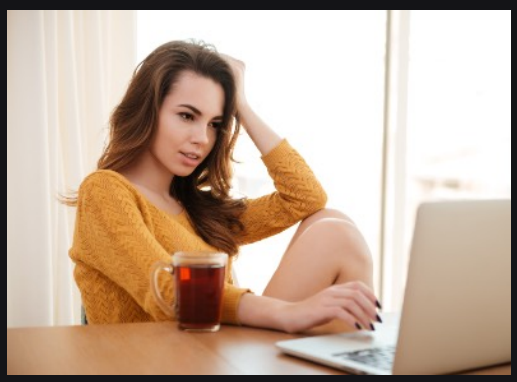
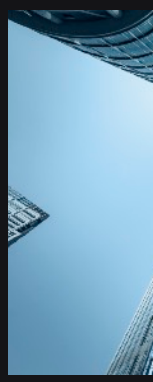
SMART BEAST
Software Engineer

12

ソフトウェア開発者の仕事

ソフトウェア開発者って何をやっているの？

ソフトウェア開発の現場で行われている内容

www.shige-matsuoka.com

13

仕事内容

ソフトウェア開発者は何をするのか？

では、ソフトウェア開発者は、どのような作業を行うのかを説明します。
なお、システムエンジニアは、基本的に設計を担当し、プログラマーは、コーディングより後の作業を行いますが、担当の境界線は曖昧なところが多いので、まとめてソフトウェア開発者と表現しています。

ソフトウェアの１番の目的は、手作業をコンピューターによって自動化する事です。
そして、ソフトウェア開発はプログラミングだけではありません。
まず、プログラミングコードを書く前に、要件定義や設計と呼ばれる工程で、何を解決しようとしているか？を理解する必要があります。

住宅を建てる時を例に例えると、家の間取りの設計図を作成する作業になります。
家の間取りの設計図がなければ、大工さんや内装の人など、それぞれ、勝手な思いで作業をしてしまい、最後にバラバラな家ができてしまいます。
ソフトウェア開発でも、一緒でまず、設計図を作るところから始まります。

14

仕事内容

ソフトウェア開発者は何をするのか？

開発プロジェクトによっては、多少の違いはありますが、基本的なソフトウェア開発の流れは、以下のようになります。

システム要件定義

以下のような、何を解決する事なのか？を明確に定義します。

・ソフトウェアで解決しようとしている問題は何か？

・何を自動化しようとしているのでしょうか？

設計

画面や機能の単位、データベースの構造などを設計します。
繰り返し開発するアジャイル開発を学んだ開発者の中には、
設計をせず、いきなりコーディングを行う人も 多くいますが、
それは間違いです。確かにアジャイル開発は、設計を重要視
しませんが、必要ないということではありません。

コーディング

ソフトウェアの設計で、ある程度のイメージができたら、設計書を元にコーディングします。

デバッグ

コーディングしたプログラムが正しく設計した通りの動きになっ

15

仕事内容

ソフトウェア開発者は何をするのか？

テストとデプロイ

デプロイとは、コーディングしたプログラムをまとめて、
動作する形にする事です。そして、その作成したものを使い、
画面を操作し、正しく動作する事を確認します。

システムテスト

以下のような観点を確認するためのテストを行います。

・処理性能を確認
・修正したところ以外が正常に動作する確認
（リグレッションテスト）
・システム全体を通しての整合性がとれているかの確認

受入テスト

この工程は、ソフトウェア開発者以外の要求元で行う作業です。
要件定義の工程で解決する事としていた事が解決できているか？
について確認します。

運用

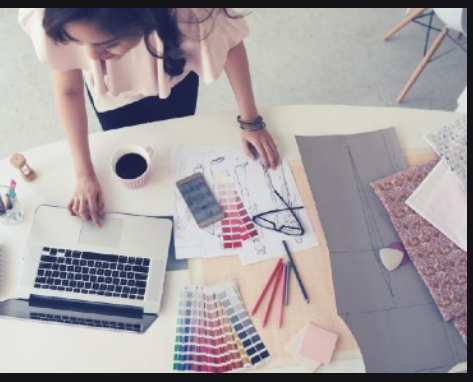
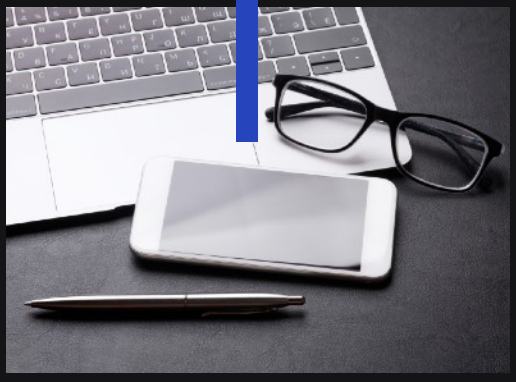
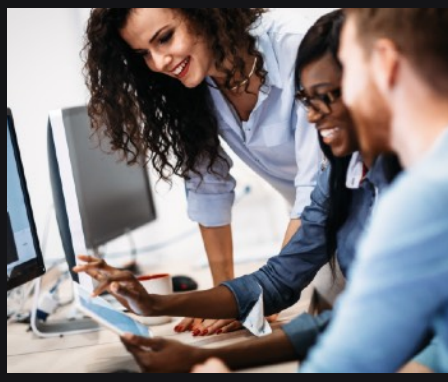
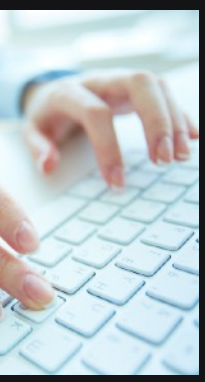
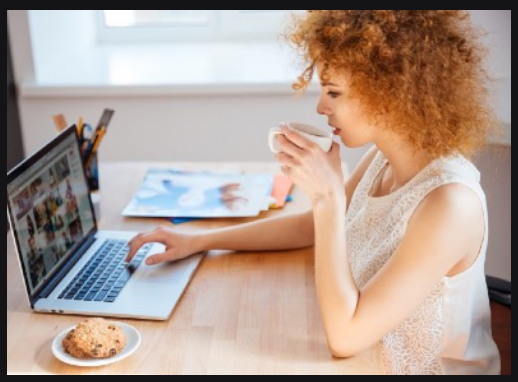
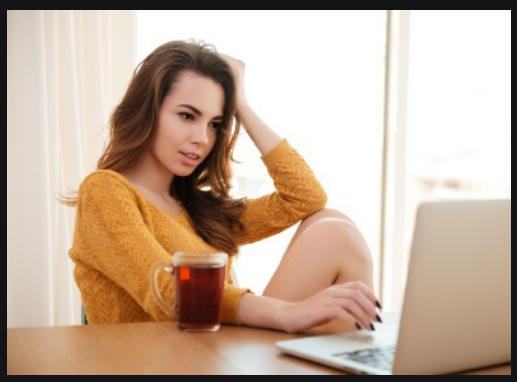
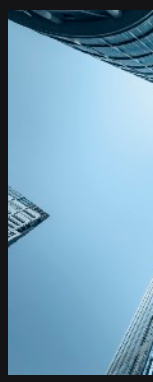
ユーザーがシステム（ソフトウェア）を使用し発生した不具合
や問題について、素早く対応を行います。

SMART BEAST
Software Engineer

16

どのプログラミング言語を学ぶべきか？

たくさんプログラミング言語があって、どれを選べば良いか？わからない！

www.shige-matsuoka.com

17

プログラミング言語の特徴

プログラミング言語の判断基準

プログラミング言語のタイプは、実行する時にコンパイルという作業が必要なもの「コンパイラ言語」と、必要なく、コードを書いた後、すぐ実行できる「インタープリタ言語」という言語の２種類あります。
「コンパイラ言語」は、さらにVM（仮想マシン）と呼ばれる、中間コードを生成し、どのようなOS（windows、mac)でも、実行できるようになっているものが存在します。

言語	タイプ	特徴	難易度
C#	コンパイル（VM)	どのような環境でも対応できる	中
Java	コンパイル（VM)	C#と似ている。案件多い	中
Python	インタープリタ	AIのライブラリが豊富	中
Ruby	インタープリタ	日本製、オブジェクト思考	中
Java Script	インタープリタ	フロントエンドで主流	中
PHP	インタープリタ	WordPressで使われている	中
C++	コンパイル(ネイティブ)	メモリ管理は手動	高

18 どのプログラミング言語？

プログラミング言語の判断基準

どの言語を選ぶかは大きな問題ではありません。それよりもライブラリーやフレームワークを活用する事のほうが重要です。
就職しやすさと将来性で決めれば良いと思います。
以下の言語は代表的な言語なので、将来性もあり就職もし易いです。
各プログラミング言語の案件の人気度や単価は、フリーランス登録サイトに登録を行えば具体的な単価を確認できます。

- C#
- JAVA
- Python
- Ruby
- JavaScript
- PHP
- C++

なお、C++は、メモリの管理をプログラマーが行う必要があり難易度は、他のプログラミング言語に比べて高くなります。
僕の個人的な、オススメとしては、AIのライブラリの豊富なPythonとC#です。

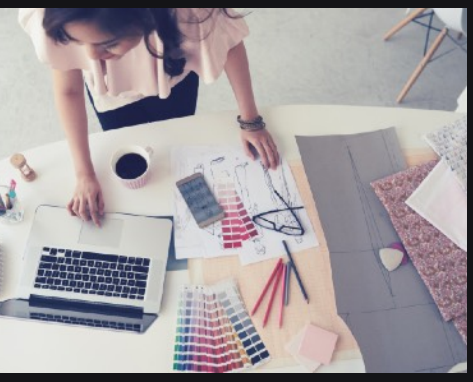
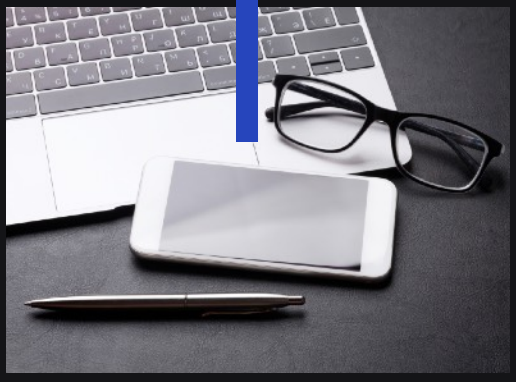
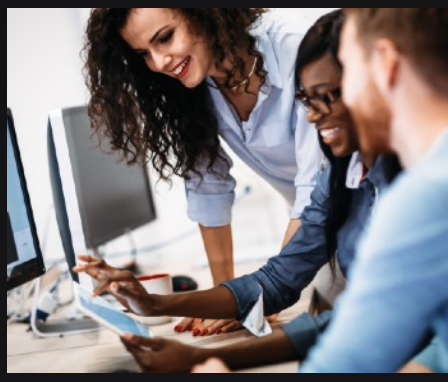
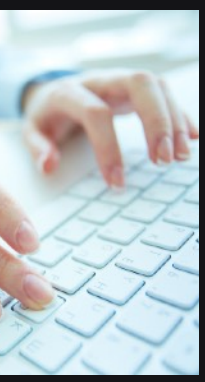
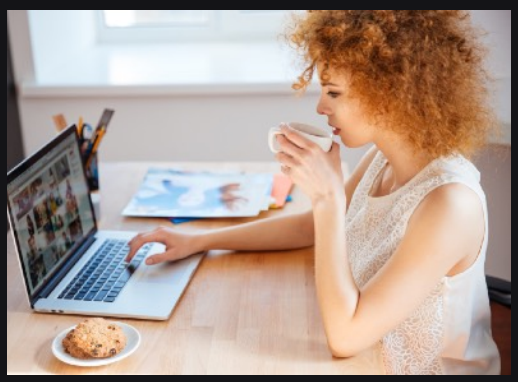
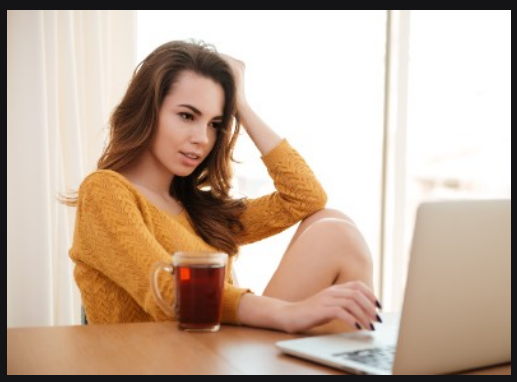
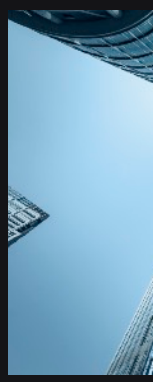
SMART BEAST
Software Engineer

ソフトウェア開発者の重要な技術スキル

19

数あるスキルの中から厳選した重要な技術スキル！

コーディング以外の技術スキルは、ソフトウェア開発現場で必ず必要になってきます。

20

重要スキル１

必ず身につけよう！

仕事をする上で学ぶことは多くあり、15分で読めるようにしているこのガイドでは説明しきれないので、全体像とWebで検索できるキーワードを記載しています。

プログラミングコードの書き方

次の原則を守りコードを書くようにします。

- プログラミングコードは、シンプルにする
- プログラミングコードのコピー＆ペーストは使用しない
- プログラミングコードは読みやすく
- 名前は、ぱっと見わかる名前にする
- 大きくなりすぎた場合、切り離し分離し小さくする
- 資源は限られていることを意識してコーディングする

具体的にどのようなコードが良いか？は、オープンソースのapache、linuxなどのコードを参考にするのがオススメです。

アルゴリズムとデータ構造

- 配列、ベクトル
- リスト／ディクショナリー／ハッシュ
- スタック／キュー
- ツリー
- 再帰

21

重要スキル２

必ず身につけよう！

コーディング基礎

最低限必要なレベルです。まずは、以下を理解し使えるようにします。

- 画面への出力
- 数学計算
- 変数への情報の格納
- 関数、メソッド、モジュールなどの作成
- 関数やメソッドの呼び出し
- 条件分岐
- ループ

データベースの基本的な知識

これも最低限必要なレベルです。他にも場合によって必要となってくる技術は存在しますが、一度に多くを詰め込みすぎると思考の混乱が大きいので、今は、基礎となるものを理解します。

- データベースの仕組み
- データを取得するための基本的なクエリーの実行
- データの挿入、更新、削除
- データの結合方法

22

重要スキル３

必ず身につけよう！

次にあげている項目は、ソフトウェア開発言語によって、どれが使用されているかが違ってきます。
全部覚えれば、ベストですが、最初は、どれか１つ覚えれば良いと思います。

OS

- windows
- Mac
- Linux

フレームワーク

- .netFramework(.net)
- TensorFlow
- Java Platform Standard Edition（Java SE)

ソース管理

- GitHub
- Team Foundation Server
- Subversion

23

重要スキル４

必ず身につけよう！

オブジェクト指向

目に見えない抽象的な概念なので、初めは、よくわからないかもしれません。この概念を理解し使いこなせるようになると、もう初心者ではなくなります。

- オブジェクト
- カプセル化
- 継承
- ポリモーフィズム（多様性）
- デザインパターン
- UML

デバッグ

デバッグは、ものすごく需要です。デバッガーを使いこなせるようにしましょう！

- IDEでのデバッガーの使い方
- ブレークポイントの設定
- 変数のウォッチ

24

重要スキル５

必ず身につけよう！

テスト

代表的なテストは以下のものです。概念だけは覚えておかないと、チームでの開発の時にコミュニケーションミスしやすくなります。

- ホワイトボックステスト
- ブラックボックステスト
- ユニットテスト
- 境界値条件テスト
- 受入テスト
- 運用テスト
- 結合テスト
- 性能テスト
- リグレッションテスト
- システムテスト

開発手法

開発方法は、昔ながらのウォーターフォール開発とアジャイル開発／スクラム開発では、作業工程のサイクルが違います。

- ウォーターフォール開発
- アジャイル開発
- スクラム開発

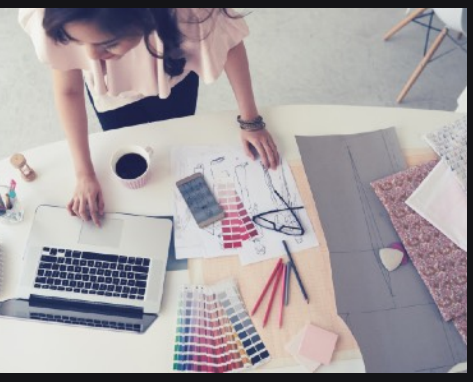
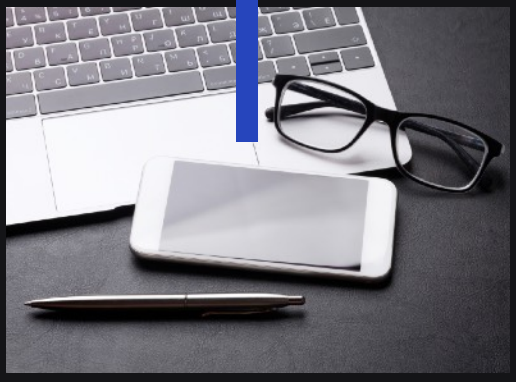
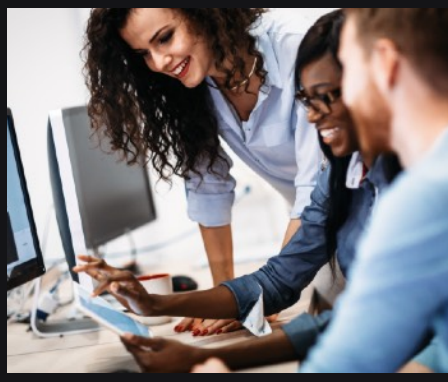
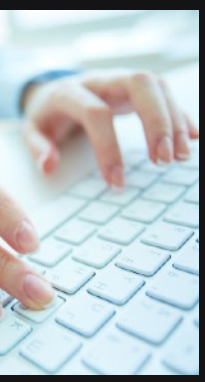
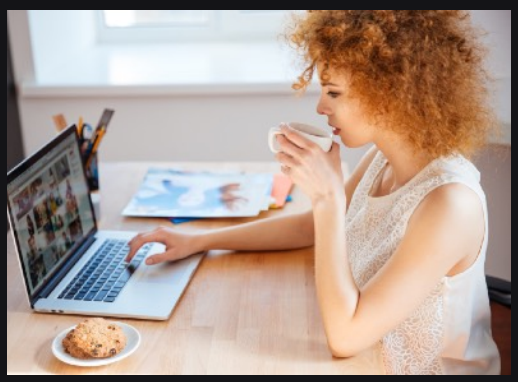
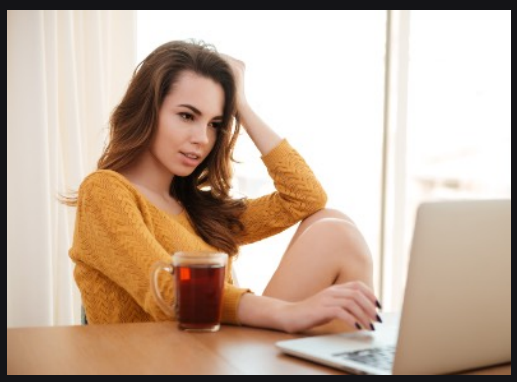
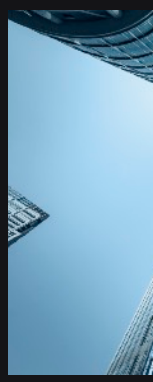
SMART BEAST
Software Engineer

25

どのように学べば良いか？

独学？プログラミングスクール？

業界事情、メリット、デメリットをお話します。

www.shige-matsuoka.com

26

独学で学ぶ場合

モチベーション維持や確認が鍵！

独学でプログラミングを学ぶ場合、以下のようなメリット・デメリットがあります。

メリット

⇧比較的、お金がかからない

⇧自分のペースで学べる

⇧本当の実力が身に付く

デメリット

⇩モチベーションの維持が難しい

⇩正しく理解できているか？確認する方法が難しい。

27

プログラミングスクールで学ぶ場合

高額なところもある！

プログラミングスクールでプログラミングを学ぶ場合、以下のようなメリット・デメリットがあります。

メリット

- モチベーション維持しやすい
- 正しく理解できているか？確認できる。
- 就職保証があるスクールもある

デメリット

- 自分のペースで学べない
- オンラインだと低額なものがありますが、オフラインは高額

28

プログラミングスクールの注意点

これだけは知っといて！

プログラミングスクールの注意点です。

途中解約などした場合の規約を言葉の説明を信じるのではなく、ちゃんと読んで確認しましょう。

無料で就職保証がついているスクールがありますが、ほとんど、就職先の企業や人材会社からキャッシュバックが入る仕組みだと思います。就職先も決まっていたりします。また、年齢制限もあり、２０代までしか受講できないところが多いです。

今、プログラミングスクールは数多く存在し、中にはアフェリエイトを導入している所もあり、スクールのアフェリエイトの記事がかなり多いです。
ちゃんと自分の目で確かめましょう！

29

おすすめプラン

最初から無理しない！

有料プログラミングスクールは、投資効果が良いとは、あまり思えません。金額分の価値があるかどうかは、疑問です。

おすすめは、オンラインを活用する方法です。
マザーテレサは、習慣は性格になり性格は運命になると言ってます。
いかに習慣化する事が重要で、隙間の時間を活用できるオンラインで学習する事がおすすめです。
進研ゼミのベネッセが日本代理店をやっているUdemyなどオンラインスクールで、まずは、少しずつ学んでゆくのが、本当の実力も付き、オススメです。
金額も数千円から可能なコースがあります。

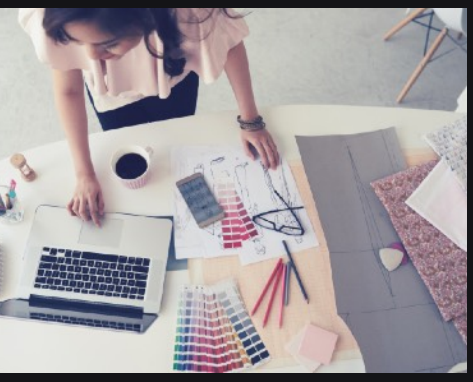
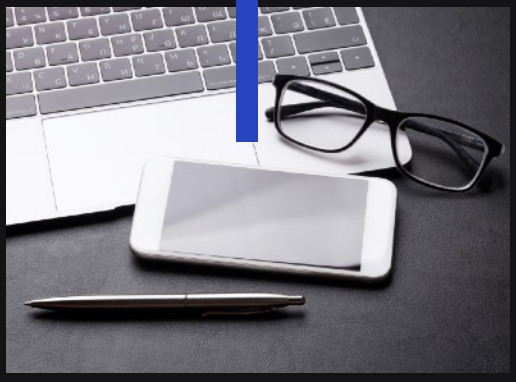
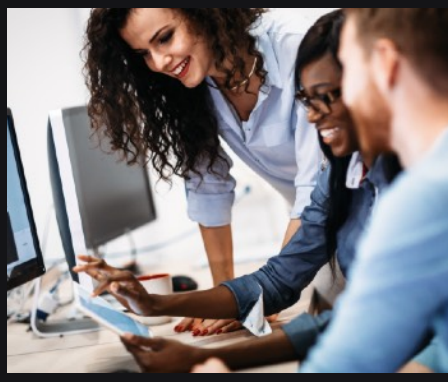
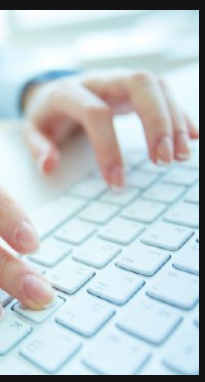
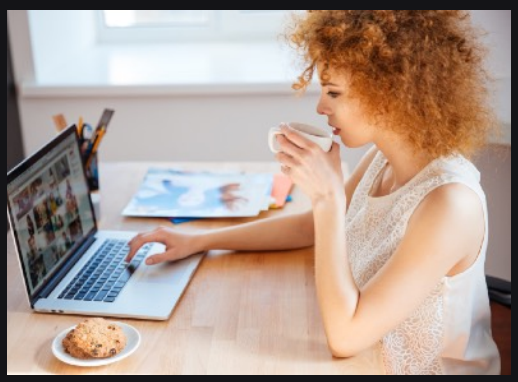
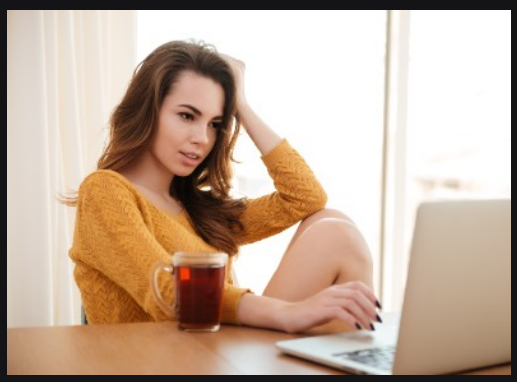
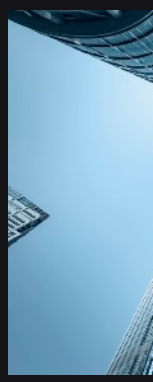
わからない所は、知り合いのシステムエンジニアやプログラマーに聞くのがお金もかからず、学べます。
まずは、大きな資金を投入するよりも、少しずつ、少しずつやってゆく方が長続きします。

SMART BEAST
Software Engineer

ソフトウェア開発者の３つの働き方パターン

会社員、フリーランス、起業？

会社員、フリーランス、起業の特徴とメリット・デメリット

www.shige-matsuoka.com

31

働き方パターン

それぞれの特徴

ソフトウェア開発者の働き方は、大きく３パターンあります。

✓**会社員の場合**

毎月、安定したお金が貰えるのが大きなメリットです。

年収は、400～700万円が平均で、それ以上は一般的でないスキルが必要になってきます。

デメリットは、仕事は基本的に選べなく会社に従う形です。

✓**フリーランス（個人事業主）の場合**

サラリーマンでは、難しかった年収1000万円以上も、比較的、簡単に可能です。

仕事は、自分で選べる分、スキルレベルが低い場合、仕事が取りにくい場合もあります。

✓**起業する（ビジネスオーナー）場合**

自分でWebサービスを構築したり、アプリを開発したり、リスクも大きいですが、リターンも大きいです。

どれも一長一短あります。今の時代、それぞれの境界線が曖昧になってきています。（サラリーマンをしながら起業できるなど）
どっちかと決めるのではなく、両方やりつつ、徐々にシフトしていく事をオススメします。

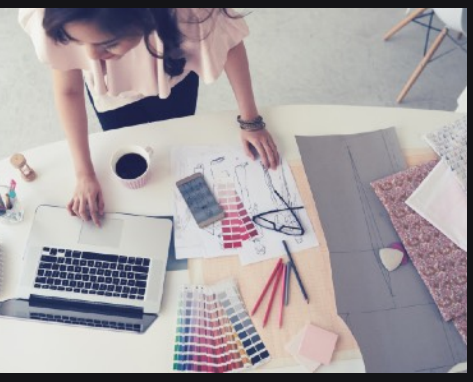
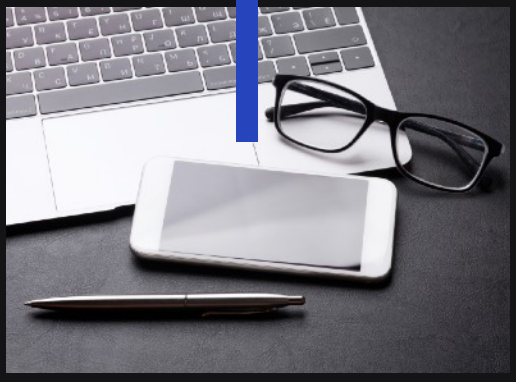
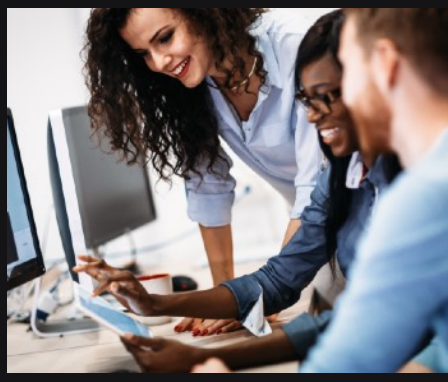
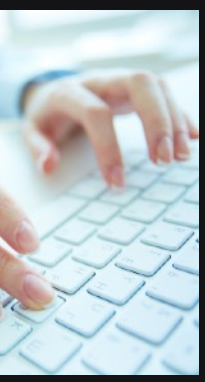
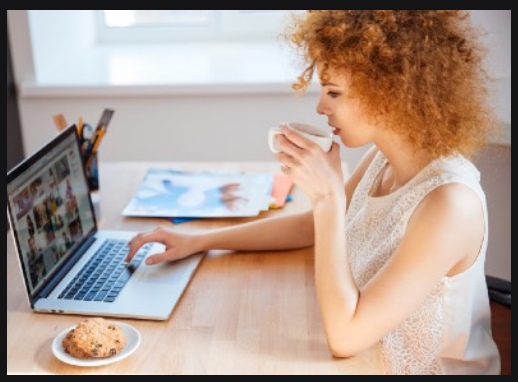
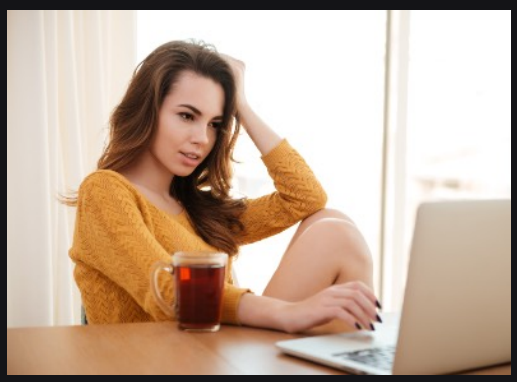
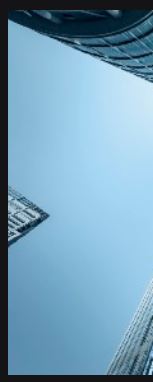
SMART BEAST
Software Engineer

32

自由な時間を手に入れる為に必要なこと

ずっとシステムを開発して、ようやくわかった、共通する１つの真実

これを理解しているかで、結果は大きく違ってくる。正直に言うとあまり知られたくない１つの真実

www.shige-matsuoka.com

33 個人事業主とビジネスオーナーの違い

このポイントは抑えておくべき！

ビジネスオーナーと個人事業主の違いってなんでしょうか？

やっている内容は同じような内容なのに１つ大きな違いがあります。

それは、、、、

ビジネスオーナーは、オーナー自身がいなくても仕事は成り立ちますが、個人事業主は、事業主自身がいないと、仕事が回らなくなるという事です。

ビジネスオーナーは、ビジネスの仕組み（システム）を作り、部分的に自分以外の他人に任せているので、オーナー自身いなくても、仕事は回ります。

つまり、自由な時間を手に入れる為には、ビジネスの仕組みを作り、部分的に仕事を他人に任せる必要があるという事です。

34 ビジネスの仕組みってどうやって作るの？

やろうとしている事は同じです。

ビジネスの仕組み（システム）を作るのって、やった事ないし、よくわからないと、あなたは思うかもしれません。

でも、あなたが今、学ぼうとしているプログラミングは、コンピューターシステムを作ることを目的としています。

つまり、コンピューターシステムの全体像が把握でき、作る事ができれば、ビジネスの仕組み（システム）を作るのと同じです。

大きく分けて２つのパターンがあります。
１つはコンピューターに任せる方法、もう１つは人間に任せる方法です。人間に任せる方法は、従業員として雇ったり、アウトソーシングで外注に仕事を任せたりする事です。

そうなると、お金が必要になってきます。
では、次にどうやって、お金を手に入れるのか？そのポイントをお話します。

豊かなお金を持っている人に共通する１つの真実って？

このポイントは超重要！

システム開発は、お客さまのかなりアバウトな要望を形にしていく事です。
よく言われるのは、作業を自動化して、かかっているコストをレポートで表示できるようにしたい。とか、ザックリとした要望です。
そこで、お客さまにヒアリングを行い、本当に望んでいる事や、具体的な作業は何をやっているか？などを仮説を立てながら検証し、システムの機能に落とし込んでいきます。

完成したシステムが、有益な素晴らしいものか、そうでないかは、１つのポイントが大きく関係します。

そのポイントが豊かなお金を持っている人と共通する事だと、私は最近、気がつきました。

そのポイントとは、、、、、、

それは、レバレッジです。

36

レバレッジとは？

てこの原理。てこを使えば小さな力でも大きなものを動かせます。

一般的にレバレッジは、少ない資金で大きな資金の取引をする事で投資での意味がありますが、ここでのレバレッジは、投資のみならず、「自分以外の何か」の力を得て、自分一人以上の結果を得る事をいいます。

例えば、１００人の人に連絡をしたいとします。
レバレッジ無しで、電話やコンピューターの力も借りずに自分１人だけで行う場合、１００人分の人に会いに行く必要があります。
とんでもない労力がいりますね。

では、レバレッジとして、コンピューターの力を借りた場合、LINEで１００人分のグループLINEにコメントすれば、一瞬で終わります。

レバレッジ無しだと１馬力、レバレッジありだと１００馬力、かかる労力は桁違いです。

豊かなお金を持っている人は、共通してレバレッジを利かせています。

37 レバレッジについて

てこの原理。てこを使えば小さな力でも大きなものを動かせます。

ビジネスオーナーは、従業員を雇い、レバレッジを利かせています。
投資家は、お金を投資し、投資先の人達が、配当の為に働きレバレッジを利かせています。

そして、レバレッジには、リスクが大きいものと小さいものがあります。
リスクが大きなものは、投資のレバレッジです。
自己資金の何倍も取引できる投資は多くありますが、その分自己資金が無くなるもの一瞬です。

リスクが小さいものは、コンピューターによるレバレッジです。
コンピューターは１度作ったものをコピーするのは、一瞬です。
これがコンピューターの大きな強みで、このレバレッジを有効活用するのが、オススメです。

コンピュータのリスクは、コピーしやすいので、真似されたり、コピーされるという事がリスクですが、たかが知れています。

38 コンピューターによるレバレッジを使おう！

１馬力ではなく、何千万馬力となるような作業を見つけて、コンピューターにやらせよう！

まとめると、豊かなお金を得る為には、コンピューターによるレバレッジを利かせる事がリスクも小さく、オススメです。

たとえば、あなたは、プログラミングを学ぶことにより、価値のあるアプリやWebサービスを作成できるようになりました。でも、それだけでは、自由な時間や豊かなお金を手に入れることは、難しいです。

ビジネスでは、集客が重要ですが、集客部分をマーケティングオートメーションによって、コンピューターに肩代わりさせレバレッジを効かせることにより、自由な時間や豊かなお金を手に入れることが可能となります。

日頃から、「この作業をコンピューターに肩代わりできないか？」と常に考えて生活してみてください。

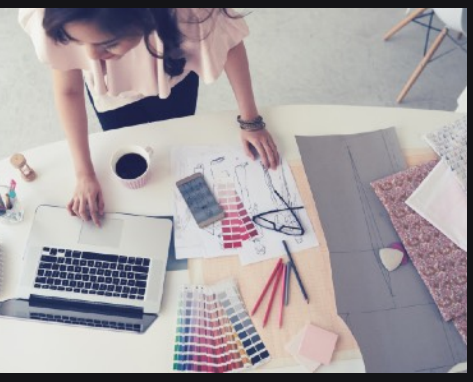
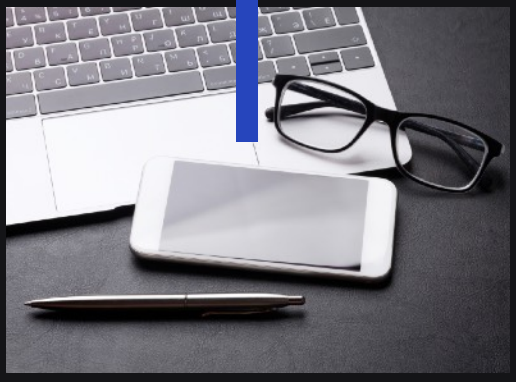
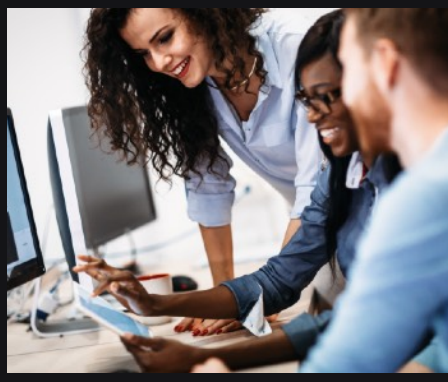
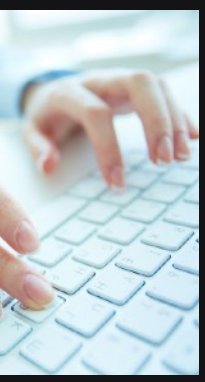
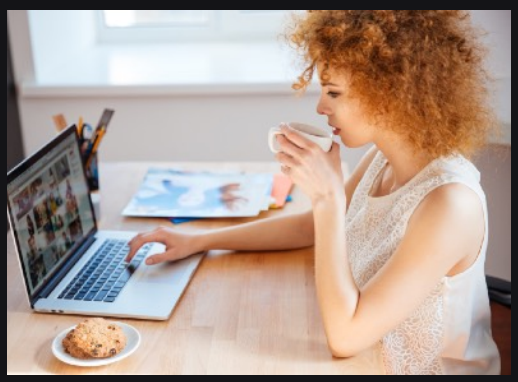
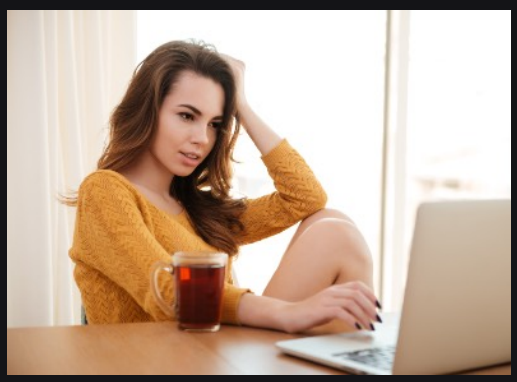
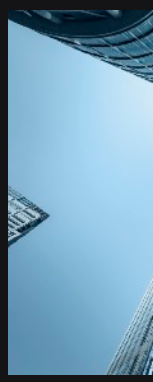
SMART BEAST
Software Engineer

39

最後に

情報を入れるだけでは、現実世界は変わらない！

重要なのは、情報を入れるインプットよりもアプトプット！

40

最後に

あなたならできる！

学習は、情報を入手するインプットでは、脳内のみ変わります。
入手した情報をアウトプット（行動）することによって、はじめて現実世界が変わります。

なので、行動に移さなければ、あまり役に立ちません。
行動を変え教わった原則やベストプラクティスをあなたの人生に取り入れたいと思うのであれば、繰り返し実践（アウトプット）してください。
繰り返し週間化することでストレスをかけずに覚えたことを消化できるようになります。

初めの第一歩は、誰でも怖かったり、面倒だったりするものです。でも、後になって、「あの時、やっておけば良かった」と後悔するより、まずは、第一歩を踏み出してみてください。

あなたがプログラミングのスキルを身に付け、自由な時間や経済的にも豊かになり、充実したライフスタイルができるようになることを楽しみにしています。

SMART BEAST
Software Engineer

41

自己紹介

富山県出身。マーケティングと心理を融合した総合ソフトウェア開発について、経験の無い未経験者でもわかるようにスキル・ノウハウを提供。未経験者からフリーランスのシステムエンジニアへの転向成功をサポートした実績は多数存在。ソフトウェア開発およびマーケティングに関する専門知識を持つフルスタックエンジニア。小学校4年からプログラミングをやり始め、社会人となった1998年から、主に、東証一部上場の大企業へのシステム開発を多数経験。心理についても学んでおり、２０代の頃には、人間の深層心理について、雑誌プレジデントにも掲載。

松岡　成佳
（まつおか　しげよし）

サイト

HTTPS://SHIGE-MATSUOKA.COM