ExSeedアウトプット講義〜設計図見本〜

企画・目標の設定

ユーザーゴール

* 電卓アプリを作り、計算業務の改善をしたい
* 電卓として基本的な機能が使えるのはもちろんのことできるだけ使いやすいUIを提供したい

要件定義

機能要件

* 電卓のように数r時を入力することができる。（大きい位から、数字を一個ずつ入力していく方式、先頭の０は表示されない）
* 小数点が表示できる。
* 四則演算を使える
* クリアボタンで数字の入力をクリアできる。（それまでの計算は保持）
* オールクリアボタンで、計算の全ての情報をクリアする。
* 四則演算のボタンは２つ目の数字を押す前に、最後に押したものが使われる。（演算子が変更可能）

非機能要件

* 直感的で理解しやすいUI。本物の電卓と同じ感じ
* 計算履歴がアプリ外に送信されない
* 最低限のレスポンス速度（とてつもなく遅いなんてことがない）

内部設計　〜ユーザー視点での挙動〜

機能設計

* 電卓と同じようなUI・表示（先頭の0を表示しない）
  + 本物の電卓と同じように０〜９の数字と四則演算の記号、小数点ボタンがあり、対応した操作が可能。
  + 数字なら、右に数字が追加されていくかのように表示。四則演算の記号を押されることで数字の入力が中断されるイメージ
  + 0を最初にいくら入れても０が一桁だけしか表示されない。
* 小数点が表示できる
  + 計算結果が小数点以下になる時もそのまま表示する。
* 四則演算が使える。
  + 演算子のボタンが連続して押されている場合は処理をせずに、上書きする形。
  + 最後に入力した演算子が適用されるようにする。
  + 数字が入力され始めたら、演算子はそこで固定されて、計算結果はこの演算子を元に行われる。
* クリアボタンで数字の入力をクリア
  + 現在扱っているのが数字ならそれを0にするってイメージ
* オールクリアボタンで、全ての情報をクリア
  + 現在保持している数字や演算子を初期化する。

内部設計　〜具体的にどう実装するか〜

データ設計

* 入力をnum1:str、num2:str、operator:strで構成されたオブジェクトに入れて扱う。
* これに、sign:intとかも加えて、signによってどこに入力を入れるかを制御するとかがいいかも。
  + sign = 0の時、num1に数字を追加していくモード、
  + sign = 1、演算子の受付モード、
  + sign = 2、num2に数字を追加していくモード。みたいな

モジュール設計

* 数字ボタンまたは小数点ボタンが押された時に反応する関数（num:str）->str:
  + 現在のsignを元に、num1かnum2に文字列を追加していく。
    - sign = 0なら、num1に数字を追加。
    - sign = 1なら、signを2にしてからnum2に入力。
    - sign = 2なら、num2に数字を追加。
  + 数字は文字列で扱い、「文字列を繋げる関数」で後ろに文字を繋げていく形で管理する。
  + 「表示する数字の文字列を作る関数」を使うことで、先頭の0を省くなどして、実際に表示するための文字列を返してもらう。
* 文字列を繋げる関数（current\_num:str, num:str）->str:
  + 現在扱ってる文字列と、入力された数字の文字を繋げる。
* 表示する数字の文字列を作る関数（current\_num:str）->str:
  + sign = 0,1の時はnum1を表示する。
  + sign = 2の時はnum2を表示する。
  + 表示するnumの文字列に小数点が含まれていたらfloat、含まれていなければint型に変換して、またstr型に戻す手順を踏む。
  + これで、先頭に0がついている場合でも正常に表示できるようにする。
* 演算子ボタンが押された時に反応する関数（operator:str）->str:
  + sign = 0の場合、数字の入力を終了するため、sign = 1にして、演算子を記憶する。
  + sign = 1の場合、既存の演算子を新しいものに更新する。
  + sign = 2の場合、計算を行い、signを1に更新、
    - 既存の演算子を元に、計算処理を行い、計算結果をnum1に格納。
    - num2を初期化した後、新しい演算子を記憶する。
    - =の場合は、num2の初期化と演算子の記憶を行わず、計算結果の格納のみを行う。
    - 返り値は、num1に「表示する数字の文字列を作る関数」を通したもの固定。
* クリアボタンが押された時に反応する関数()->str:
  + sign = 0の場合、num1の値を初期化する。
  + sign = 1の場合、何もせずに待機。
  + sign = 2の場合、num2のみを初期化、つまり演算子とnum1を保持したまま。
  + 返り値は空の文字列。
* オールクリアボタンが押された時に反応する関数()->str:
  + オブジェクトが保有している情報を全て初期化する。
  + 返り値は空の文字列

アプリ部分

* GradioのBlocksで、outputとしてTextbookinputは指定せずに、各ボタンに対応する関数を呼び出す形にする。
  + ※.clickではinputに文字列そのまま入れることができないため
* 配置としては.Columnと.Rowうまく使って表示させる。