プログラミング質問ガイドライン

　プログラミングを学ぶ人が他人に質問するとき、円滑に問題を解決するために伝えておくと良いことをまとめました。書ける範囲、わかる範囲で良いので、これらを伝えて質問してみてください。

前提を伝える

　立ち向かっている問題について伝える前に、まずはご自身の状況について軽く教えてください。

何に関連する質問ですか？	例: 課題 / 研究 / 趣味	この問題を解決すること自体に意味があるのか、何かやりたいことがあってこのトラブルを早急に解決したいのかを知りたいです。
期限はありますか？	例: X月Y日まで / 期限なし	急ぎの用なのかを伝えましょう。
ご自身の技量は？	例: 完全初心者 / 授業で触った程度 / ガンガン書きます	技量の推定ミスによる無駄なやり取りをなくしましょう。過去の制作物があれば紹介しても良いかもしれません。
余談は許しますか？	例: 余談歓迎 / 問題解決だけ教えてくれ	ついついテンションが上がって余計な発展的な話をしたくなることがあります。そうなった時には話しても良いですか…？

基本事項を伝える

　いくらプログラミングが得意な人でも、いきなりコードを見せられては何もわかりません。何をしたくてどんなコードを書いているのかを伝えましょう。

どんな内容の質問ですか？	例: 書いたコードにエラーが出る / 環境構築で躓いている / やりたいことはあるが技術選定ができない	どんなところで困っているのかを軽く伝えましょう。軽くで良いです。
何を作りたいですか？	例: ○○を××するアルゴリズムを実装したい / ○○をするスマホアプリを作りたい	コードだけでは何がしたいか判断できないことがあります。何をしたいかを簡潔に伝えましょう。
言語/フレームワークは何ですか？	例: C++ / Python / Flutter	もしかしたら回答者があまりよく知らない言語やフレームワークかもしれません。それでも回答者は頑張るので、とりあえず伝えておきましょう。

詳細を伝える

　ここでやっと本題です。コードやエラーを見せましょう。問題解決に向けて、一番大事なところです。

どこまで自分で作ってみましたか？	例: 愚直に書いてみた / プロトタイプを作ってみた / がむしゃらに書いてみた	今までの進捗を知ることで、回答者が一層この問題について詳しく理解できます。
何が問題ですか？	例: エラーが出る / バグがある / 動くが遅い	問題の種類によって着眼すべき点が大きく変わります。
想定する動きは何ですか？	例: Xという入力に対してYと出力してほしい	何を想定しているのか具体的にわかると、回答者が手元で実験しやすいです。
該当コードを貼ってください	例: コードをコピペ / コードを公開したGitHubレポジトリのリンク	回答者がそのままコピペできるよう、テキスト形式だと望ましいです。さらに、可能であれば問題が再現する最小構成のコードが嬉しいです。
実行結果を貼ってください	例: エラー文 / 間違った出力	こちらもテキスト形式だと望ましいです。隠したい個人情報は隠してください。

　この画像は再配布していただいて構いません。画像のソースはhttps://github.com/Nyanyan/Programming_Questionにて公開しています。プルリク歓迎です。コードはMITライセンスです。