Rails自走開発教育研究所

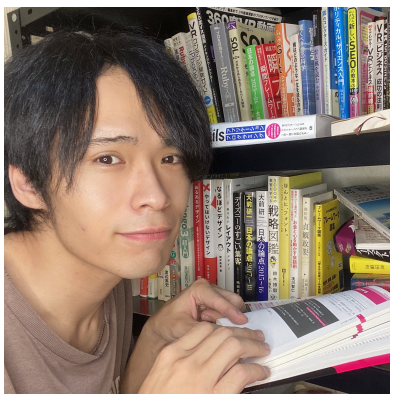
～ 教育訓練事業のご紹介 ～

作成者　：CyberTank
作成年月：2025年7月作成

目次

Hello, world. We are CyberTank !

私たちは、CyberTankです。よろしくお願いします！



CyberTank代表
平野夏光(25)

開業年月: 2022年4月
事業所　: 神奈川県横浜市泉区和泉町4889-3
電話番号: 090-8102-7272
メール　: H-natsumican@outlook.jp

弊社のプログラミングスクール

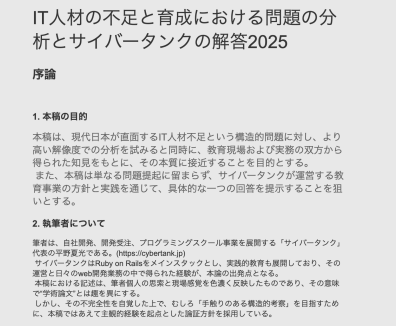
「Rails自走開発教育研究所」

について詳しく説明いたします。

Rails自走開発教育研究所とは？

- webエンジニアのスクールであり、その教育を研究する研究所でもあります。
- アカデミックな手法でIT人材不足問題と向き合い、教育に反映させています。

執筆論文
『IT人材の不足と育成における問題の分析と
サイバータンクの解答2025』
（著者：CyberTank代表　平野夏光,2025）



IT人材の不足と育成における問題の分析とサイバータンクの解答2025

序論

1. 本稿の目的

本稿は、現代日本が直面するIT人材不足という構造的問題に対し、より高い解像度での分析を試みると同時に、教育現場および実務の双方から得られた知見をもとに、その本質に接近することを目的とする。
また、本稿は単なる問題提起に留まらず、サイバータンクが運営する教育事業の方針と実践を通じて、具体的な一つの回答を提示することを狙いとする。

2. 執筆者について

筆者は、自社開発、開発受注、プログラミングスクール事業を展開する「サイバータンク」代表の平野夏光である。(https://cybertank.jp)
サイバータンクはRuby on Railsをメインスタックとし、実践的教育も展開しており、その運営と日々のweb開発業務の中で得られた経験が、本論の出発点となる。
本稿における記述は、筆者個人の思索と現場感覚を色濃く反映したものであり、その意味で学術論文とは趣を異にする。
しかし、その不完全性を自覚した上で、むしろ「手触りのある構造的考察」を目指すために、本稿ではあえて主観的経験を起点とした論証方針を採用している。

1. 調査力育成の必要性

我々のスクールの目標は、

「調査力の育成」

です。

知識よりも**調査力**こそが

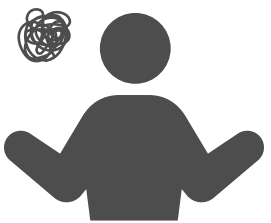
エンジニアの技術力だからです

一般的なスクール（＝知識を詰め込む教育）の卒業生がよくぶつかる壁①

現場でよく見る他社スクール卒業生

え、スクールでやってない

何がわからないかわからない…



誰も教えてくれないんだけど

調べてもよく分かんない

⇒「自走力の無いエンジニア」が量産されている。

他社スクール生にありがちな勘違い

スクールに通えば、なんでも作れるようになる！というのは<u>間違いです。</u>

プログラミングは
超広大な技術の広野

×

現場によって
採用技術が千差万別

Point

スクールで全てを教え切るのは「**不可能**」

もしスクールで知識だけを詰め込んだとしても...

スクールで
頑張ってきました！！

現場の暗黙のルール

本来なら相談ややり取りで自然に身につくはずの暗黙のルールを拾えず、チームに合わせた動きができなくなる。

柔軟性に欠ける

環境や仕様に変更が入ったときに
「前のやり方しかわからない…」となる

チーム基準から外れた実装

プロジェクト固有のルールを把握せず進めるので、
成果物がチームの基準と合わなくなる。

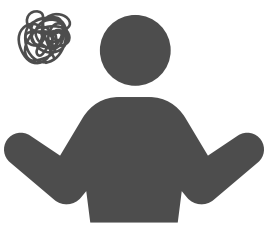
エンジニア環境では知識に限らず
未知の領域への対応力も求められる

一般的なスクール（＝知識を詰め込む教育）の卒業生がよくぶつかる壁②

現場でよく見る他社スクール卒業生

え、スクールでやってない

何がわからないかわからない…



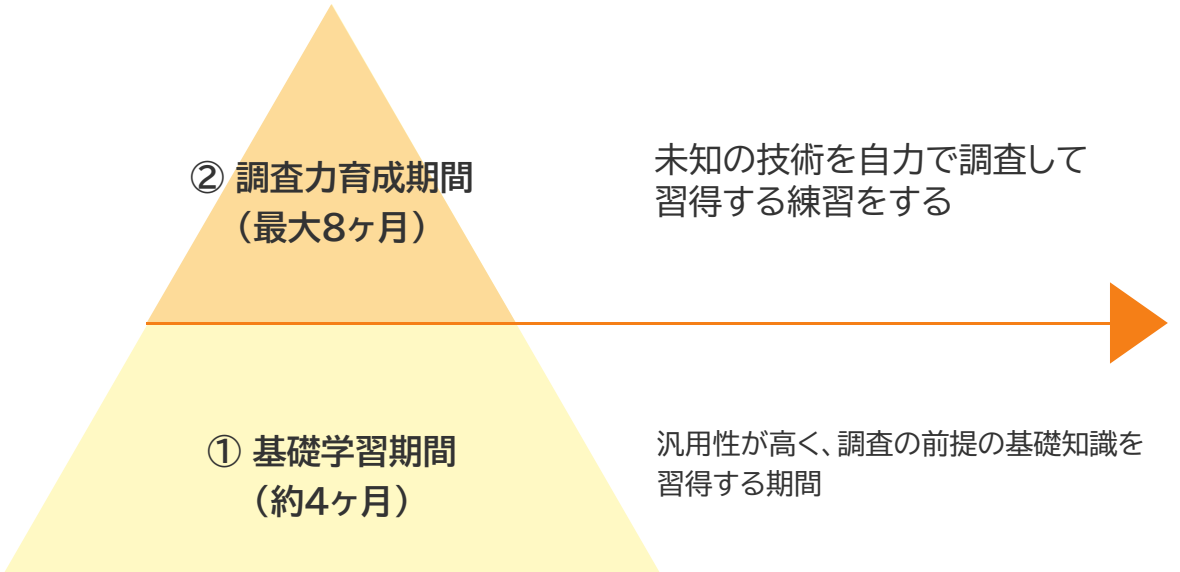
誰も教えてくれないんだけど

調べてもよく分かんない

⇒未知の領域に踏み込む力がない

未知の領域を進むことができる
調査力が必要です

未知の領域を進むためには



未知の技術を自力で調査して
習得する練習をする

汎用性が高く、調査の前提の基礎知識を
習得する期間

どんな技術も
自力で習得できる
状態を目指す。

2.調査力を身に着けるカリキュラム

私たちは

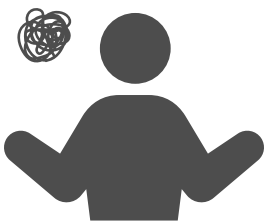
調査力＝問題特定能力

であると考えています。

現場でよく見る他社スクール卒業生

え、スクールでやってない

何がわからないかわからない…



誰も教えてくれないんだけど

調べてもよく分かんない

⇒問題特定能力が無い

問題特定能力を得るためには何をするの？

実装手順書

作業前の言語化訓練、
手順の可視化

**実装課題の
映像提出**

講義で扱った構造の
理解チェック

**Techブログの
義務化**

未知の領域に対して
の思考過程の可視化

「思考の見える化」により**問題特定能力**を獲得

調査力を育むプロセス：2段階構成のカリキュラム

問題特定に必要な基礎知識を座学期間で習得します。

① 座学期間
最初の約４ヶ月
講師による授業
確認テスト
質問力育成課題
Techブログ投稿練習

② 調査力育成期間
座学期間終了後最大８ヶ月間
自由実装
自力での技術調査
講師からの調査内容チェック
講師によるTechブログの添削

基礎的知識の徹底

調査力をつけるための実装指導

【モデルについての基礎】
1. モデル・カラム・テーブル・レコード
2. カラム型・データ型・予約列の説明（＋主キーの導入）
3. モデルの作成とモデルファイル（コンソールの使い方と cd コマンド）
4. テーブルの作成、マイグレーションの不可逆性とスキーマ
5. レコードの作成と保存（new, save, create, all）
6. ルーティングとアクションの関係
7. データベースとモデル（ActiveRecord とマイグレーションの基本）
8. ER図の見方と作り方
9. アソシエーションとデータの関連付け（has_many, belongs_to など）
10. Railsコンソールでのデバッグ（binding.irb, logger.debug など）
【動的な Web とは？】
11. 問題提起：「会員が増えるたびにルートやビューを増やすのは現実的？」
12. 変数の使い方（n段階目ビュー → アクション → n+1段階目ビューへの受け渡し）
【単一 MVC での CRUD を完成させよう！】
13. n段階目ビューでの n+1段階目への情報セット（link_to の動的パス）
14. n+1段階目アクションでの情報取得と変数定義・代入
15. ビューファイルでの表現（each による繰り返しと動的リンク）
16. Bookモデルの create アクション・form_with・redirect_to・ストロングパラメータ
17. Bookモデルの update アクションと form_with の挙動
18. ルートの命名規約と resources、destroy の実装、バリデーションの追加
【単純な複数モデルの MVC を作ってみよう！】
19. Userモデルの作成と外部キー追加（user:references、add_column）
20. 外部キーの挙動を体験しよう（user_id の割り当てと保存）
21. ユーザーのマイページ作成（外部キーで Book 一覧を取得）
22. アソシエーション 1:n（has_many, belongs_to, dependent: :destroy）
23. アソシエーションメソッドを使って books/index と show に作者リンクを追加
24. session変数でログイン機能を作り、CRUD に権限確認機能を追加
【複雑な複数モデルの MVC を作ってみよう！】
25. お気に入り機能の問題提起（なぜ n:m が必要なのか？）
26. 中間テーブルの作成とアソシエーション定義（Favorite モデル）
27. お気に入り登録機能の実装（create と << の比較）
28. お気に入り一覧ページの作成とお気に入り解除機能

⇨実装フェーズの学習へ

3. 講師の教え方について

私たちは、

「それは何に使うのか」
「それで何ができるのか」

を教えるから、分かりやすい。

「IPアドレス」についての普通の説明例

IPアドレスは、インターネットプロトコル(IP)を使用して通信する際に、各機器を識別するために使用される番号です。

分かりやすさの簡単な例

「IPアドレス」についての当スクール流の説明例

IPアドレスは、「デバイスを特定するための番号」。
IPアドレスがあることで、デバイスが特定できるので、通信を正しい相手に届けられるようになります。

講師の授業準備

「何に使うのか」「何ができるのか」という「目的の説明」を準備します。

POINT 01 今日は〇〇を教えます

POINT 02 これは、〇〇の時に出てきます

POINT 03 関連技術は〇〇や〇〇です

POINT 04 **これらで〇〇や〇〇ができるようになります**

POINT 05 文法はこうです

4. 学習に不安を感じている人へ

初学者たちがぶつかりがちな壁とその対策

学習カリキュラムについていけるか不安	学習を継続できるか不安	専門的な知識がないと難しいんじゃないか
3種類のテスト	**バディ制**	**直感的な理解**
確認テスト　再テスト　応用テストの3種類のテストを用意しています。 学習者の進捗に合わせて使い分け、知識定着まで伴走します。	学習者同士のバディ制を取り入れています。 一緒に学習できる仲間がそこにいます。	すべての学習者に対して、平等にゼロからの学習を提供します。 Ruby on Rails は理系出身者でなくても直感的な理解が可能です。

私たちの準備

調査力習得のために

教育の属人化を防ぐ

- 「指導フロー」の体系化により、内容の抜け漏れを防ぎ、理解しやすい講義内容を準備しています。
- 各講義前に講師に「指導フロー」に則っているかの確認を義務付け、フィードバックを行うことで、講義内容の品質担保を行なっています。

教育体制を見直し続ける仕組み

- 講師のBSCを設定しています。
- 感情、環境、集中度合いなどの要素まで含めた日報を作成し、日々お互いの行動改善をしています。

経験に裏打ちされた共感力

- 心がうまく動かない時期を経験したスタッフも在籍しております。
- 進捗に応じた声かけ・再調整を通じて、無理なく学習と向き合える環境を提供しています。

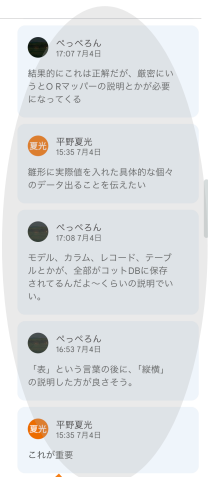
私たちの準備のご紹介

講義準備資料の例

モデルとレコードで、アプリ内で取り扱う情報を、取り扱いやすい形で保存できるようになる。
ex)Userモデルを作っておくことで、Userの年齢や名前などの情報をまとめて取り扱える。
テーブルで、そのデータを探しやすくしたり、見つけて変更を加えたりすることができる。
ex)Userのデータは、usersテーブルにあって、そのカラムの値とかでレコードを特定して修正したりできる。→excelでデータを整理することで得られるメリットに近いイメージ。

a. モデル
   i. 情報量が増えてきて、特定のモデルのカラムが増えていきそうな時、該当部分を新たなモデルとして切り分けると管理しやすい（この表現だと伝わらなさそう）
b. カラム
   i. カラムを増やすと、そのモデルに登録できる情報が増える
   ii. 例えばUserにbirthdayというカラムを追加すると、そのUserの誕生日を記録することができるようになる
c. レコード
   i. できることというよりは、新規データ登録をするとレコードは勝手に増えていく
d. テーブル
e. アソシエーション
   i. そのユーザーが書いた本をメソッド一つで呼び出したい時、
   ii. UserモデルとBookモデルのアソシエーションを記述すると、userが書いたbooksを簡単に呼び出すことができるようになる
f. バリデーション
   i. 「本を登録する時に、著者が未入力の時はエラーにするよ」といった設定を追加しりできる
g. メソッド
   i. ユーザーの年齢をメソッド一つで呼び出せるようになる
h. enum型
   i. その本の販売の状態について管理できるようになる
   ii. 「発売前」「発売中」「販売終了」とか。

5. 文法
   a. モデルを新しく作成する時
      i. どこに書くか
         1. ターミナル上
      ii. 何を書くか
         1. 作成するモデル名、どのようなカラムを追加するのかを書く
         2. カラムはモデル作成時にターミナル上で記入しなくても後から追加できる



講師が提出した「指導フロー」に関して、講義意図や講義内容について、講師同士でフィードバックを行なっています。

3種類のテストの例

「確認テスト」

第1回

第1回 確認テスト（モデル・カラム・テーブル・レコード）

Q1. モデルに関して正しいものを選びなさい。（単一選択）

A. モデルはデータベースの行を指すRails用語である
B. モデルはデータベースとのやり取りを担うクラスであり、MVCのMに相当する
C. モデルはRailsでは必ずsnake_caseで記述される

→ **正解：B**

Q2. カラムについての説明として最も適切なものはどれですか？（単一選択）

A. カラムはデータベースの縦方向の情報を指す
B. カラムはMVCのCに該当する構成要素
C. カラムはモデルの属性を定義し、テーブルの横方向に追加される

→ **正解：C**

Q3. テーブル・レコードの関係性について正しいものを選べ。（複数選択可）

A. テーブルはレコードの集合体である
B. レコードはテーブルの横方向に追加される
C. 1つのレコードは複数のカラムを持つ

→ **正解：A, C**

Q4. レコードとは何か？1文で説明しなさい（記述式）

→ **例回答**：テーブルの1行に相当し、各カラムの情報が具体的に入力された個別のデータのこと。

Q5. モデルとテーブルの役割の違いを説明しなさい（記述式）

→ **例回答**：モデルはデータのやりとりをプログラム上で扱うためのクラスであり、テーブルはそのデータを保存するデータベース上の表。

モデルについて説明しなさい
→カラムによって構成され、レコードの雛形としての役割を担う、データ保持・作成の型
カラムについて説明しなさい
・モデルを構成する要素
・テーブルを構成する要素
・レコードを構成する要素
テーブルについて説明しなさい
レコードについて説明しなさい

1番い部分はなんですか？(モデル、カラム、テーブル、レコード)

User

id	name	height	address
1	佐藤太郎	170	東京都
2	加藤次郎	160	北海道
3	田中三郎	180	沖縄県

「再テスト」

第8回 再テスト（復習・意味論再確認）

Q1. 次のうち、Railsで自動生成されるコントローラクラス名として正しいものはどれですか？（単一選択）

A. bookController.rb
B. BookscontrollerrB
C. booksControllerrb
D. BooksControllerrb

→ **正解：D**

Q2. モデルがUserの場合、対応するコントローラ名で正しいものを選びなさい。（単一選択）

A. user_controller.rb
B. UserControllers
C. UsersController
D. userController

→ **正解：C**

Q3. コントローラがモデルと対応していることのメリットを1つ挙げて簡潔に説明しなさい。（記述式）

→ **例解答：**

データの種類ごとに処理を整理でき、MVC構造がわかりやすく保守性も高まる。

Q4. Railsの命名規則に従わなかった場合に起こりうる問題を1つ簡単に述べなさい。（記述式）

→ **例解答：**

ビューやルーティングなどが正しく関連付けられず、エラーになったり意図した動作にならなくなる。

「応用テスト」

第9回 応用問題（統語論的理解）

Q1.

create アクションで render :new が実行されるのはどのような状況か？（記述式）

→ **例解答：**

createアクションに失敗、バリデーションに失敗して保存できなかったとき。再度フォームを表示して入力を促すために new ビューを描画する。

Q2. 以下のアクションが動作したあとに表示される画面はどれか。理由も含めて説明しなさい。

```
def create
  @book = Book.new(book_params)
  @book.save
end
```

→ **例解答：**

ビューファイルがないため、エラー（MissingTemplate）が発生する可能性がある。redirect_to または render を使って画面遷移を明示すべき。

Q3.

render とredirect_to の使い分けを説明し、具体的なコード例を1つ挙げなさい。（記述式）

→ **例解答：**

render は同じリクエスト内でビューを表示し、redirect_to は別のアクションにリダイレクトする。

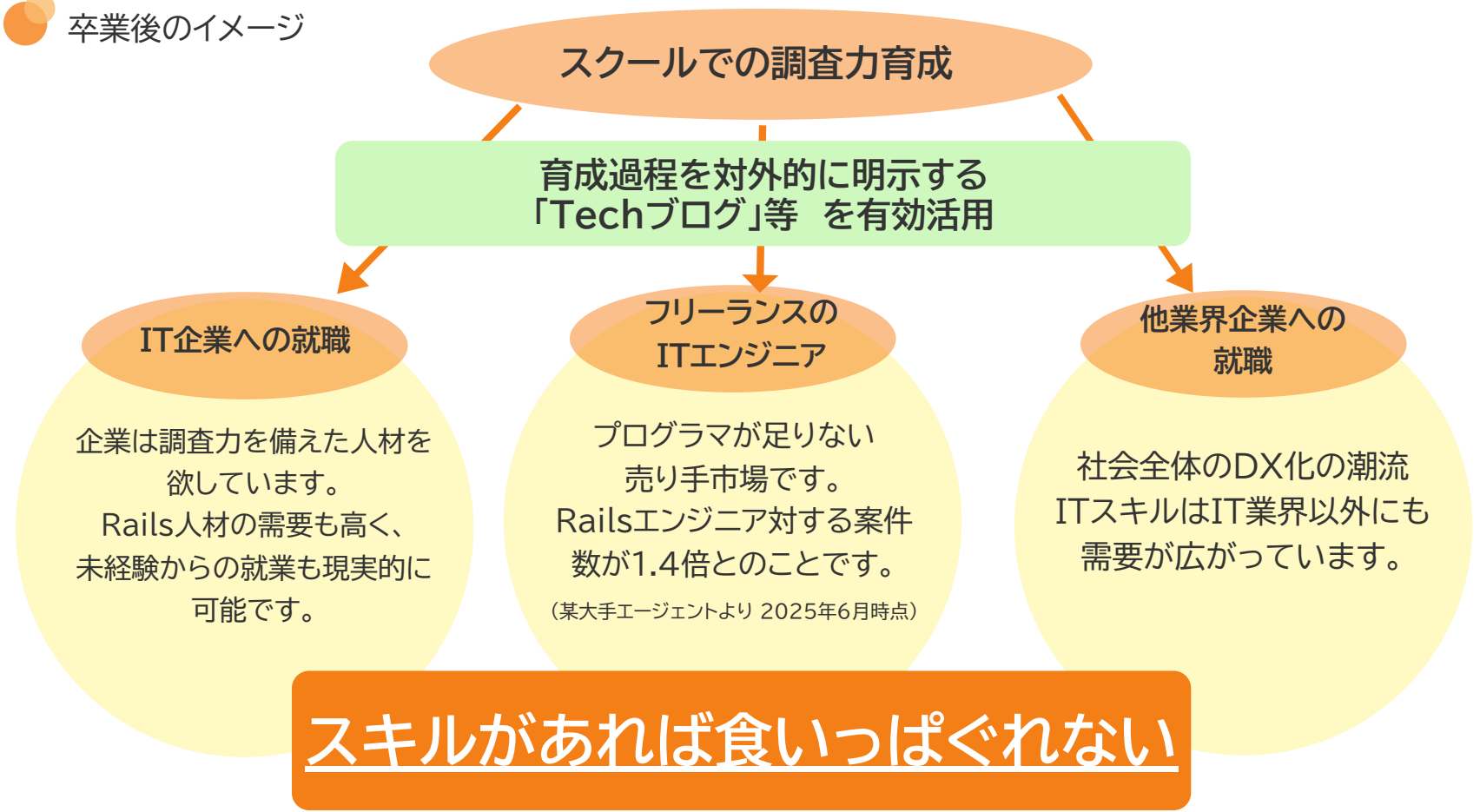
例：

```
if @book.save
  redirect_to books_path
else
  render :new
end
```

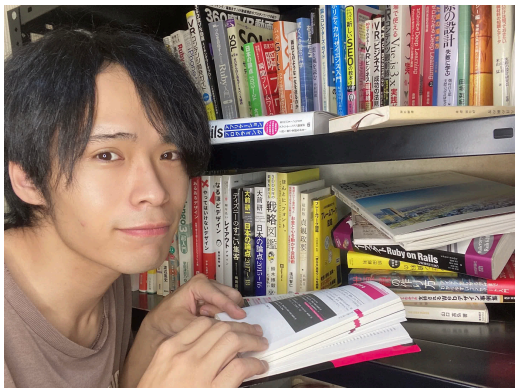
テスト種別
内容
確認テスト
アクションとビューの対応ルール、POST系の例外性の理解
再テスト
認識の補強、構造への再接続
応用問題
render/redirect_to の使い分け、動作結果とビューの挙動の説明力

5. 卒業後のイメージ

卒業後のイメージ


スクールでの調査力育成
育成過程を対外的に明示する
「Techブログ」等　を有効活用
IT企業への就職
企業は調査力を備えた人材を
欲しています。
Rails人材の需要も高く、
未経験からの就業も現実的に
可能です。
フリーランスの
ITエンジニア
プログラマが足りない
売り手市場です。
Railsエンジニア対する案件
数が1.4倍とのことです。
（某大手エージェントより 2025年6月時点）
他業界企業への
就職
社会全体のDX化の潮流
ITスキルはIT業界以外にも
需要が広がっています。
スキルがあれば食いっぱぐれない

6. 学習を始めるあなたへ



どう生きたいのですか？

選択とは、人生の表現です。
職業の選択も、確かにあなたの「人生の表現」なのです。

あなたの「エンジニアになりたい」という発露が、
すでに「人生の表現」の一部であると私は思います。

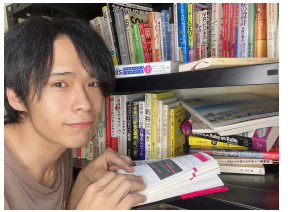
そう思うからこそ、我々には大きな責任があり、その責任と向き合う義務があります。
教育とは、その人生＝「あなたの表現」に触れることだからです。
だから我々は、「研究所」として、教育の「内容・質・方法」と多面的に学術的に向き合っているのです。

あなたは選択に集中して下さい。
やりたいと思ったら、あとは任せて下さい。

CyberTank代表　自走開発教育研究所長　　平野夏光

運営者情報

本資料の作成、Rails自走開発教育研究所は、CyberTankがしています。



代表　平野夏光

サイバータンク概要

- 代表者 平野夏光
- 開業 2022年4月

取引実績

- レバテック株式会社
- 株式会社マイナビ
- 株式会社ギミック
- 株式会社アピリッツ 他

事業内容

- web開発案件受注事業
- webサービス開発、運営事業
- プログラミングスクール事業

お問い合わせ

- 代表平野電話番号　09081027272
- メールアドレス H-natsumican@outlook.jp
- HP https://cybertank.jp

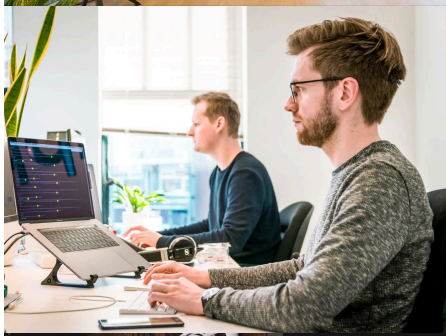
ご興味があれば、いつでもご連絡ください。



Thank you !

あなたの良き成長を
Cyber Tankは祈っています。

お問い合わせは
090-8102-7272(代表直通）へどうぞ


クイズ

Messageモデルの中から、1番目のデータを探すためのコードを選びなさい

①Message.find(1)

②Message.find_by(id: 1)

③Message.where.not(created_at: nil).first

④Message.all.first

クイズ

！！！！！！！！！！全部正解！！！！！！！！！！

これらのコードを入力してみると、、、？

```
=> #<Message id: 1, message: "Hello, world!">
```