JavaScript 入門　テキスト課題　最終補足　例外処理

例外処理とはエラーが発生した場合に処理を中断し、エラーメッセージを表示したりする処理です。エラーは自分で発生させることもできます。

```
try{
   例外発生の可能性がある処理
   （自分でエラー発生させる場合 : throw new エラーオブジェクトの種類("エラーメッセージ")　）
}catch(e){
   例外発生時の処理（catch の引数にはエラーオブジェクトが入る。）
} finally{
   例外発生の有無に関わらず実行される処理（省略可能）
}
```

※エラーオブジェクトには Error（エラー全般）、TypeError（データ型のエラー）等がある。

保存時ファイル名：textb-reigai-sample-1.html、textb-reigai-1.html 等。

＜サンプルプログラム＞

```
try{
      let a = 1;
      alert(a.trim());
}catch(e){
      alert("name:" + e.name);
      alert("msg:" + e.message);
}

try{
      document.write("開始" + "<br>");
      throw new Error("わざとエラー");
}catch(e){
      alert("name:" + e.name);
      alert("msg:" + e.message);
}finally{
      document.write("終了" + "<br>");
}
```

・name：エラーオブジェクトの種類
・message：エラーメッセージ

・Error オブジェクトを発生させる

・Error が表示
・わざとエラーが表示

＜練習問題＞

練習１

存在しない関数である、parseNantoka()を実行しようとして下さい。

※ちょうどよい場所に例外処理を入れて下さい。例外発生時は Error オブジェクトの name と message を画面表示して下さい。finally で「終了」を画面表示して下さい。

正常結果例：「ReferenceError」「parseNantoka is not defined」「終了」が表示

練習２

関数 warizan を定義して下さい。引数１と引数２に数値が入り、引数１を引数２で割った値を戻り値にして下さい。引数２が 0 以下の場合は、エラーを発生（種別：Error、メッセージ：割り算エラー）させて下さい。

変数「kazu1」、「kazu2」を作成し、結果例を参考にデータを代入して下さい。
kazu を引数に指定し、warizan を呼び出して下さい。

※関数外のちょうどよい場所に例外処理を入れて下さい。例外発生時は Error オブジェクトの message を画面表示して下さい。

正常結果例：「6」「2」の時、「3」が表示
「6」「0」の時、「割り算エラー」が表示

練習３

関数 checkLength を定義して下さい。引数１に文字列が入り、引数１の文字列の長さ（文字列変数.length で取得可能）を戻り値にして下さい。引数１が空文字の場合は、エラーを発生（種別：Error、メッセージ：空文字エラー）させて下さい。

変数「moji」を作成し、結果例を参考にデータを代入して下さい。
moji を引数に指定し、checkLength を呼び出して下さい。checkLength の戻り値を画面表示して下さい。

※関数外のちょうどよい場所に例外処理を入れて下さい。例外発生時は Error オブジェクトの message を画面表示して下さい。

正常結果例：「こんにちは」の時、「5」が表示
「""（空文字）」の時、「空文字エラー」が表示