多目的臨床データ登録システム(MCDRS)チュートリアル 応用編

チュートリアル応用編の概要

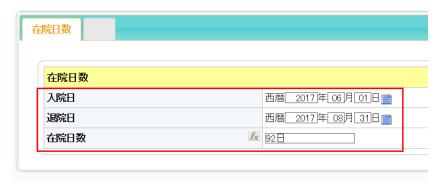
チュートリアル応用編では「テーブル型項目の作成」、「カスタム演算(在院日数、検査時の年齢)」について手順を説明します。

チュートリアルで作成する症例入力画面

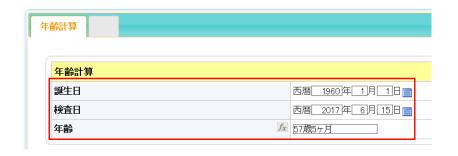
テーブル型項目



「入院日」、「退院日」から「在院日数」を算出



「誕生日」、「検査日」から検査日の「年齢」を算出



テーブル型で項目:検査(「検査日付」、「検査項目名」、「検査値」)を作成します。

既にグループが登録されている場合は2.へ

1. グループ作成をクリックします。





3. 各入力項目を入力します。

項目名称 : 検査

データ型:テーブル型

4. 追加をクリックします。

5. 各入力項目を入力します。

項目名称 : 検査日付

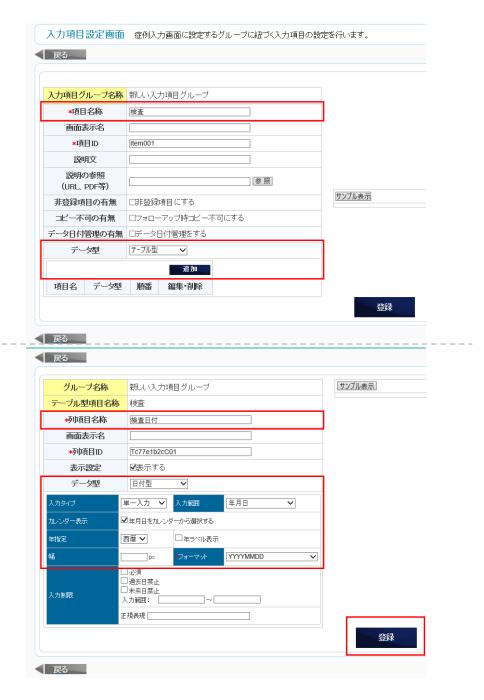
データ型:日付型

入力タイプ :単一入力

入力範囲:年月日

カレンダー表示:チェックを入れます

フォーマット : YYYYMMDD



- 7. 項目「検査日付」が登録されていることを確認します。
- 8. 追加をクリックします。

9. 各入力項目を入力します。

項目名称 : 検査項目名 データ型 : 文字列型





- 11. 項目「検査項目名」が登録されていることを確認します。
- 12. 追加をクリックします。

13. 各入力項目を入力します。

項目名称:検査値データ型:文字列型





- 15. 項目「検査値」が登録されていることを確認します。
- 16. 登録をクリックします。



- 17. データ型「テーブル型」の項目があることを確認します。
- 18. 登録をクリックします。



入力項目「入院日」と「退院日」を日付型で作成します。 出力項目「在院日数」を文字列型または単位付き項目型で作成します。

既にグループが登録されている場合は2.へ

1. グループ作成をクリックします。





3. 各入力項目を入力して、登録をクリックします。

項目名称: 入院日データ型: 日付型入力タイプ: 年月日別入力範囲: 年月日

カレンダー表示:チェックを入れます

年指定:西暦

年ラベル表示:チェックを入れます

4. 登録をクリックします。



入力項目グループ名称 新しい入力項目グループ

退院日

*項目名称

画面表示名

在院日数を算出する

6. 各入力項目を入力して、登録をクリックします。

項目名称: 退院日データ型: 日付型入力タイプ: 年月日別

入力範囲:年月日

カレンダー表示:チェックを入れます

年指定:西暦

年ラベル表示:チェックを入れます

7. 登録をクリックします。

*項目ID Item002 説明文 説明の参照 参照 (URL、PDF等) サンプル表示 非登録項目の有無 □非登録項目にする 西暦[_____年[__月[__日 | コピー不可の有無 □フォローアップ時コピー不可にする データ日付管理の有無 ロデータ日付管理をする データ型 日付型 年月日別 🗸 入力範囲 年月日 ☑ 年月日をカレンダーから選択する 加ンダー表示 初期表示 □ 現在日時を設定する ☑年ラベル表示 西暦 🗸 丰指定 □必須 □過去日禁止 □ 未来日禁止 人力制限 入力範囲: 正規表現 登録 く 戻る



9. 各入力項目を入力して、登録をクリックします。

項目名称:在院日数

データ型:文字列型(単位付き数値型<単位:日>も可)

10. 登録をクリックします。





タブ設定済みのグループに「入院日」、「退院日」、 「在院日数」項目がある場合は14.へ

12. タブ設定をクリックします。



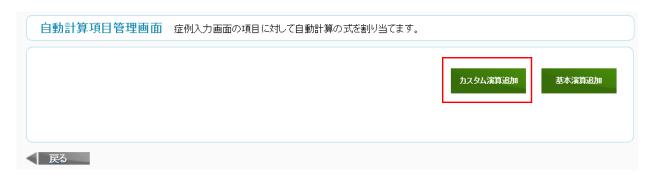
13. タブ追加をクリックして、「入院日」、「退院日」、「在院日数」項目があるグループをタブ登録します。



14. 自動計算項目をクリックします。



15. カスタム演算追加をクリックします。



16. 各入力項目を入力・選択します。

名称:在院日数計算

算出結果項目:在院日数

タイミング: 自動

計算結果編集の有無 :チェックを入れます

- 17. 対象項目から「入院日」と「退院日」を選択して、 それぞれ追加をクリックします。
- 18. args[0]=「入院日」, args[1]=「退院日」となるように 並び順を変更します。



19. 下記のコード(赤字)を演算式へ入力します。

```
//入力項目は入院日args[0], 退院日args[1]とする
//入退院日は日付型のYYYYMMDD, YYYY年MM月DD日とする
//出力項目は文字列型または単位付き数値型とする
//入院日args[0],退院日args[1]を文字列型にする
args[0]=String(args[0]);
args[1]=String(args[1]);
```



```
//入院日と退院日が8桁の場合、項目「入院日」「退院日」から日付を変数に格納する
//入院日と退院日が8桁以外の場合、NULL(空文字)を出力する
if(args[0].length==8 && args[1].length==8){
    //入院日をYYYY、MM、DDに分解
    var NyuinYear = args[0].substr(0,4);
    var NyuinMonth = args[0].substr(4,2) - 1; //下記new Date()は、1月は0、2月は1,…で表すため
    var NyuinDay = args[0].substr(6,2);

//日付計算するために、new Date()で入院日の日付を取得
    var Nyuin = new Date(NyuinYear,NyuinMonth,NyuinDay);
```

```
//退院日をYYYY. MM. DDに分解
  var TaiinYear = args[1].substr(0,4);
  var TaiinMonth = args[1].substr(4,2) - 1; //下記new Date()は、1月は0, 2月は1,・・・で表すため
  var TaiinDay = args[1].substr(6,2);
  //日付計算するために、new Date()で退院日の日付を取得
  var Taiin = new Date(TaiinYear, TaiinMonth, TaiinDay);
  //入院日が退院日より後の日付の場合、NULL(空文字)を出力する
  //正しく入退院日が設定されている場合、在院日数を算出する(Taiin - Nyuinの単位はミリ秒)
  if(Nyuin>Taiin){
    var Ans = "":
  lelse
    var Ans = ((Taiin - Nvuin)/(1000*60*60*24))+ 1 + "日"; //単位付き数値型の場合「+ "日"」は入力しない
}else{
  var Ans = "":
//算出結果を算出結果項目に出力する
return Ans;
```

- 20. チェックをクリックして、演算式が正しく入力されているかチェックします。
- 21. 登録をクリックします。



22. カスタム演算が登録されていることを確認して、戻るをクリックします。



入力項目「誕生日」と「検査日」を日付型で作成します。 出力項目「年齢」を文字列型で作成します。

既にグループが登録されている場合は2.へ

1. グループ作成をクリックします。





3. 各入力項目を入力して、登録をクリックします。

項目名称 : 誕生日 データ型 : 日付型 入力タイプ : 年月日別

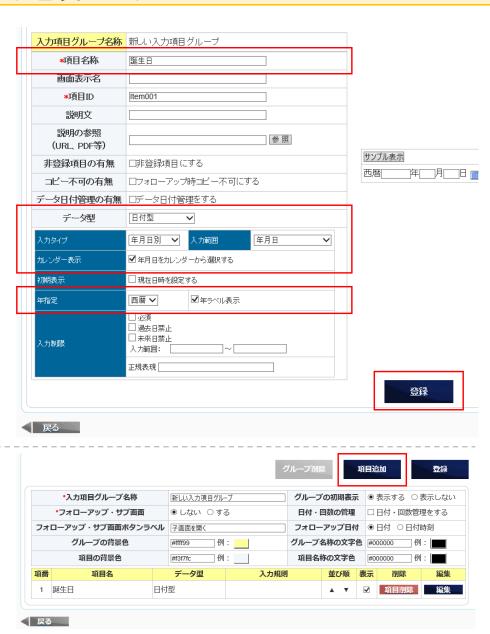
入力範囲:年月日

カレンダー表示:チェックを入れます

年指定:西暦

年ラベル表示:チェックを入れます

4. 登録をクリックします。



6. 各入力項目を入力して、登録をクリックします。

項目名称 : 検査日 データ型 : 日付型 入力タイプ : 年月日別

入力範囲:年月日

カレンダー表示:チェックを入れます

年指定:西暦

年ラベル表示 :チェックを入れます

7. 登録をクリックします。

入力項目グループ名称 新しい入力項目グループ *項目名称 検査日 画面表示名 Item002 *項目ID 説明文 説明の参照 参照 (URL、PDF等) サンプル表示 □非登録項目にする 非登録項目の有無 □フォローアップ時コビー不可にする コピー不可の有無 データ日付管理の有無 □データ日付管理をする データ型 日付型 年月日別 🗸 入力範囲 年月日 ルンダー表示 ☑ 年月日をカレンダーから選択する 加斯表示 □ 現在日時を設定する 西暦 🗸 ▼年ラベル表示 丰指定 □必須 □過去日禁止 □ 未来日禁止 人力制限 入力範囲: 正規表現「 登録 ■ 戻る



9. 各入力項目を入力して、登録をクリックします。

項目名称 :年齡

データ型:文字列型

10. 登録をクリックします。





タブ設定済みのグループに「誕生日」、「検査日」、 「年齢」項目がある場合は14.へ

12. タブ設定をクリックします。

13. タブ追加をクリックして、「誕生日」、「検査日」、「年齢」項目があるグループをタブ登録します。





14. 自動計算項目をクリックします。



15. カスタム演算追加をクリックします。

自動計算項目管理画面	症例入力画面の項目に対して自動計算の式を割り当てます。		
		もった」を整合to	##SPECha
		カスタム演算追加	基本演算追加
戻る アン・アン・アン・アン・アン・アン・アン・アン・アン・アン・アン・アン・アン・ア			

16. 各入力項目を入力・選択します。

名称:年齢計算

算出結果項目:年齢 タイミング:自動

計算結果編集の有無 :チェックを入れます

- 17. 対象項目から「誕生日」と「検査日」を選択して、 それぞれ追加をクリックします。
- 18. args[0]=「誕生日」, args[1]=「検査日」となるように 並び順を変更します。



```
19. 下記のコード(<del>赤字・</del>青字)を演算式へ入力します。
                                      赤字のみ:〇歳 赤字+青字:〇歳〇ヶ月
//入力項目は誕生日args[0], 検査日args[1]とする
//入退院日は日付型のYYYYMMDD, YYYY年MM月DD日とする
//出力項目は文字列型とする
//誕生日args[0],検査日args[1]を文字列型にする
args[0]=String(args[0]):
args[1]=String(args[1]);
//誕生日と検査日が8桁の場合、項目「誕生日」「検査日」から「〇年〇ヶ月」を算出する
//誕生日と検査日が8桁以外の場合、NULL(空文字)を出力する
if(args[0].length==8 && args[1].length==8){
  //入力項目から関数(calcAge, calcMonth)により「〇歳〇ヶ月」を算出する。
  //calcAge()は「〇歳」、calcMonth()は「〇ヶ月」を算出する
  //算出した結果を変数AgeとMonthに代入
  var Age = calcAge(args[0],args[1]);
  var Month = calcMonth(args[0],args[1]);
  //算出結果の判定
  //誕生日が検査日より後の時、NULL(空文字)を出力する
  //(AgeとMonthが負の時)
  if(Age>=0 && Month>=0){
    //変数AgeとMonthのデータ型を数値型から文字列型へ変換
    Age = String(Age);
    Month = String(Month);
```

```
//出力項目に算出結果を出力する「〇歳〇ヶ月」
    return Age + "歳" + Month + "ヶ月";
  }else{
    return "":
}else{
  return ´
//「〇歳」算出
//引数「誕生日,検査日」戻り値「〇歳」
function calcAge(BirthDate, KensaDate) {
  //データ型を数値型にする
  BirthDate = Number(BirthDate):
  KensaDate = Number(KensaDate):
  //(検査日-誕生日)/10000で「○歳」を算出
  //Math.floorは計算結果の小数点以下を切り捨てる
  return (Math.floor((KensaDate - BirthDate) / 10000));
```

```
//「〇ヶ月」算出
//引数「誕生日,検査日」戻り値「〇ヶ月」
function calcMonth(BirthDate,KensaDate){
  //変数宣言 誕生日の月日, 検査日の月日
  var MonBirth = 0:
  var MonKensa = 0:
  var DayBirth = 0;
  var DavKensa = 0:
  //変数宣言 戻り値ans 初期値999
  var ans = 999:
  //誕生日または検査日の月、日を抜き出し
  MonBirth = String(BirthDate);
  MonBirth = MonBirth.substr(4.2):
  MonBirth = Number(MonBirth):
  MonKensa = String(KensaDate):
  MonKensa = MonKensa.substr(4.2):
  MonKensa = Number(MonKensa):
  DayBirth = String(BirthDate);
  DayBirth = DayBirth.substr(6,2);
  DayBirth = Number(DayBirth);
  DayKensa = String(KensaDate);
  DayKensa = DayKensa.substr(6,2);
  DayKensa = Number(DayKensa);
```

```
//検査日の月〈誕生日の月かどうか
if(MonKensa < MonBirth){
  //検査日の日〈誕生日の日かどうか
  if(DayKensa < DayBirth){
    ans = 12 - MonBirth + MonKensa -1;
  lelse
    ans = 12 - MonBirth + MonKensa:
lelse{
//検査日の月 = 誕生日の月かどうか
if(MonKensa == MonBirth){
  //検査日の日〈誕生日の日かどうか
  if(DayKensa < DayBirth){
    ans = 11;
  }else{
    ans = 0:
}else{
```

```
//検査日の月 > 誕生日の月かどうか
if(MonKensa > MonBirth){

//検査日の日 < 誕生日の日かどうか
if(DayKensa < DayBirth){
    ans = MonKensa - MonBirth -1;
    }else{
    ans = MonKensa - MonBirth;
    }
}else{
}

return ans;
}
```

20. チェックをクリックして、演算式が正しく入力されているかチェックします。

21. 登録をクリックします。



22. カスタム演算が登録されていることを確認して、戻るをクリックします。

