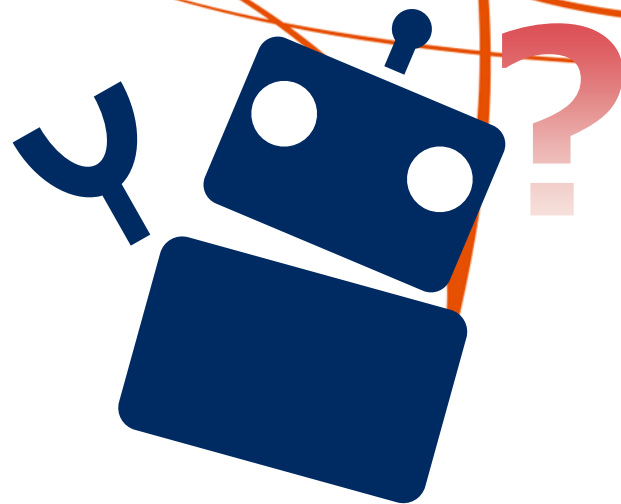
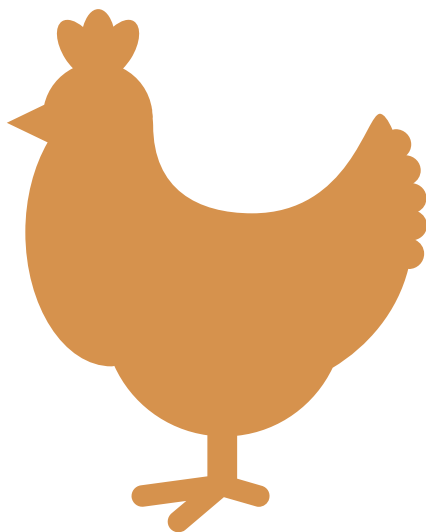


電ロボくんの家庭教師

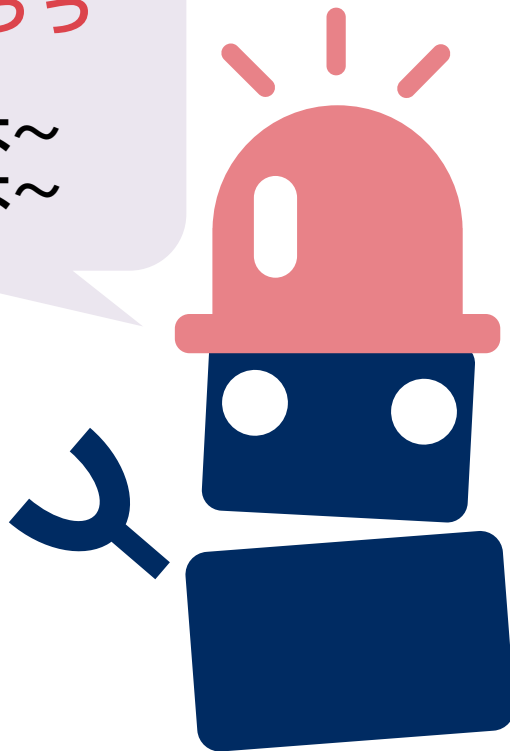
エラー処理ってなに？



あら、電ロボくん、パトカーごっこ？

う～う～うううう

ぼくの中で
エラーが起きたよ～
エラーが起きたよ～



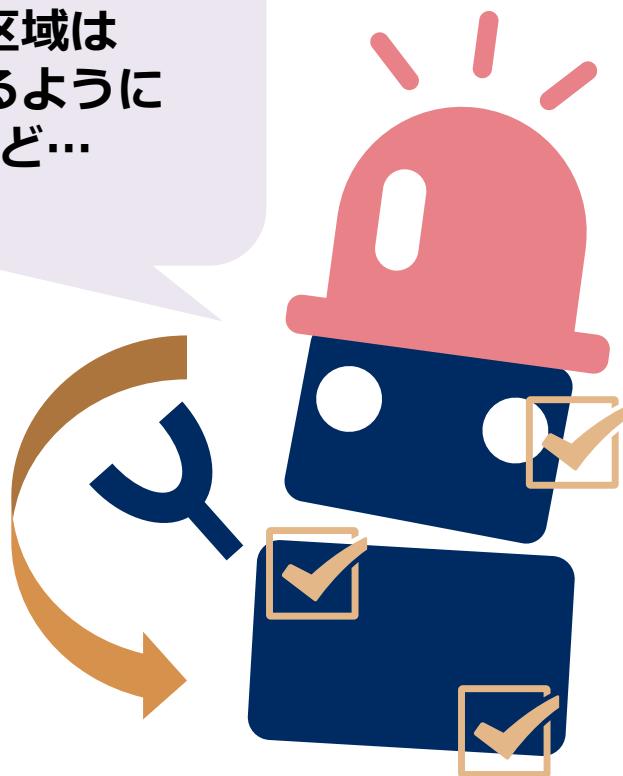
ERROR_CODEに
エラー番号
ERROR_MSGに
エラーメッセージを
保存し、メールで送った！

現場に急行せよ！！！！

確かにそうだったわね！

あれ、この区域は
エラー監視するように
教わったけど…

身体で
エラーが
起きてないか
じっと見る



エラーになっても ぼくが
自分の身体を監視しておけば

ダイアログを出して
止まることが無いから…って

これまでの会話、意味は分かりましたか？



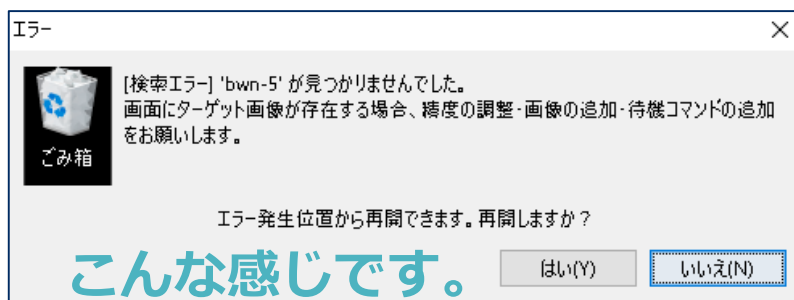
では、ひとセリフずつ
確認してみましょう！

まず、監視中にエラー発生した このセリフ！

う～う～うううう

ぼくの中で
エラーが起きたよ～
エラーが起きたよ～

監視なしの場合
RoboSolのコマンド内で
エラーが発生すると
ダイアログを出して 処理を止めます



こんな感じです。

処理が止まると
困りますよね

ERROR_CODEに
エラー番号
ERROR_MSGに
エラーメッセージを
保存し、メールで送った！

現場に急行せよ！！！！

監視中にエラーが発生すると
変数 ERROR_CODE に エラー番号
変数 ERROR_MSG に エラー内容
が 保存されます

ERROR_MSG	[検索エラー] 'bwn-5' が見つかりません...
@retval	
%userhome%	
ERROR_CODE	1

エラー番号1で
画像認識エラーだとわかります

次に、天の声に説明をするこのセリフ！

あれ、この区域は
エラー監視するように
教わったけど…

監視なしだとエラー発生時に
処理が止まって困るので
エラーが起きそうな区域では
監視をお願いしました

エラー処理 監視開始 Try >>		
クリック	監視区域	 M:95%
エラー処理 エラー発生時 << Catch >>		
メール送信		
エラー処理 終了 << End-Try		

エラーになっても ぼくが
自分の身体を監視しておけば

ダイアログを出して
止まることが無いから…って

監視しておけば
ダイアログで処理を止めず済みます

エラー処理 エラー発生時 << Catch >>		
メール送信		
エラー処理 終了 << End-Try		

<<Catch>>の中に
メール送信を追加すれば
処理を止めず通知できて安心です

ロボくに思いを伝える三種の神器！

エラーで処理を止めないため
もしもの備えに「エラー処理」！

繰り返し！



繰り返し



出現まで
繰り返し



条件で
繰り返し



CSVで
繰り返し

分岐！



分岐



条件分岐

分岐では
操作対象で起きた
事象に対処します

エラー処理！



エラー処理

分岐と違い
RoboSolコマンド
内で起きたエラーに
対処します