

Documentation du Projet : Calculatrice PHP

1. Introduction

Ce projet consiste en la création d'une calculatrice web développée en PHP. Elle utilise les sessions pour stocker et manipuler les données de calcul, ce qui permet de conserver l'état des opérations en cours. Cette calculatrice intègre les opérations arithmétiques de base (addition, soustraction, multiplication, division), ainsi que le calcul de pourcentage et la gestion des nombres décimaux.

2. Structure du Projet

Le projet est organisé en plusieurs fichiers distincts :

- `index.php` : Interface utilisateur de la calculatrice.
- `c_index.php` : Contrôleur gérant les opérations et les sessions.
- `style.css` : Feuille de style pour l'apparence de la calculatrice.

3. Interface Utilisateur

L'interface de la calculatrice a été pensée pour être simple et intuitive.

```
<div class="calculator-container">
  <!-- Affichage -->
  <input class="calculator-display" type="text" name="display" value="<?php echo $_SESSION['result']; ?>" readonly>

  <!-- Boutons -->
  <form action="c_index.php" method="POST" class="calculator-buttons">
    <button type="submit" name="btn" value="C" class="calculator-button c">C</button>
    <button type="submit" name="btn" value="%" class="calculator-button operation">%</button>
    <button type="submit" name="btn" value="/" class="calculator-button operation">/</button>
    <button type="submit" name="btn" value="*" class="calculator-button operation">x</button>
    <button type="submit" name="btn" value="7" class="calculator-button">7</button>
    <button type="submit" name="btn" value="8" class="calculator-button">8</button>
    <button type="submit" name="btn" value="9" class="calculator-button">9</button>
    <button type="submit" name="btn" value="-" class="calculator-button operation">-</button>
    <button type="submit" name="btn" value="4" class="calculator-button">4</button>
    <button type="submit" name="btn" value="5" class="calculator-button">5</button>
    <button type="submit" name="btn" value="6" class="calculator-button">6</button>
    <button type="submit" name="btn" value="+" class="calculator-button operation">+</button>
    <button type="submit" name="btn" value="1" class="calculator-button">1</button>
    <button type="submit" name="btn" value="2" class="calculator-button">2</button>
    <button type="submit" name="btn" value="3" class="calculator-button">3</button>
    <button type="submit" name="btn" value="." class="calculator-button operation">.</button>
    <button type="submit" name="btn" value="0" class="calculator-button">0</button>
    <button type="submit" name="btn" value="=" class="calculator-button equal">=</button>
  </form>
</div>
```

Un mod debug a été conçu à l'origine pour tester la logique de mon raisonnement, je l'ai maintenu et stylisé afin de mieux présenter l'application.

```
<div class="debug-container">
  <p>$result:<?php var_dump($_SESSION['result']); ?></p>
  <p>$op:<?php var_dump($_SESSION['OP']); ?></p>
  <p>$nb:<?php var_dump($_SESSION['NB']); ?></p>
</div>
```

4. Style et Conception

Le design utilise des boutons arrondis et des couleurs distinctes.

```
/* Affichage */
.calculator-display {
  width: 100%;
  font-size: 1.5em;
  color: #ff3b3b;
  background-color: #f4f4f4;
  border: none;
  border-radius: 10px;
  text-align: right;
  padding: 10px;
  margin-bottom: 20px;
  box-shadow: inset 0 2px 4px rgba(0, 0, 0, 0.1);
}

.calculator-display:focus {
  outline: none;
}

/* Grille des boutons */
.calculator-buttons {
  display: grid;
  grid-template-columns: repeat(4, 1fr);
  gap: 10px;
}

/* Boutons */
.calculator-button {
  background-color: #ffffff;
  border: 1px solid #e0e0e0;
  border-radius: 50%;
  height: 60px;
  width: 60px;
  font-size: 1.2em;
  font-weight: bold;
  color: #666;
  cursor: pointer;
  box-shadow: 0 4px 6px rgba(0, 0, 0, 0.1);
  transition: all 0.2s ease;
}
```

5. Logique de Fonctionnement

Le fichier c_index.php gère toute la logique du projet.

5.1 Gestion des Sessions

```
if (!isset($_SESSION['result'])){$_SESSION['result'] = '';}
if (!isset($_SESSION['OP'])) {$_SESSION['OP'] = '';} // Debug
if (!isset($_SESSION['NB'])) {$_SESSION['NB'] = 0.0;} // Debug
```

5.2 Traitement des Opérations

La fonction operation(\$op) exécute l'opération choisie par l'utilisateur en fonction de l'opérateur stocké dans la session.

Les valeurs sont stockées dans \$_SESSION['NB'] (le premier nombre saisi), \$_SESSION['result'] (le deuxième nombre ou le résultat intermédiaire), et \$_SESSION['OP'] (l'opérateur sélectionné).

```
function operation($op) {
    switch ($_SESSION['OP']){
        case '/':
            if ((float)$_SESSION['result'] == 0) {
                $_SESSION['result'] = 'Infinity';
            } else {
                $_SESSION['result'] = (float)$_SESSION['NB'] / (float)$_SESSION['result'];
                $_SESSION['NB'] = 0.0;
                $_SESSION['OP'] = '';
                break;
            }
        case '*':
            $_SESSION['result'] = (float)$_SESSION['NB'] * (float)$_SESSION['result'];
            $_SESSION['NB'] = 0.0;
            $_SESSION['OP'] = '';
            break;
        case '-':
            $_SESSION['result'] = (float)$_SESSION['NB'] - (float)$_SESSION['result'];
            $_SESSION['NB'] = 0.0;
            $_SESSION['OP'] = '';
            break;
        case '+':
            $_SESSION['result'] = (float)$_SESSION['NB'] + (float)$_SESSION['result'];
            $_SESSION['NB'] = 0.0;
            $_SESSION['OP'] = '';
            break;
    }
    $_SESSION['NB'] = 0.0;
    $_SESSION['OP'] = '';
}
```

Division (/) : Vérifie si le diviseur est zéro pour éviter une erreur et affiche "Infinity" si c'est le cas. Multiplication (*), Soustraction (-), Addition (+) : Effectue simplement l'opération mathématique correspondante. Après chaque calcul, NB est remis à 0.0 et l'opérateur est effacé pour préparer une nouvelle opération. Ce code permet donc d'enchaîner des calculs tout en stockant les résultats dans la session pour garder la continuité des opérations.

5.3 Gestion des Boutons d'Opération

La fonction bouton(\$btn) stocke l'opérateur sélectionné dans \$_SESSION['OP'] et prépare les valeurs pour le calcul.

Stockage de l'opérateur : La variable \$btn (correspondant au bouton pressé) est enregistrée dans \$_SESSION['OP'].

Gestion du pourcentage (%) : Si \$_SESSION['result'] contient %, la valeur est convertie en décimal ($NB = result / 100$) et result est réinitialisé.

Gestion de l'infini : Si le résultat précédent est "Infinity", la fonction s'arrête immédiatement.

Stockage du nombre : Sinon, NB prend la valeur du résultat convertie en flottant, et result est effacé pour la prochaine saisie

```
function bouton($btn) {
    $_SESSION['OP'] = $btn;
    if (str_contains($_SESSION['result'], '%')){
        $_SESSION['NB'] = (float)$_SESSION['result'] / 100;
        $_SESSION['result'] = '';
    }
    else if ($_SESSION['result'] === 'Infinity'){
        return;
    }
    else {
        $_SESSION['NB'] = (float)$_SESSION['result'];
        $_SESSION['result'] = '';
    }
}
```

.5.4 Traitement des Entrées Utilisateur

Ici, je gère les entrées de l'utilisateur lorsqu'il appuie sur un bouton de la calculatrice.

Récupération du bouton pressé : Si un bouton est soumis via POST, sa valeur est stockée dans \$btn.

Gestion des chiffres (0-9) : Si l'utilisateur entre un chiffre et que le résultat actuel n'est pas "Infinity", il est ajouté à \$_SESSION['result'].

Gestion des opérations :

Réinitialisation (CL) : Réinitialise complètement la session pour remettre la calculatrice à zéro grâce à session_unset() et session_destroy().

Gestion du pourcentage (%) : Si result n'est pas "Infinity", ajoute % uniquement si ce symbole ou un point décimal n'est pas déjà présent.

```
if (isset($_POST['btn'])) {$btn = $_POST['btn'];}

if ($btn >=0 && $btn <= 9){
    if($_SESSION['result'] != 'Infinity'){$_SESSION['result'] .= $btn;}
}
else{
    switch ($btn){
        case 'CL':
            session_unset();
            session_destroy();
            session_start();
            break;
        case '%':
            if($_SESSION['result'] != 'Infinity'){
                if (str_contains($_SESSION['result'], '%') || str_contains($_SESSION['result'], '.')){
                }
                else {$_SESSION['result'] .= '%';}
                break;
            }
        case '/':
            if($_SESSION['result'] != 'Infinity'){
                bouton($btn);
                break;}
        case '*':
            if($_SESSION['result'] != 'Infinity'){
                bouton($btn);
                break;}
        case '-':
            if($_SESSION['result'] != 'Infinity'){
                bouton($btn);
                break;}
        case '+':
            if($_SESSION['result'] != 'Infinity'){
                bouton($btn);
                break;}
        case '.':
            if($_SESSION['result'] != 'Infinity'){
                if (str_contains($_SESSION['result'], '%') || str_contains($_SESSION['result'], '.')){
                }
                else {$_SESSION['result'] .= '.';}
                break;}
        case '=':
```

6. Fonctionnalités Principales

- Opérations arithmétiques de base
- Calcul de pourcentage
- Gestion des décimaux
- Gestion des erreurs (division par zéro)
- Mode Debug

7. Améliorations à Venir

- Enchaîner les calculs sans appuyer sur le bouton égal.
- Réutiliser le bouton égal pour répéter l'opération précédente.
- Correction de bugs mineurs.

8. Méthodologie de Développement

1. Création de l'interface.
2. Implémentation des fonctions de base.
3. Gestion des sessions.
4. Traitement des cas spéciaux.
5. Ajout du mode debug.

9. Conclusion

Ce projet démontre une application web fonctionnelle utilisant les sessions PHP pour maintenir l'état des calculs entre les interactions.