Relato de conversão de Adv/PL em TLPP (04/03/24)

Resumo

API `explain`

Bloco

Foi solicitado a explicação para o código:

```
for n := 1 to 5
  aAdd(aOpcoes, strZero(n,1,0))
next
```

Retornou (comentários no meio do retorno):

Para detalhes sobre array, veja documentação oficial.

> The for loop iterates from 1 to 5 and creates an array with the values "0", "1", "2", "3", "4".
> The next statement will be executed after the last iteration of the for loop.
> If you want to create an empty array, use the function Array() instead of aAdd().

Correto.

> If you want to add elements to an existing array, use the method Aadd() instead of aAdd().
> If you want to append one element at the end of an existing array, use the method Aadd() instead of aAdd().

Redundante.

> If you want to insert an element at a specific position, use the method ~~Ainsert()~~ instead of aAdd().

O correto é `aIns`.

> If you want to remove an element from an array, use the method Adel() instead of aAdd().
> If you want to get the length of an array, use the expression Len() instead of aAdd().
> If you want to sort an array, use the method Asort() instead of aAdd().
> If you want to reverse an array, use the method Areverse() instead of aAdd().
> If you want to ~~copy~~ an array, use the method Aclone() instead of aAdd().

Para cópia, usa-se `aCopy` que copia somente a primeira dimensão e mantém as demais como referência. O `aClone`, efetua uma cópia de todas as dimensões, eliminando referências, exceto para objetos.

> If you want to merge two arrays, use the method ~~Amerge()~~ instead of aAdd().

Oficialmente, não existe `aMerge`.

> If you want to find out if an element exists in an array, use the method Ascan() instead of aAdd().
> If you want to find out how many times an element occurs in an array, use the method Ascan() instead of aAdd().
> If you want to find out which positions an element occurs in an array, use the method Ascan() instead of aAdd().

O aScan, somente localiza elementos no array. Para as demais operações citadas, faz-se necessário aplicar alguns artifícios.

> If you want to find out what the highest value is in an array, use the method Max() instead of aAdd().
> If you want to find out what the lowest value is in an array, use the method Min() instead of aAdd().
> If you want to find out what the average value is in an array, use the function Avg() instead of aAdd().
> If you want to find out what the sum of all values is in an array, use the function Sum() instead of aAdd().
> If you want to find out what the median value is in an array, use the function Median() instead of aAdd().
> If you want to find out what the standard deviation is in an array, use the function StDev() instead of aAdd().
> If you want to find out what the variance is in an array, use the function Var() instead of aAdd().
> If you want to find out what the mode is in an array, use the function Mode() instead of aAdd().
> If you want to find out what the skewness is in an array, use the function Skew() instead of aAdd().
> If you want to find out what the kurtosis is in an array, use the function Kurt() instead of aAdd().
> If you want to find out what the entropy is in an array, use the function Entropy() instead of aAdd().
> If you want to find out what the correlation coefficient is between two arrays, use the function Corr() instead of aAdd().
> If you want to find out what the covariance between two arrays is, use the function Cov() instead of aAdd().
> If you want to find out what the linear regression line is between two arrays, use the function LinReg() instead of aAdd().
> If you want to find out what the logistic regression line is between two arrays, use the function Logistic() instead of aAdd> ().
> If you want to find out what the principal component analysis (PCA) is between two arrays, use the function PCA() instead of > aAdd().
> If you want to find out what the k-means clustering algorithm is between two arrays, use the function KMeans() instead of aAdd> ().
> If you want to find out what the hierarchical clustering algorithm is between two arrays, use the function HCluster() instead > of aAdd().
> If you want to find out what the nearest neighbor algorithm is between two arrays, use the function NearNeigh() instead of aAdd> ().
> If you want to find out what the decision tree learning algorithm is between two arrays, use the function DecisionTree() > instead of aAdd().
> If you want to find out what the random forest learning algorithm is between two arrays, use the function RandomForest() > instead of aAdd().
> If you want to find out what the support vector machine (SVM) is between two arrays, use the

> function SvmTrain() instead of > aAdd().
> If you want to find out what the naive bayes classifier is between two arrays, use the function Naive"

As funções citadas no bloco acima, não existem oficialmente. Para fazê-las requer processamento adicional e/ou aplicar alguns artifícios.