

```
print(
```

プログラミング * 人脈

```
)
```

```
$ python techgym.py
```



techgym

**テックジム入校案内
高速学習のポイント**



AIに興味があり、
来自分サービスを作りたい



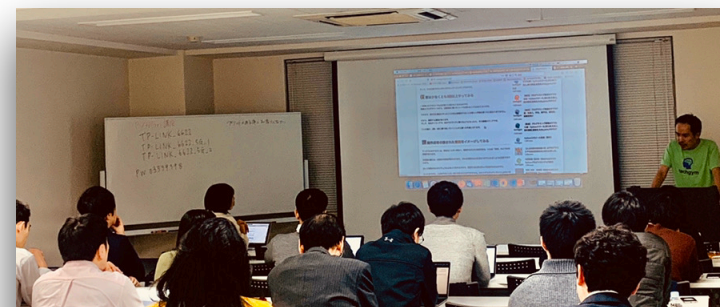
今の仕事も続け
つ、ステップア
ップとして



社内のエンジニア
と円滑に仕事をす
めるため



子供に教えるため
に、まずは自分も
習得したい！



テックジムなら 最短200時間でスキルが身につく

テックジム 方式とは？

最速でマスターできる

「**テックジム方式**」とは、基礎知識なしでも、座学なしでプログラミングに専念できるように設計されたプログラミングのカリキュラムメソッドです。優れたエンジニアの多くは、職業訓練学校や研修スクールで学ぶような学び方をしません。作りたいものを作っていたら、自然に習得できているのです。そして、優れた指導者（メンターや師匠）との出会いがエンジニアとしてのプロフェッショナル性を高めます。

この自発性とプロフェッショナル性を兼ね備えたのが「テックジム方式」です。**授業を聞いたりテキストを読むことで失う時間を純粋にプログラミング時間に向ける**ことで、習得効率は飛躍的に向上し、モチベーションも維持できます。その時点で知らなくていい知識や概念を可能な限り削ぎ落とし、知るべきタイミングが訪れた時にはすでに課題を通じて覚えているような**スキル習得の自動化**を図ります。

講座で用いるサンプルソースや課題は、最も効率よく学べるように細部に渡って設計されています。**まるで魔法にかかったように**プログラミングスキルが習得できるのが「テックジム方式」の醍醐味です。

現在、初学者の方へはpythonコース、現役エンジニア向けにAIコースそして、副業をテーマとしたRuby on Railsコースをご用意しております。

Python 一択の理由

AIの近道

「少ないコード量で簡単にプログラムがかける」「コードが読みやすい」といったメリットのある言語で、広い分野で使われています。なかでもPythonは、ディープラーニングのライブラリーが圧倒的に豊富で、「**人工知能（AI）をやるならPython**」と言われるほどです。

参考）Pythonのライブラリー事例

- ・ tensorflow：ディープラーニングのフレームワーク
- ・ keras：ディープラーニングのフレームワーク
- ・ jupyter：Webブラウザ上で操作できるPythonカーネル
- ・ h5py：h5形式で保存
- ・ Flask：Webアプリのフレームワーク
- ・ scikit-learn 機械学習ライブラリ
- ・ matplotlib：グラフィックライブラリー
- ・ OpenCV-python：画像処理ライブラリー

世界的に見ても、**PythonはJavaやPHPを抜き、1位の人気**を博しているプログラミング言語なのです。また、Pythonは教育用のプログラミング言語としても有能で、最初に学ぶ言語としては最適との評価も高いプログラミング言語です。というのも、誰が書いても同じようなコードになるため、**他の人が書いたコードを元に勉強するのが簡単**だというわけです。

PythonはGoogleやNASAの内部でも利用されており、現在でもGoogleの求人情報にはPythonエンジニアの求人が存在します。これからプログラミングを勉強するなら、Pythonしかありえないといっても過言ではありません。

月額2万の プログラミング ライフ

ビジネス人脈や仲間も

「テックジム（TechGYM）」は、どんなテクノロジーにも即座に対応できる「自走できるエンジニア」をゴールにした**自習形式のプログラマー養成塾**です。

どんなレベルからでも始められますので、全くのプログラミング初心者からベテランエンジニアまで、各個人のスキルレベルにあわせて学ぶことができます。

最速でプログラミングスキルの獲得をしたい方にはテックジムのカリキュラムがオススメです。世の中の多くのプログラミングスクールは、初学者のみを対象としておりますが、テックジムはエンジニアが集まる会員コミュニティです。実際に訪れている人は、初学者はもちろん、起業仲間を見つけない人、Pythonを高速で学びたい人、リクルーティングしたい人など、動機は様々です。

自習形式だからこそ、課題解決のために自らが積極的になって動く必要があり、他の会員とコミュニケーションをとりながら進めていくこともあるでしょう。

こうした中で仲間が生まれ、それは同士となり仕事仲間を見つけることができます。会員同士の交流は自由ですので、エンジニア仲間や生涯の仲間を見つける感覚でご参加ください。まさに、**いつでも参加でき、ずっとスキルを磨き続けることのできる「ジム」**の要素が「テックジム（TechGYM）」の大きな特徴です。

ちょこっと 副業したい

Ruby on Rails

Railsコースは、**ずばり副業でウェブアプリの開発のお仕事をとれるレベル**を目指したコースです。ランサーズやクラウドワークスなどで、資料請求フォームやホームページなどのお仕事をして副業にしたいという方にはオススメです。

エンジニア採用の現場をヒアリングしたところ、「スクール出身の方のポートフォリオは、そのスクールが用意したテンプレートを味付けしただけで、実力が感じられない」とのことでした。

テックジムでは、もっと難易度の高く、採用現場の人たちを唸らせるほどのレベルを設定しました。

テックジム4つの特徴

- ・ 授業がない、教科書がない → 受動的な学習法を排除
- ・ 自分のペースでできる → 落ちこぼれることはない
- ・ 自己解決できる能力がつく → 新しいテクノロジーに追いつける
- ・ 仲間ができる → 就転職・案件紹介・起業などのビジネスチャンス

カリキュラムの進め方

教材は課題のテキストとサンプルソースで構成されており、1課題ごとにサンプルソースやヒントが用意され、課題の通りにコーディングしていきます。私たちの教材（プリント配布）はいわゆる短歌の世界で、少ないコードの中にも創意工夫があります。

自己鍛錬に耐える良問集ですので、復習をすることが前提となっています。テックジムでは5回の復習を推奨しておりますが、復習するたびに自分の成長を実感できるはずです。簡単な問題を何回も解いたところで、スキルは上達しません。技術習得の階段を効率よく上がっていただくために、最小限のテキストで様々なプログラミングテクニックを学んでいただきます。

テックジムの教材をたとえるならば、公文式の自習と、Z会の良問です。公文に通うように自分一人で進められて、何度やっても新しい発見があるZ会のような色々なトリックが隠されている良問集です。

【副業でウェブアプリ開発】

Ruby on Railsコース

- 1章「問合せフォームを作ろう」
- 2章「クローラーを作ろう」
- 3章「APIを使って読書管理アプリを作ろう」
- 4章「rails new」ゼロからアプリケーションを作成しよう
- 5章「アプリケーションをHerokuで公開しよう」
- 6章「coffee・sass」アプリケーションにデザインと動作を加えよう
- 7章「様々なGemを使ってみよう」
- 8章「予約サイトを作ろう」
- 9章「カタログサイトを作ろう」
- 10章「卒業課題」設計段階からアプリケーションを作成しよう

【初学者向け】Python基礎コース

プログラミングの基礎を学んでいただきます。就職できるレベルに達するには500時間の履修が必要だと言われておりますが、テックジムでは200時間でそのスキルレベルに達することができます。余った300時間は、AIコースに注ぎ込んでください。

- | | |
|-----|--------------------|
| 第1章 | 「じゃんけんゲームを作る」 |
| 第2章 | 「間違い探しゲームを作る」 |
| 第3章 | 「野球シミュレーションゲームを作る」 |
| 第4章 | 「カジノゲームを作ろう」 |
| 第5章 | 「発展系の書き方をしよう」 |
| 第6章 | 「ブラックジャックゲームを作ろう」 |
| 第7章 | 「卒業課題」 |

【現役エンジニア向け】 AIエンジニア養成コース

数学の素養がなくても取り組め、機械学習やディープラーニングを効率よく習得できるカリキュラムです。現役エンジニアによる「現場に役立つテーマ」を揃えております。

- | | |
|------|---------------------------------|
| 第0章 | 「データ分析入門とデータ可視化。Kaggleにチャレンジ」 |
| 第1章 | 「教師なし学習と自然言語処理入門」 |
| 第2章 | 「特徴量エンジニアリング」 |
| 第3章 | 「回帰問題」 |
| 第4章 | 「分類問題」 |
| 第5章 | 「実際のビジネス現場でデータ分析」 |
| 第6章 | 「分析に必要な統計学の基礎」 |
| 第7章 | 「DeepLearningの基礎理解。画像分類。」 |
| 第8章 | 「DeepLearning応用編」 |
| 第9章 | 「GANによる画像生成とRNN/LSTMによる自然言語処理」 |
| 第10章 | 「衛星データと機械学習」 |
| 第11章 | 「アンサンブル学習と精度向上手法」 |
| 第12章 | 「機械学習でオリジナル画像判定アプリをつくる」 |
| 第13章 | 「AI x IoT（センサーデータ分析と画像分類・物体検出）」 |
| 第14章 | 「効果検証に機械学習を導入しよう」 |
| 第15章 | 「推薦システムを作ろう」 |

プログラミングを効率よく学習する10のポイント



techgym

1.いま覚えなくてもいい事はあえて流す

サンプルコードの全てを概念から覚えようとするとなかなかとっつきづらいものです。教材の裏には作成者の意図があり、ここで学んで欲しいところと、ここでは学ばなくていいところがあります。

あとで学ばばいところに詰まることは、教える側の意図と反することなので、細かいことに気にせず、「こういうものなんだ」ぐらいで流しておきましょう。言ってみれば「スルー力」。脳内メモリーが満タンになってスタックしないように「スルー力」を身につけましょう。

スティーブ・ジョブズは、毎日ジーンズに黒のタートルネックだったのは「余計なことで脳内メモリーを消費したくないからだ」と言ったそうです。

2.とにかく適当でもいいから書いてみる

過去に書いたことがあるコードや、教材のヒントを参考に、とりあえず書いてみるのが重要です。サンプルコードを見る場合は、動いているプログラムと見比べて、この辺りにこの機能（動き）がかかれていると見当をつけて「類推」します。すると、この関数はこういうものなんだろうと、なんとなく認識できます。こんな感じでいいんです。この「類推力」。結構重要です。類推する癖がつくと、センスが磨かれていきます。逆に「まずメソッドを覚えて」というやり方だと、いちいちリファレンスを見ないと前に進めない臆病者になってしまいます。テキスト読解を主にするような勉強法では、わかったつもりでいて、実は書く時に何を書いていいか困ることがあります。この困った時間が実にもったいない。なにはともあれミスしてもいいから書いてみて、ビルドしてみる。エラー文をもとに、また書いてみる。「これではどうだ？」みたいに試行錯誤してみることは、後から血となり肉となります。とりあえずコーディング（プログラミング）するという癖がつけば、初歩的なミスタイピングを防ぐことができます。プログラミング学習では、「習うより慣れろ」という部分が結構あるものです。言ってみれば「アテ感力」。アテ感、行き当たりばったり、上等。「こう書いてみれば行けるんじゃないかな」で、いいんです。

3.わからなかったら、答えを見ればいい

わからなかったら、答えのソースを先に見るというのもアリです。といいますのも、答えのソースを「読むことができる」こともスキル習得の大事な一歩となります。いわば「諦め力」。自分で考えてコーディングする時間をあらかじめ決めて、できなかったら答えをみればいいんです。また、脳内メモリーがパンパンになって手が動かなくなった時は、すぐに「諦め力」を発揮して、写経をはじめればいいんです。最初のうちは答えとの差分をゴッソリコピペ。次に、一行一行コピペ。最後に、ちゃんとタイピングするって感じです。コピペ上等上等。世の中にはコピペエンジニアと揶揄される人たちもいます。でもちゃんと上等な給料をいただいている。初学者ならなおさら、コピペ上等です。ただ、自分でなにがしかコーディング（プログラミング）してから、答えを見ることがポイントです。自分で書いてみることで、この答えを見るときの意識が変わり、答えのソースから新しい発見を見出すことができます。ですので、くれぐれも、最初から答えを見るという誘惑に負けないようにしましょう。

4.暗記をする必要は全くない

受験教育では、暗記を前提とした学習法が蔓延していて、「公式は覚えておかなくてはならない」という気分に陥りがちですが、プログラミングの世界では、わからなかったらググるのが当たり前です。要求仕様に沿ったプログラム開発ができることがエンジニアの目的なのでから、わからなかったら調べればいいのです。そのうちに覚えてしまったら結果オーライですし、そんなに使わないものは忘れてしまったとしても、また調べ直せばいいのですから、忘れて結構です。ただ、調べる時間が短縮されることに越したことはありませんので、どここの部分でこの関数を使ったことがあるというように、一度学んだことは、あとで調べやすくすることが無駄時間の短縮のポイントです。そのために、ここは覚えておくといいなと思ったことを、ノートにとっておきましょう。とはいえ無闇にノートをとりまくらなくてもいいです。何度も間違えたとか忘れるようなことがあればノートにとるような癖をつけましょう。

5.とにかくググる

優れたエンジニアはグーグル検索がお上手です。グーグルはノウハウの宝庫。知りたいことはすべてグーグル先生が教えてくれます。試験会場にいるあなたがいたとします。あなただけはスマホがある。カンニングし放題ですよ。これがスーパーエンジニアなのです。私たちプログラマーは賢いことを証明するために仕事はしていません。仕様を満足させるプロダクトを作るのが我々の仕事です。そして、その仕事めっちゃ早いのがスーパーエンジニアなのです。「ググリ力」は常に鍛えておきましょう。

プログラミングを効率よく学習する10のポイント

6.復習は少なくとも3回以上やってみる

一度解いたテキストでも日が経つと解けなくなるものです。熟練のプログラマーですら、2週間前に書いたコードを忘れることもあるくらいです。ですので、別の日に過去にやったことがある課題をやることは新しい問題を解くのと変わりありません。だから、復習する意味があります。そして、あるタイミングで、自分でスラスラと解けるようになったら、その課題はクリアです。こんな風に、2回、3回と繰り返していくことが上達への早道となります。

7.課題作成者の隠された意図をイメージしてみる

テックジムのテキストは、無駄なことが一切なく、復習するたびに発見がある、いわば「短歌」のような世界観があります。各問題の裏には、出題者の意図が隠されており、それを見破れる力を身につけていただくように設計されています。決して無駄のないカリキュラムとなっておりますので、復習するたびに新しい発見があるものです。といいますのも、人は全てのことを一瞬に覚えることができません。ひとつずつクリアしていくしかないので。一方で、教材作成側は色々な仕掛けをこしらえており、ひとつの課題で、複数の覚えることを仕組んでいます。課題ごとにも段階を踏んでおり、作成者の意図をイメージしつつ、「ひっかけがどこにあるのか？」という視点も持ち合わせて、作成者との駆け引きを楽しんでみてはいかがでしょうか？

8.とにかくアウトプット（コーディング）が大事

プログラム説明の文章を読む時間や、講義を聞く時間は無駄な時間になりがちです。その中には無為に悩む時間があったり、わからなくて思考が止まったり、「読んでいるから大丈夫」とか「聴いているから大丈夫」という自己満足に終わる時間が大部分を占めたりします。そもそも、受動的に摂取する情報は定着しづらいですし、自分の理解スピードにぴったりあうことも稀なのです。あえて断言しますが、プログラミング学習はソースコードを書く時間に比例して上達します。また、自分でソースコードを書いた上で、解答のソースコードを読むことも理解を促進します。逆に、ただ与えられた情報を摂取するだけではプログラム上達になりません。理解したつもりでも「いざアウトプットしようとしても全く歯に立たない」ということがよくあるのです。ここが、プログラミング学習での落とし穴であることを肝に銘じましょう。テックジムでは基本的には自習スタイル。わからなかったらトレーナーに聞けばいい。このように必要最低限の情報を能動的に摂取するというスタイルが最も効率の良い学習法なのです。

9.ひとつ上達したら、その上達部分を人に教えてみる

まず、人に教えることで、自分の理解度を試すことができます。もし、うまく教えることができなかったら、まだあんまり理解していないことを自覚しますし、教えてみることで改めて気づくことがあり、理解が進むものです。実際、エンジニアは仲間を集めて勉強会を開催するのが好きです。これは、わからないことを教えあうことで、お互いの弱点を補完しあういい機会になります。一緒にいるだけで頑張ろうという気持ちにもなります。自分より上達した人と一緒にやることも、自分より上達していない人と一緒にやることも、両方とも有益なことを彼らは知っています。何より、教えることで教えた相手に喜ばれることがモチベーションにつながり、上達のステップを上げることができます。世間にお役に立てたということが、自己肯定感を育くむのです。

10.他人と比べない。ライバルは未来の自分と過去の自分。

「授業」のデメリットは、一度落ちこぼれると復活しづらいということです。授業が終わるまで苦痛で、時間の無駄がでますし、別の時間で復習を要します。一方で、できる人にとっては、退屈だったり、落ちこぼれをマウンティングしだす始末。こんなに社会悪なことはありません。明治以降に欧米列強のやり方を取り入れた教育はもう限界にきています。このやり方は、植民地にて原住民にスペイン語や英語を教えるために、効率よくかき集めただけなのです。その点、江戸時代の寺子屋形式は理想的です。学習が進んでいる人は、わからない人に教えてあげます。先述の通り、教える方にもメリットがあり、教わる方も感謝します。このように「教育」は「お互い支え合うこと」がベースであるべきです。実際、エンジニアの現場は、いろいろな人種、スキルセットのバラバラな人たちがあつまって（あるいはリモートのときもある）、ひとつのプロダクトを作ります。スキルセットに応じて色々な作業がありますし、「ギャラ」と「納期」と「要求しているクオリティ」がマッチすれば次の仕事もよばれます。「俺はあいつよりも優れたエンジニアだ」なんてことを言い合う暇はありません。すべては役割分担です。「できないエンジニアが使えない」というわけではありません。エンジニア以外の仕事だって、こうあるべきです。色々な人は色々なスキル（低スキルも歓迎）と時間を持ち合って、お互い尊重しあえばいいのです。均一な人材を集めがちな企業はダサすぎます。コミュニケーションスキルを気にしすぎなのもJAPANならではの残念ピーポーマックスですね。流暢な日本語をしゃべる外国人なんていませんからね。日本人同士の中でも「コミュニケーション能力」で人材の振り分けをするということは、「外国人と仕事をしないぜ」って言っているようなもんです。

【授業の流れ】

テックジムは自習形式ですので、
開校期間であればいつでも入室・退室できます。

- ・最初に第1章の課題プリントを配布します。
- ・5回復習したら次の章のプリントを配布します。

わからないことがあればトレーナーに質問してください。
また会員SLACKでは会員同士で交流することができます。
zoomでのオンライン対応も行っています（※校舎により）

【全国展開】

2020年7月現在、全国30校で運営しております。

テックジムが存在していない府県に在住の方向けには通信販売を用意しております。詳細は公式サイトをご覧ください。

SlackとZoomでのオンライン対応有り

【申し込み方法】

テックジムの各校舎のページからオンラインでお申し込み
ください。カード決済となります。

【料金について】

【入会金】 33,000円

【料金】 通常会員：月額22000円

【割引】 学生・シニア（60歳以上）：月額11000円

※校舎によって違う場合があります。

【留意事項・お支払いについて】

- ・参加時にノートPCをご持参ください。
- ・Gメールアドレスを利用します。
- ・登録時に「入会金＋当月日割り計算分＋来月分会費」をお支払い頂きます。
- ・次月分の会員費は毎月26日に自動引き落とし（カード払い）
- ・途中解約や休会は前月10日迄にご連絡ください。（月末休会は当月10日まで）
- ・累計3ヶ月以上在籍された元会員の場合、再入会の入会金は無料です。

【トレーナーからの学習アドバイス】

- ・プログラムの質問ができる人を探す。
- ・できない部分をそのまま聞くのではなく、原因調査方法・検索方法など解決に至るまでの考え方についてを質問する。
- ・プログラムに書いてあるひとつひとつ”なぜ”そう書いているのか突きつめる。
- ・模倣と量をこなす。
- ・様々な人の様々なコードを真似してると違いが見えてくる。その違いの原因やベターなやり方を調べる。
- ・自分でこれだと思ったプログラミング方法を選択して継続する。
- ・基礎理論的な部分は「勉強する必要性」を感じてから勉強する。（量が膨大すぎて、モチベーションが下がることを避けるため）
- ・チュートリアルやサンプルコードを見つけて、正常に動くものをローカル環境に構築してから改造したり、調査したりする。
- ・新しいフレームワークとか言語は、一回失敗して全部作り直すの覚悟でどんどん手を動かす。

受講者の声



Mくん
中学校3年生

僕はこの塾をネットで見つけ入門講座で体験をしてから入りました。

どの塾も月に数十万するものばかりで、学生が行けるような場所ではないと思っていましたが「TechGym」の月謝を見たとき、ぜひ一度見学しに行こうと思いました。

大人ばかりの場所に行くことが少ないので不安でしたが、TechGymの方がわからないところを優しく教えてくださったり主婦の方がいたりして、とても親しみやすい場所だなと思いました。

Python学習を終えて教材以外のプログラミングをやっていると自然とコードが打てて、重要な学習事項が網羅されている教材だったのだと実感しました。

テクノロジー関係の話に興味がある僕は、学習だけでなくコミュニティの場所としても楽しい場所だと感じています。



Mさん
20代後半・元銀行勤務

いわゆる普通のプログラミングスクールのような講義等はなく、与えられた課題を作り、作り終わると自然とそれに必要な知識が付いていて、「経験から学ぶ」ということを感じました。

特にテックジムをお勧めしたいのは私のように未経験の方です。

非効率な学習をしようがないカリキュラムが用意されているので、最短でスキルを身につけることができるからです。

確かに最初はハードに感じることもありますが、それが良い具合に筋肉痛を与え、楽しさに変わっていきます。講師の方々や他の受講生との交流の機会もあるので、途中の挫折も少ないかと思います。

おまけに料金も他のプログラミングスクールと比較してもかなり安く、コスパ最高です！

テックジム代表からのメッセージ



テックジム株式会社
代表取締役 中村安幸

■最適な自習形式

プログラミングの学習は自習形式が最も適した方法です。

通常プログラミングの学習は個人ごとに習得スピードが違ってきます。

それは当然で学校での勉強とは異なりセンスや意欲が大きく関わってくるためです。

習得スピードが早い方と遅い方を学校のように同じ授業をしても、お互い不満が溜まるばかりで良くないことは明らかです。

TechGYMではそれぞれ個人ごとに合った、自分のペースで学習を進めることができます。

そうすることにより習得スピードが早い方はどんどん理解しながら先に進めることができ、

習得スピードが遅い方はゆっくりと1つ1つを確実に理解しながら学習が進められます。

■習得スピードが遅いことは悪ではない

習得スピードの早い、遅い話をしましたが遅いことは悪ではありません。

例えば早く進む方の中で重要なところを理解しないまま（理解したと思い込んで）先に進んでしまった方がいたとします。

その方はその先で理解していないことが出てきたときに急にストップしてしまいます。

その時点ではなぜできないかを理解するまでに時間がかかる場合もあります。

遅く進んでいる方は1つ1つを確実に理解してから次に進むので学習が急にストップすることは少ないかもしれません。

もちろんこの話は極端な例え話かもしれません。

ここで言いたいことは「習得スピードが遅くても継続して学習をすれば必ずプログラミングができるようになる」ということです。

■継続は力なり

プログラミングを学習する上で最も重要なことは継続することです。

プログラミングの学習は数週間や数ヶ月で終わるものではありません。

継続してずっとやっていくものです。

最先端で活躍する現役のエンジニアのみなさんも常に新しいことを学んでいるのが現実で、

だからこそ有名企業からオファーが来たり、高単価な開発案件が依頼されたりするのです。

TechGYMはスポーツジムと同じです。

毎日通う方がいても良いと思いますし、週何日か、場合によっては月何日か、のように通う方もいます。

それぞれの生活のリズムがあるのでそれはまったく問題ありません。

大事なのはその生活リズムの中にTechGYMを入れて継続させることです。

学習を継続すれば必ずプログラミングができるようになります。

TechGYMは様々なカリキュラムがあり、これからも継続的に増やしていきますので、

みなさまの継続的なプログラミング学習の大きな手助けとなることをお約束します。

TechGYMでプログラミングを学び、プログラミングができる人生と一緒に歩んでいきましょう。

