

# エンジニア特化 職務経歴書テンプレート集

Web / インフラ / アプリ 職種別カスタムテンプレート

成果数字の入れ方・スキルスタック記述例 付き

## セクション1：本テンプレートの使い方と3つのポイント

### 1. 成果数字を必ず入れる理由と書き方

- 定量的な成果は書類選考の評価を30%向上させます。
- 単に「～を実施」と書くのではなく、「～により〇〇%改善」と結果まで記述します。

### 2. スキルスタック記述順ルール

- 実務経験年数が長い順、または得意な順に記載します。
- レベル表記を統一します（例：実務3年／精通／基礎レベル）。

### 3. 明光キャリアパートナーズ推奨フォーマット

- STAR法（状況・課題・行動・結果）での記述を意識した構成です。
- 採用担当者が「5秒」でスキルレベルを把握できる視認性を重視しています。

## セクション2：【職種①】Webエンジニア 職務経歴書テンプレート

氏名	明光 太郎	生年月日	199X年X月X日
連絡先	090-XXXX-XXXX	Email	taro.meiko@example.com
GitHub / URL	https://github.com/meiko-taro / https://my-portfolio.com		

## 職務要約

記入例:

PHP、JavaScriptを用いたWebアプリケーション開発に5年従事。ECサイトの新規開発からリプレースまで幅広く経験。直近ではマイクロサービス化プロジェクトのリードエンジニアとして、システムのレスポンス時間を40%改善。

## 活かせるスキル・スキルスタック

フロントエンド	記入例: React (実務3年)、Vue.js (実務2年)、TypeScript (実務3年)
バックエンド	記入例: PHP (実務5年/精通)、Node.js (実務2年)、Python (基礎レベル)
データベース	記入例: MySQL (実務5年)、PostgreSQL (実務2年)、Redis (実務1年)
インフラ	記入例: AWS (実務3年)、Docker (実務2年)、Kubernetes (学習中)
ツール	記入例: Git、GitHub Actions、Jenkins、Slack

## 職務経歴

期間	20XX年XX月 ~ 20XX年XX月 (X年Xヶ月)
会社名	株式会社〇〇〇〇 (正社員)
役割	バックエンドエンジニア → リードエンジニア
プロジェクト内容	記入例: ECサイトのマイクロサービス化プロジェクト (チーム5名)
担当業務	記入例: <ul style="list-style-type: none"><li>モノリシックアーキテクチャからマイクロサービスへの段階的移行設計</li><li>API設計・実装 (REST API 20エンドポイント)</li><li>パフォーマンスチューニング (クエリ最適化、キャッシュ戦略)</li></ul>
実績・成果	記入例: <ul style="list-style-type: none"><li>システムレスポンス時間を平均2.5秒から1.5秒に短縮 (40%改善)</li><li>サーバーコストを月額50万円から35万円に削減 (30%削減)</li><li>障害発生率を月5件から月1件以下に低減</li></ul>

# 成果数字の入れ方ガイド (Webエンジニア編)

NG例 (数字なし)	OK例 (数字あり)
Webサイトのパフォーマンスを改善	ページ読み込み速度を3.2秒から1.8秒に短縮 (44%改善)
バグ修正を実施	月平均20件のバグを発見・修正し、品質向上に貢献
チーム開発をリード	5名のチームをリードし、予定より2週間早くリリース達成

## 自己PR

記入例:

ユーザー体験の向上とシステムの安定性向上を両立させることに強みがあります。直近のプロジェクトでは、パフォーマンス改善により離脱率を15%削減し、売上向上に貢献しました。技術選定では、チームのスキルセットと保守性を重視し、長期的な視点での判断を心がけています。

## セクション3: 【職種②】インフラエンジニア 職務経歴書テンプレート

### 職務要約

記入例:

AWSを中心としたクラウドインフラの設計・構築・運用に4年従事。オンプレミスからクラウドへの移行プロジェクトを複数経験し、可用性99.9%以上のシステム運用を実現。IaCによる自動化推進により、構築時間を60%短縮。

### スキルスタック (インフラ特化)

OS	Linux (RHEL/CentOS、実務5年/精通)、Ubuntu (実務3年)、Windows Server (実務2年)
クラウド	AWS (実務4年/精通)、GCP (実務1年)、Azure (基礎レベル)
ネットワーク	VPC設計、ルーティング、ロードバランサー、CDN
セキュリティ	IAM設計、セキュリティグループ、WAF、脆弱性診断
監視・ログ	CloudWatch、Datadog (実務2年)、ELKスタック (実務1年)
IaC/自動化	Terraform (実務2年)、Ansible (実務2年)、CloudFormation

# 職務経歴（設計・構築・運用フェーズ別）

設計フェーズ	<ul style="list-style-type: none"><li>可用性・拡張性を考慮したAWSインフラアーキテクチャ設計</li><li>マルチAZ構成、Auto Scaling設計</li></ul>
構築フェーズ	<ul style="list-style-type: none"><li>TerraformによるIaC実装（300リソース以上）</li><li>CI/CDパイプライン構築（GitHub Actions + AWS CodeDeploy）</li></ul>
運用フェーズ	<ul style="list-style-type: none"><li>24時間365日監視体制の構築</li><li>インシデント対応（平均復旧時間15分以内）</li></ul>

## 成果数字の入れ方ガイド（インフラ特化）

指標	記入例
稼働率	可用性99.95%を達成（年間ダウンタイム4.38時間以内）
コスト削減	インスタンス最適化により月額コストを80万円から50万円に削減（37.5%削減）
障害対応	平均復旧時間（MTTR）を45分から15分に短縮（67%改善）
自動化	手動構築時間を8時間から3時間に短縮（62.5%削減）

## 自己PR

記入例：

安定稼働と継続的改善を両立させるインフラ運用に強みがあります。障害発生時の迅速な原因特定・復旧対応だけでなく、再発防止のための仕組み化を徹底しています。TerraformによるIaC推進により、属人化を排除し、チーム全体の生産性向上に貢献しました。

## セクション4：【職種③】アプリエンジニア（モバイル）職務経歴書テンプレート

## 職務要約

記入例：

iOS/Androidアプリ開発に4年従事。ネイティブ開発からReact Nativeを用いたクロスプラットフォーム開発まで幅広く経験。累計ダウンロード数100万以上のアプリ開発に携わり、ユーザー評価4.5以上を維持。

## スキルスタック（モバイル特化）

iOS	Swift (実務3年/精通)、Objective-C (実務1年)、SwiftUI (実務1年)
Android	Kotlin (実務3年/精通)、Java (実務2年)、Jetpack Compose (実務1年)
クロスPF	React Native (実務2年)、Flutter (学習中)
API連携	REST API連携、GraphQL (実務1年)
ツール	Xcode、Android Studio、Firebase、App Store Connect

## 職務経歴（アプリ開発プロジェクト例）

プロジェクト名	フィットネス管理アプリ (iOS/Android)
担当範囲	<ul style="list-style-type: none"><li>UI/UX実装 (Swift/Kotlin)</li><li>RESTful API連携実装</li><li>プッシュ通知機能実装 (Firebase Cloud Messaging)</li><li>アプリ内課金機能実装</li></ul>
実績・成果	<ul style="list-style-type: none"><li>App Store評価4.6、Google Play評価4.5を獲得</li><li>リリース後3ヶ月で10万ダウンロード達成</li><li>クラッシュ率0.5%以下を維持 (業界平均2%)</li><li>アプリ起動速度を2.5秒から1.2秒に改善 (52%高速化)</li></ul>

## 成果数字の入れ方ガイド（アプリ特化）

指標	記入例
ダウンロード数	リリース後6ヶ月で累計50万ダウンロード達成
ユーザー評価	App Store評価4.7 (1,200件のレビュー)
リリース実績	iOS/Android合計15回のメジャーリリース、40回のマイナーアップデート
パフォーマンス	アプリサイズを45MBから28MBに削減 (38%削減)

## セクション5：成果数字の入れ方 完全ガイド（共通）

### NG例 → OK例 の対比表

NG例（抽象的）	OK例（具体的・定量的）	ポイント
システムを改善しました	データベースクエリ最適化により、検索処理時間を5秒から0.8秒に短縮 (84%改	具体的な数値と改善率

	善)	
チームをリードしました	7名のチームをリードし、スプリント速度を週20storyから30storyに向上 (50%向上)	チーム規模と成果指標
品質向上に貢献しました	コードレビュー文化を定着させ、バグ検出数を月平均8件から3件に削減 (62.5%削減)	具体的施策と結果
新機能を開発しました	ユーザー要望上位の機能を開発し、DAUを2万人から2.8万人に増加 (40%増加)	ビジネスインパクト
コスト削減しました	不要なリソースの削除とインスタンス最適化により、月額費用を120万円から75万円に削減 (37.5%削減)	具体的金額

## 数字が出しにくい場合の対処法

- 期間・回数で表現：例：「週次でデプロイ実施 (年間50回以上)」
- 規模・範囲で表現：例：「50万行のレガシーコードをリファクタリング」
- 比較・改善度で表現：例：「手動テストから自動テスト導入により、テスト時間を1/3に短縮」
- 頻度・継続性で表現：例：「月次で技術勉強会を主催 (累計12回開催、参加者延べ80名)」

## 明光キャリアパートナーズ推奨「STAR法」記述フレームワーク




<b>S</b> (Situation)	<b>状況・背景：</b> 例：「ユーザー数急増により、サーバーレスポンスが遅延 (平均3秒)」
<b>T</b> (Task)	<b>課題：</b> 例：「1秒以内のレスポンスタイム達成が必要」
<b>A</b> (Action)	<b>行動：</b> 例：「データベースインデックス最適化、N+1問題解消、キャッシュ戦略導入」
<b>R</b> (Result)	<b>結果：</b> 例：「レスポンスタイムを平均0.8秒に短縮 (73%改善)、離脱率を20%から12%に低減」

# セクション6：スキルスタック 明光推奨記述ルール

## 記述順・レベル表記の統一ルール

レベル	表記方法	目安
精通レベル	「実務〇年／精通」	3年以上、他者に教えられる、設計からできる
実務レベル	「実務〇年」	1～3年、実務で問題なく使える
基礎レベル	「基礎レベル」または「実務〇ヶ月」	1年未満、基本的な機能は使える
学習中	「学習中」	個人学習、実務経験なし

### 「業務経験あり／個人開発あり／学習中」の使い分け

-  **業務経験あり**：実務プロジェクトで使用、コードレビューを経ている
-  **個人開発あり**：個人プロジェクトやOSS貢献で使用（GitHubに公開推奨）
-  **学習中**：書籍・Udemy等で学習、簡単なサンプルコードを書いた程度

### スキル記述のNGパターン

- 「多数の言語を扱えます」→ 具体的な言語名と経験年数を記載
- 「HTML/CSS/JavaScript/React/Vue/Angular...」→ 経験年数順に整理
- 「基本的なプログラミングスキル」→ 抽象的表現は避ける

 本テンプレートは明光キャリアパートナーズ利用者限定の特典資料です