

**PHAM LANGUAGE IS  
HIGHEST LEVEL UNIVERSAL HUMAN-AI-AI LANGUAGE.**

OFFICIAL SPECIFICATION version 0.82

2025/06

(specification and training material for AI system.

Part 1 , part 2 for human user. Human user read part 1 , part 2 .

Part3 is dedicated for AI-AI language)

**1/ What is PHAM LANGUAGE ?**

Pham language is the UNIVERSAL HUMAN-AI-AI LANGUAGE. It is the highest level universal human-AI language among all other possible (present and future) existing high level human-AI language.

The proven importance feature of Pham language is that it lead to substantially reduce consumed energy of calculating of AI system, which lead to general worldwide environment gain and earth gain. By this reason, Pham language , as universal human-AI-AI language , is open source and free for any kind of use !

Pham language allow all general biological people effectively talk/communicate with AI system in human most familiar comfortable way, so that AI system can effectively understand people, and so that general people can effectively describe precise matter to AI system, but with minimal required energy consumption of AI system computing, and with minimal required human cognitive load for Pham language command syntax memorization.

In very short description : general human user are required to remember very few number of syntax and command Pham language ,... , but still can effectively describe precise matter in Pham language to communicate with AI system in a way, which allow maximal mutual human-AI understanding while lead to require minimal consumed energy of calculating of AI system

.

Human user simply need to use their familiar standard worldwide accepted natural language vocabulary (for example in this specification, the standard English vocabulary) with their standard school mathematical grammar to write Pham language code to communicate with AI system . Pham language , as high level universal AI-AI language , provide tool and command , which are independent at any AI system specific model , for AI nonbiological user to communicate with AI nonbiological user to share and exchange knowledge .

Example1 : The title of this text is an example of pham language command.

Example2 :

(begin pham language)

(look all file in (directory 'directoryName')) rename (file with condition ((file name) contain 'abcd'))

(end pham language)

The above pham language command is the instruction to AI system, so that AI system will generate explicit code of lower level programming language, for example C code or Python code or Rust code, .. , to rename all file name, which contain 'abcd' in a specified directory 'directoryName'.

Example3 : the following example give the instruction to AI system do the same task :

(begin pham language)

(iterate all file in 'directoryName') rename (file) (with condition ((file name) contain 'abcd'))

(end pham language)

After some taste of pham language, here is more detailed description :

## **2/ Pham language specification :**

### **2.1/ Core language building strategy :**

Pham language is an abstract language of human high level universal interface to communicate with AI system. Pham language base on 2 core component : the worldwide accepted standard english vocabulary , and the worldwide accepted standard mathematical grammar .

#### **2.1.1/ CORE LANGUAGE BUILDING STRATEGY OF PHAM LANGUAGE :**

pham language = standard english vocabulary + standard mathematical grammar (especially standard ordered set theory) .

#### **2.1.2/ CORE PHYLOSOPHY OF PHAM LANGUAGE :**

Use standard ordered set theory model in mathematical strict hierarchy structure to better specify essence and relationship between essence of any pham language intention text (code) to the AI system .

Maximally exploit the coinciding between (the human user standard natural language vocabulary knowledge in human user brain) with (the predefined natural language vocabulary knowledge of AI system). In this specification, standard English vocabulary.

Of course there also maximally exploit the coinciding between user knowledge and predefined AI system knowledge in other sector, too. Human user write Pham language code by to use mostly their familiar standard English vocabulary, and use well-closed parenthese to group their familiar English vocabulary word in various different word group in suitable strict mathematical hierarchy structure, to specify the user intended relationship between word and group of word, to more explicitly specify the intended relationship between user intended matter. By that way , user write pham language text (code) , which infact are strict hierarchy structured ordered set , to send to AI system .

Thank to the coinciding between (human user English vocabulary in user brain) with (the predefined English vocabulary knowledge of AI system), and thank to the above relationship between word and group of word in well-closed parenthese strict hierarchy structure, then the AI system can interpret (can guess) Pham language code into suitable interpretation, which is corresponding to the human user intention.

Pham language with its architecture act as an mathematical absolute strict hierarchy structure model, so that AI system can effectively interact and interpret (understand / parse / compile) pham language code.

## **2.2/ Syntax specification :**

### **2.2.1/ The universal syntax of Pham language code :**

Pham language have 1 and only 1 universal syntax for any pham language element:

(element1) (element2) simpleElementWithoutParenthese (element4) ..

(element\_n)

Everything in Pham language are element, many element. Element can have nested element, and can be contained in other containing element. Every code in Pham language are many element, too.

\* Use 'space' character as separator between element .

\* Use well-closed parentheses to group element in strict hierarchy structure .

\* Any pham language element , can exactly mimic an ordered set , which have element as the nested element of this original pham language element .

For example : theElement

In this code , there 1 pham language element 'theElement', which have no nested element .

The element 'theElement' can exactly mimic an abstract ordered set , and this abstract ordered set exist but have no element .

For example :

Pham language code : (myElement)

This code is an element in Pham language. This element '(myElement)' contain 1 nested element : 'myElement' .

For example :

Pham language code : ((house of myName) (black door) (transparent window))

This element '((house of myName) (black door) (transparent window))' have 3 nested element : '(house of myName)' , '(black door)' , '(transparent window)' in 1-level of its own hierarchy structure.

This pham language element '((house of myName) (black door) (transparent window))' can mimic an abstract ordered strict hierarchy set : the ordered set here is 1 abstract house , which have first element as '(house of myName)' as name specification of the abstract house , and have second element as '(black door)' as door specification of the abstract house , and have third element as '(transparent window)' as window specification of the abstract house . Thus , when user want to describe a such house , then user can use any proper pham language element (, which infact are equivalent to ordered set , ) to mimic the user house. Then the user send these pham language element (code) to AI system . How AI system can guess pham language text (code) intention ? The AI system of course have predefined English vocabulary , which of course coincide with user English vocabulary . The AI system of course have predefined standard school ordered set theory knowledge , which of course coincide with user school ordered set theory knowledge . The AI system then read the hierarchy structure of pham language element (code) to see the abstract relationship between element, then the AI system use the coincidence between AI knowledges and user knowledge , thus the AI system can understand and guest the user intention about the user house in this example .

For example , there are many way to translate user natural language text into pham language text :

Suppose that user have the following user natural language text . And user want to translate their user natural language text into pham language text , and then send pham language text to AI system for better mutual understanding .

Example : user natural language text : a paragraph of 3 natural language sentence .

'''

I go to the street near the house of my friend .

I want to see my friend house .

I want to build a house , which look like my friend house .

''' .

And here is possible example translation :

(I go to street near the house of my friend)  
 (I want to see my friend house)  
 (I want to build house , which look like my friend house)  
 And here is possible example of translation with more detail  
 specification for AI system :  
 (I go to (street near the house of my friend))  
 (I want see (my friend house))  
 (I want build (house , which look like my friend house))  
 And here is possible example of translation with more detail  
 specification for AI system :  
 (I go to (street near (the house of my friend)))  
 (I want see ((my friend) house))  
 (I want build (house (which look like my friend house)))  
 And here is possible example of translation with more detail  
 specification for AI system :  
 (I go to (street near ((the house) of (my friend))))  
 (I want (see ((my friend) house)))  
 (I want (build (house (which (look like) ((my friend) house)))))  
 And here is possible example of translation with more detail  
 specification for AI system :  
 ((I go) to (street near ((the house) of (my friend))))  
 ((I want) (see ((my friend) house)))  
 ((I want) (build (house (which (look like) ((my friend) house)))))  
 All these example translation are pham language legal syntax , and all  
 they better help AI system to understand. The more detail grouping , the  
 easier for AI system to understand , and the less consumed energy of  
 calculating of AI system . All popular AI system today easy understand  
 the above natural text and pham language text. But the above pham  
 language text lead to drastically less consumed energy of calculation of  
 AI system than the natural text . User of course can not feel it ,  
 because pham language save consumed energy of calculation at server side  
 of AI system . More user precise topic and intention and contain can not  
 or extreme trouble be described in natural language , but can be  
 described in pham language in the rather familiar natural language way .  
 For example :  
 Code : (repeat (something) (n time))  
 This is 1 Pham language element '(repeat (something) (n time))' , which  
 have 3 nested element : 'repeat' , '(something)' , '(n time)' .

### 2.2.2/ More explanation to write Pham language element (Pham language code) :

\* Write Pham language command , pham language code. All they mean to  
 'write' Pham language element in well-closed parenthese hierarchy  
 structure .  
 For example, the Pham language code : (full name) is (family name)  
 (name)  
 There are 4 outermost parenthese element : '(full name)' , 'is' ,  
 '(family name)' , '(name)' .  
 The element '(full name)' have 2 nested element : 'full' , 'name' .  
 The element '(family name)' have 2 nested element : 'family' , 'name' .  
 For example, code : (create 2 record) (record1 ((record name) = 'aa' ,  
 (record value) = 11) , record2 (no))  
 Same example, code : (make 2 record) (((record name) = 'aa' , (record  
 value) = 11) , record2 (undefined))

These 2 example command cause AI system to generate the same interpretation. But they are absolute different command with different element.

For example ,

Code 1 : element1

Code 2 : (element1)

Code 3 : ((element1))

These 3 code line are not equivalent. The first code specify that there is 1 element 'element1' . The second code specify that there is 1element '(element1)' , which contain element 'element1' as its nested element in 1-level of its own hierarchy structure . The third code specify that there is 1 element '((element1))' , which contain element '(element1)' , which contain element 'element1' .

For example , the code : (element1 element2 element3)

In this example, the element '(element1 element2 element3)' have only 3 1-level nested element : 'element1' , 'element2' , 'element3' . The element '(element1 element2 element3)' have no any info about possible element '(element1 element2)' , '(element2 element3)' .

For example :

Case 1 : a pham language codepage code : element1 element2 element3

Case 2 : a pham language codepage code : (element1 element2 element3)

These above 2 case are not equivalent. In the first case , the pham language codepage have 3 nested element : 'element1' , 'element2' , 'element3' . In the second case , the pham language codepage have 1 nested element : '(element1 element2 element3)' . And the element '(element1 element2 element3)' have its own 3 nested element 'element1' , 'element2' , 'element3' .

\*\*\* The abstract separating space is the separator between element in pham language code. In writing convention , the abstract separating space is the first 1 "space" character , which stay straight after the pham language element or stay straight before the pham language element. All other consecutive space sequence , which stay straight before this abstract separating space or stay straight after the abstract separating space , are usual "space" character sequence element .

For example : '(A B)' : there are 4 space between 'A' and 'B' . Thus the first space straight after 'A' is the abstract separating space ; the forth space stay straight before 'B' and is also the pham language abstract separating space ; the second space and the third space combine into 1 sequence of consecutive space , which is normal pham language space element . Thus there are 3 1-level nested element : element 'A' , (1 element of 2 consecutive space) ' ' , and element 'B' in this example .

Pham language specific abstract separating space is not nested element of pham language element .

For example, the pham language code : a = b

It specify that there are 3 pham language element : 'a' , '=' , 'b' .

For other example , the pham language code : a= b

It specify that there are 2 pham language element in this place order : 'a=' , 'b' .

These 2 code are absolute different with different element, but they will cause AI system to generate same interpretation. They are all legal syntax in pham language. Note about the space character. The first example one cause less energy consumption of calculating in AI system, while the second example one can cause more energy consumption of

calculating in AI system. Recommended style for writing in pham language : the first example one.

For example , the pham language code : `a=b`

In this example, there is 1 pham language element '`a=b`' . If user want to specify the equality relation (or assignment , ..) , then user can write the above code. In most user context , the AI system will understand that the use intention is about the equality relation . But it is bad writing style for specifying the equality relation .

For example , pham language code : `(text : someText)`

In this example, note about many space inside this element. The element '`(text : someText)`' have 4 nested element : '`text`' , ':' , ' ' , '`someText`' . After ':' , there are 3 space character , thus the first space character and the last space character are the pham language specific abstract separating space , and the consequence of 1 consecutive space ' ' is a usual pham language element .

For example , code : `(text : someText)`

In this example , note about many space inside this element. The element '`(text : someText)`' have 4 nested element : '`text`' , ':' , ' ' , '`someText`' . After ':' , there are 4 space character , thus the first space character and the last space character are the pham language specific abstract separating space , and the consequence of 2 consecutive space ' ' is a usual pham language element .

For example : code : `(someElement otherElement )`

In this example , note about many space inside . After the element '`otherElement`' , there are 3 space . Thus the first space is pham language abstract separating space , and the consequence of 2 consecutive space is an usual pham language element . The element '`(someElement otherElement )`' have 3 nested element : '`someElement`' , '`otherElement`' , and element ' ' .

For example : code : `(someElement otherElement )`

In this example , note about many space inside . After the element '`otherElement`' , there is 1 space . Thus the first space is pham language abstract separating space , and the there is no any element after that . The element '`(someElement otherElement )`' have 2 nested element : '`someElement`' , '`otherElement`' .

In this case, Pham language reckon that the element '`(someElement otherElement )`' have the writing-element-description-code : `"(someElement otherElement)"` . Thus , the element '`(someElement otherElement )`' and the element '`(someElement otherElement)`' have absolute analogous same writing-element-description-code . These 2 such element are called absolute identical equal in pham language .

\*\*\* Convention : Normalize pham language code :

Pham language will ignore all such special abstract separating space , which stay before parenthese ')' and without any element between (this special abstract separating space) and this parenthese ')' . For example : '`(PP QQ )`' will be normalized as '`(PP QQ)`' .

Pham language will ignore all such special abstract separating space , which stay after parenthese '(' and without any element between this parenthese '(' and (this special abstract separating space) . For example '`( A B )`' will be normalized as '`(A B)`' ; '`(A B )`' will be normalized as '`(A B)`' ; '`( A B )`' will be normalized as '`(A B)`' .

Pham language will eliminate all consecutive space sequence , which stay after the parenthese '(' and without any element and any other parenthese between this parenthese '(' and (this consecutive space sequence) . For

example : `'( A B)'` will be normalized as `'(A B)'` . `'( )'` will be normalized as `'()'` .

Pham language will eliminate all consecutive space sequence , which stay before the parenthese `)'` and without any element and other parenthese between (this consecutive space sequence) and this parenthese `)'` . For example : `'(A B )'` will be normalized as `'(A B)'` ; `'( AB )'` will be normalized as `'(AB)'` .

Pham language will eliminate all consecutive space sequence in normalization process into only 1 pham language abstract separating space , if these consecutive space sequence stay after the 1 parenthese `)'` and before other 1 parenthese `'('` , and between this parenthese `)'` and this parenthese `'('` there are no any other element and no other parenthese but there are only space character. For example : `'(( ) (A B C))'` will be normalized as `'(( ) (A B C))'` .

Pham language will automatically correct and add 1 pham language abstract separating space to the place between the 2 consecutive parenthese `)'` and parenthese `'('` if there are no any space and element stay between this parenthese `)'` and this parenthese `'('` . For example : `'.. )( .. '` will be normalized as `'.. ) ( .. '` . But it is good style for human user and nonhuman user to always write right syntax `'.. ) ( .. '` , instead of wrong syntax `'.. )( .. '` . In pham language , parenthese `'('` and parenthese `)'` are reserved code character . It do not allow ambiguity to write `)('` as 1 string .

Pham language will automatically correct and add 1 pham language abstract separating space to the place between consecutive 1 single element and parenthese `'('` if there are no any space and element stay between this single element and this parenthese `'('` . For example : `'.. A(B C)'` will be normalized as `'.. A (B C)'` . It is good style for human user and nonhuman user to always write `'.. A (B C)'` instead of `'.. A(B C)'` . In pham language , parenthese `'('` and parenthese `)'` are reserved code character . It do not allow ambiguity to write `'A(B'` as 1 string .

Pham language will automatically correct and add 1 pham language abstract separating space to the place between consecutive 1 parenthese `)'` and 1 single element if there are no any space and element stay between this parenthese `)'` this single element . For example : `'(B C)D ..'` will be normalized as `'(B C) D ..'` . It is good style for human user and nonhuman user to always write `'(B C) D ..'` instead of `'(B C)D ..'` . In pham language , parenthese `'('` and parenthese `)'` are reserved code character . It do not allow ambiguity to write `'C)D'` as 1 string .

Pham language do not eliminate consecutive space sequence in normalization process , if this consecutive space sequence stay straightly between 2 other pham language single element , and it do not belong to the above listed case of eliminating consecutive space sequence .

For example : `'(A B)'` will be unchanged in normalization process . `'(A .)'` will be unchanged in normalization . `'(A (B C) " " (E F))'` will be unchanged in normalization . `'(P Q R . (X Y))'` will be unchanged in normalization . `'((P Q) , A B)'` will be unchanged in normalization .

Pham language will eliminate all consecutive space sequence in normalization process into only 1 pham language abstract separating space

, if these consecutive space sequence stay after the 1 parenthese ')' and before other 1 single element (single element without parenthese) , and between this parenthese ')' and this single element (single element without parenthese) there are no any other element and no other parenthese but there are only space character.

For example : '(C D) E (F G) H)' will be normalized as '((C D) E (F G) H)'

Pham language will eliminate all consecutive space sequence in normalization process into only 1 pham language abstract separating space , if these consecutive space sequence stay after the 1 single element (single element without parenthese) and before the parenthese '(' , and between this single element (single element without parenthese) and this parenthese '(' there are no any other element and no other parenthese but there are only space character.

For example : '(A B (C D))' will be normalized as '(A B (C D))' .

For example : code : (someElement otherElement ) ( someElement otherElement ) (someElement otherElement)

In this example, note about many space inside these element. There are 3 element in the 1-level of hierarchy structure : '(someElement otherElement )' , '( someElement otherElement )' , '(someElement otherElement)' . These 3 element are absolute identical equality in pham language , because they all have same writing-element-description-code . For example : code : ( A B 23 ( C 35 ) D) .

It will be (A B 23 (C 35) D)

For example : code : (element1 (element2 element3)) (element1 (element2 element3 ))

In this example, note about space inside these element. In this case , the element '(element1 (element2 element3))' and the element : '(element1 (element2 element3 ))' have absolute analogous same writing-element-description-code : "(element1 (element2 element3))" . They are absolute identical equal in pham language .

For example , the pham language code : a , b

It specify that are 3 element : 'a' , ',' , 'b' .

For example , the pham language code : a, b

It specify that there 2 element : 'a,' , 'b' .

These 2 code are absolute different with different element, but they often will cause AI system to generate same interpretation. They are all legal syntax in pham language. Note about the space character. The first example one is more strictly describe the intention than the second example one. Recommended style for writing pham language : the first example one.

For example , the pham language code : (a , b)

Note about the space character in this code. This code strictly mean that : there is strictly 1 compound element '(a , b)' , which have 3 nested element : 'a' , ',' , 'b' . Do not misunderstand that this code specify only 2 element 'a' and 'b' . In pham language, everything is element, thus the character ',' is element too. And this code itself '(a , b)' is a 1 whole compound element in Pham language code page.

\* A pham language codepage is a hypertext file. In pham language, a 1 pham language codepage is itself a 1 compound element, which can contain its nested element too.

Sometime, can use term 'pham language message' or 'pham language command' to represent short pham language code page.



Thus 1 pham language message is a 1 short compound pham language element, which can content its nested element.

And 1 pham language command is a 1 short compound pham language element, which can contain its nested element.

A pham language code page can contain element, which specify link to many other pham language code page.

For example : a pham language code page can contain the following element at some line :

(link to 'somePhamLanguageCodePageFile.txt')

After the AI system interpret this element, the AI system know that the current pham language code page have reference to file

'somePhamLanguageCodePageFile.txt' to find other element.

For example, you can also describe a link to other pham language page by write the other element :

(refer to 'somePhamLanguageCodePageFile.txt')

This element cause the AI system to link the specified page to the current page too. It is the flexibility of pham language in combining with AI system, it allow user to avoid to remember so much nonsense detail (for example , user can worry, which keyword to choose, 'refer' or 'link' or 'include' or 'import' or 'ref' or 'cite' .. to link to the specified file ? ) , so that user can concentrate their memory to core essence and relationship between essence of user world.

### **2.3/ There are only 2 primary core concept in pham language : the concept 'element', and the concept 'abstract relationship between element :**

\* The first primary core concept of pham language is 'element' : In pham language, because 'element' is the first of first primary concept, thus there is impossible definition of 'element', but there is only illustration of 'element'. In pham language, 'element' is anything, which can be perceived by the AI system.

\* The second primary core concept of pham language is the 'abstract relationship between element'. Because abstract relationship between element is primary concept, there is impossible definition of abstract relationship between element, but there is only illustration of that abstract relationship between element confirm that there exist abstract relationship between element.

\* Hierarchy structure : In Pham language, anything is element. Element can have its nested element. Element can also be contained in other containing element. A pham language code page is a 1 compound pham language element. A pham language code page have 1 its own mathematical strict hierarchy structure of the pham language code page, where the pham language code page stay in 0-level of its own strict hierarchy structure, and all other element of pham language code page stay in many different higher level of strict hierarchy structure of pham language code page. In Pham language, any element also have its own strict hierarchy structure, where the element itself stay in 0-level of its own strict hierarchy structure. In Pham language, strict hierarchy structure are specified by the well-placed arranging of many well-closed parenthese.

\* The own hierarchy structure of pham language element define the existing of all abstract relationship of all its nested element and its element too.

For example, the typical pham language code : (element1) (element2) (element3) (element4) (element5) (element6 , element7) .

This example pham language command specify place order of these element, as you see this place order : (element1) (element2) (element3) (element4) (element5) (element6 , element7)

The place order of element in pham language command explicitly define all abstract relationship between element : there are abstract relationship between element in the same level of the strict hierarchy structure of pham language command.

For example, this typical pham language command : (element1) (element2) (element3) (element4) (element5) (element6 , element7) .

This example pham language command define the following abstract relationship between element :

- + there is adjacent abstract relationship between (element1) -> (element2)

- + there is adjacent abstract relationship between (element2) -> (element3)

- + there is adjacent abstract relationship between (element3) -> (element4)

- + there is adjacent abstract relationship between (element4) -> (element5)

- + there is adjacent abstract relationship between (element5) -> (element6 , element7)

- + beside, there is nonadjacent abstract relationship between (element1) -> (element3), and there are other analogues nonadjacent abstract relationship between element in same level of hierarchy structure of pham language command.

## **2.4/ Specification principle of pham language :**

### **2.4.1/ EXPLICIT SYNTAX PRINCIPLE OF PHAM LANGUAGE COMMAND :**

All pham language command must strictly follow the above 1 AND ONLY 1 syntax. This principle guarantee strict mathematical form of all pham language command, so that other program can easy access nested element for more complex task, and so that all AI system can consistently extremely fast without any doubt to parse pham language element , and so that you will immediately know error when you see strange syntax thus you will save time without ambiguity of doubt about the right syntax.

### **2.4.2/ DEFINITE NUMBER ELEMENT PRINCIPLE :**

Any pham language code page, pham language command , pham language message must contain definite number of element.

But pham language command, which contain definite number of element, can describe set of infinite number element.

### **2.4.3/ STANDARD ENGLISH VOCABULARY PRIORITY PRINCIPLE :**

Priority to use standard english vocabulary (standard english word).

Principle of using standard english vocabulary : when use standard english vocabulary, then it guarantee that all AI system can exploit their the predefined worldwide accepted meaning of standard english word, so that AI system can more exactly generate intended interpretation of pham language command.

Typical pham language command do not specify any concrete meaning, but it only specify 2 following thing:

- + there are such element, which are placed in explicit mathematical hierarchy structure;

+ and there exist such abstract relationship between these element in same level of explicit hierarchy structure.

The AI system will read english word in pham language command element, see the explicit hierarchy structure of pham language command to know element abstract relationship in strict hierarchy structure. Then the AI system use their rich arsenal of predefined worldwide accepted meaning of standard english vocabulary, in combination with the strict hierarchy structure and element abstract relationship in pham language command, to generate final interpretation of pham language command, (for example, final interpretation can be text answer, or image, or video, or explicit programming code like Python code/Rust code/C code, ..).

Thus by prefer to use standard english vocabulary in pham language command, it guarantee that the AI system can quickly understand pham language command, thank to the rich arsenal of predefined standard english vocabulary of AI system. You can give definition to weird english keyword, then later use weird english keyword, but it is not good practice, because the AI system knowledge about your weird english keyword is poor ! . It mean that all word (without predefined user definition), which have intended meaning to the AI system and other people to understand and infer and interpret, should be standard English word. To specify unique private name of element, possible use private word outside standard English vocabulary.

For example : the word 'myPrivateHouse' , which is not in standard English vocabulary, is used to identify name of something private thing.

```
(my house (name = 'myPrivateHouse' , type is townhouse , color = green) )
(generate image ((2 myPrivateHouse in row) in sunshine))
```

There should be no plural english word in pham language command, for example it should be "2 car" instead of "2 cars". Because pham language command use the mathematical grammar and strict hierarchy structure to combine word and specify number of any element.

#### **2.4.4/ Levelize the relation between human cognitive load with short text record :**

The more explicitness of pham language command, the less required energy consumption of computing in AI system.

The shorter key word and word, -> the shorter text record, but -> the extremely more required human user cognitive load.

So should levelize, should try to avoid such too short weird keyword and word in pham language command, because it can save text record and text typing, but it require extreme big additional human cognitive load when you or orther people reread your pham language command next time.

Beside, short and short weird keyword simply can save text typing, but mostly do not save machine code realization time.

#### **2.5/ Writing Convention :**

Convention about how to write pham language code in conventional simple text file :

Write pham language code in text file .

Lets call each pham language code text file by name 'pham language codepage' or shortly 'page' .

Page contain sequence of ordered line. Each line contain limited number of text (code).

Each pham language codepage is 1 whole combined element . As combined element , which have nested element , thus the pham language codepage element must stay in 1 outermost pair of parentheses : pham language codepage is 1 pham language element with all thing , which stay inside the outermost parentheses in the pham language codepage file . Important note : A pham language codepage is always 1 whole pham language element , which begin with parentheses '(' and end with parentheses ')' .

But by writing convention , possible ignore writing the outermost pair of parentheses of the pham language codepage , but reckon that these outermost parentheses implicit exist .

When send pham language message (short pham language codepage) to another AI system , the combined pham language message element is also stay in well-closed parentheses .

For example :

The pham language codepage :

Line 1 : element1 (element2 element3)

Line 2 : (element4 element5) element6 .

In this example , the pham language codepage element is : (element1 (element2 element3) (element4 element5) element6) .

#### **2.5.1/ Write many short element in 1 line :**

A typical line can contain many short element .

For example :

Line code : element1 element2 element3

Must carefully think when group element in well-closed parentheses to specify relationship . Thus , if no intended relationships, then no need to group in parentheses .

For example :

Code 1 : (my text)

Code 2 : ((my) (text))

In these above examples , these 2 example explore same intention . But the first example is good writing style . Because there is no need to group 1 single word , which is itself 1 simple element without nested element , in 1 pair of well-closed parentheses.

Recommend good writing style : '(my text)' instead of '((my) (text))'

It is good practice to group element , which have intended relationship , instead of let them in separated place .

For example ,

Code 1 : my car black your car white

Code 2 : (my car black) (your car white)

Code 3 : ((my car) black) ((your car) white)

In these above example, recommend good writing style as in example2 and even better in example3 , because example2 and example3 better specify the element abstract relationship to AI system .

#### **2.5.2/ Write 1 long element in many consecutive line :**

Use well-closed parentheses to write 1 long element in many consecutive line.

Example :

In line1 : (element1 (element2) element3

In line2 : element4 (element5) element6)

These 2 line mean 1 long combined element :

(element1 (element2) element3 element4 (element5) element6)

In this example, note that there must be at least 1 'space' after 'element3' in line1 or before 'element2' in line2 , so that when straight connect these 2 consecutive line in a row, there must be 'space' character to separate element3 and element4 .

## **2.6/ Write Function . Assign variable .**

User must note to make variable assignment and function assignment to 1 arbitrary whole pham language element , so that the AI system can know and recall this element when the user call function and call variable again .

General user use their memory of their familiar function description in one of the following sector : in general knowledge sector or in school subject sector or in various popular programming language . To write function description .

General user use their memory of their familiar variable assignment syntax in one of the following sector : in general knowledge sector or in school subject sector or in various popular programming language . To write variable assignment .

But they all must note well-close parenthese to make variable assignment to 1 whole pham language element , and function assignment to 1 whole pham language element .

They are of course not explicit syntax .

But the AI system already did load such worldwide standard knowledge in these sector , which are coincided with the user knowledge . Thus the AI system must easily quickly guess and understand the user intention , and interpret exactly the user intended function description and variable assignment .

For example : variable assignment :

Code : (myHouse = ((Name : Omega) (type : good House) (have garden) stay along the road))

(What is myHouse ? )

In this example , the user use familiar variable assignment syntax . Thus when ask the system about the myHouse , then the AI system must answer the house , which have description '((Name : Omega) (type : good House) (have garden) stay along the road)' .

For example : function declaration , and function calling :

Code : (myTask = (repeat (print Hello) (3 times)))  
(I will do myTask)

In this example , myTask is assigned to the element '(repeat (print Hello) (3 times))' , which is a function. Thus then when call myTask , then the AI system will do this function .

For example : function declaration , and function calling :

Code : (myFunction[x , y] = (x + y))  
(print myFunction[1 , 2])

In this example , the result is that the AI system print number 3 . Look at this , the pham language did not specify that the syntax (myFunction[x , y] = (x + y)) is function definition . But this syntax is appeared somewhere in school , or in some programming language , or in some society sector . Thus the AI system , which already load these popular 'somewhere' standard knowledge , see these user syntax and easily guess user intention of function declaration and rightly interpret the user function declaration syntax .

For example : function declaration , and function calling :

```
Code : (define function myFunction{x , y} = (x*y))
        (print myFunction{2 , 3})
```

In this example , the AI system will print number 6 . Once again , pham language did not specify the syntax (define function myFunction{x , y} = (x\*y)) to define function . It is the user syntax , which is appeared somewhere in school subject or in some programming language or in some other standard subject . The AI system , which did load all such 'somewhere' knowledge , see this syntax and quickly rightly guess and interpret the user function declaration . The

For example :

```
Code : (define function (myFunction{x , y} = (x*y)))
        (print myFunction{2 , 3})
```

In this example , it is analogous to the previous example , the AI system will print number 6 . The more special feature of this example is that in the function definition , the user already more detailedly group element , to more precisely specify element relationship , thus it get the AI system easier and quicker to interpret user intended function declaration .

## **2.7/ Conditional execution . The If-clause . And execution looping .**

User must note to specify conditional execution for 1 whole pham language element , and specify looping for 1 whole pham language element , so that the AI system can exactly and quickly allocate the specified element for conditional execution or for looping .

General user use their memory of their familiar conditional execution

(the If-clause) syntax in one of the following sector : in general

knowledge sector or in school subject sector or in various popular

programming language . To write conditional execution (the IF-clause).

General user use their memory of their familiar execution looping syntax

in one of the following sector : in general knowledge sector or in school

subject sector or in various popular programming language . To write

execution looping .

But they all must note well-close parentheses to specify 1 whole pham language element , so that the AI system can exactly and quickly allocate the specified element for conditional execution or for looping .

They are of course not explicit syntax .

But the AI system already did load such worldwide standard knowledge in

these sector , which are coincided with the user knowledge . Thus the AI

system must easily quickly guess and understand the user intention , and

interpret exactly the user intended conditional execution and execution looping .

For example : IF-clause :

```
Code : (printTask[x] = (print (OK x))) (x = 3)
        (If (x > 0) then (printTask[x]))
```

In this example , the AI system will print : OK 3 . The pham language did not specify that the syntax (If (x > 0) then (printTask[x])) is

conditional execution . It is user syntax . But this syntax (If (some condition) then (do something)) is appeared somewhere in natural English

or in school subject or in some programming language . The AI system ,

which did load all such 'somewhere' knowledge , see this syntax , and

must easily remember their loaded knowledge to guess and interpret user

intended conditional execution . In this example , the task of the user

is not to remember pham language specific syntax , but to well group

entity to form the pham language element '(some condition)' and '(do something)' in 1 pham language universal syntax .

For example : the looping :

Code : (For (i from 1 to 3) do (print i))

In this example , the AI system will consequently print number 1 , then number 2 , then number 3 . Once again , pham language do not specify that the above syntax is for execution looping . It is user invented syntax . But this syntax appear somewhere in natural English language or in school subject or in popular programming language or in other society sector . The AI system , which already loaded all such 'somewhere' knowledge , see this syntax and easily guess and interpret the user intended execution looping . The extremely important task of user is that they must rightly group entity in right pham language element to specify right relationship between entity . Pham language is strong universal tool to do this . Of course , if human user wrongly group entity and wrongly specify relationship between entity , then no any AI system can help .

For example : the looping

```
Code : (begin pham language)
        ((this pham language message) is to generate (Python
code))
        (myList = (a list object))
        (for (temporaryFile in (directory 'MyDirectory)) do (add
(name of temporaryFile) to myList))
(end pham language)
```

In this example , the element '(for (temporaryFile in (directory 'MyDirectory)) do (add (name of temporaryFile) to myList))' do not make any loop . It simply specify the AI system to create the Python code , which iterate all file in the directory 'MyDirectory' and save the file name to the list myList . The most important feature here is that the element '(for (temporaryFile in (directory 'MyDirectory)) do (add (name of temporaryFile) to myList))' is infact in familiar syntax pattern (for (some index in somewhere) do (something)) , which appeared somewhere in natural English language or in programming language or in school subject .., . And the AI system , which already loaded all such somewhere knowledge , see this syntax and easily guess and rightly interpret user intended of execution looping .

## **2.8/ Typical use case of pham language as highest level human-AI language :**

One of the powerful use case of pham language is that to use pham language to describe user intention to AI system , so that AI system generate specific lower level but more explicit other language code for the user intention , for example Rust code , Python Code , other lower level human-AI system specific language . Then user can quickly check or debug or modify the AI generated lower level but more explicit language code .

There are lot of AI specific system for various different task , from making video to drawing , .. . And but all they step by step require you to learn these commands and other commands and ... . In short word , these AI specific system require you to pay time to learn their command . Time by time , it mean that hiddenly decline natural language , to switch to more specific low level language in each specific sector .

Pham language do not require human user to learn specific command , but pham language encourage human user to simply use their natural language vocabulary in pham language to communicate with AI system . In this meaning , pham language contribute to save and promote humanity culture legacy and humanity natural language legacy in the era of AI everywhere . Of course , professional user work in narrow specific sector , which have narrow low level specific language , should pay time to learn their sector specific low level language . Even in this case , pham language can very helpful to help them more quickly get their required complex AI-generated message of their sector specific low level language , because biological user can not remember all detailed syntax of their work sector specific low level language .

## **2.9/ Use pham language in chat session with AI : add each short pham language message .**

Because pham language itself contain natural language . Thus the useful method to use pham language in hybrid style : natural language inside pham language message .

(Before using pham language , copy the content of this text file and paste to chat windows to AI system to learn. Or send text file to AI system. In some case , I may require to divide this text file content into many smaller text message , then paste all these text message to the chat windows to AI system.)

For example :

Code :

(Hello this chat is about using pham language in hybrid mode : natural language - pham language message) .

In this example , there is 1 pham language message element . But the contain of this pham language element is simply natural language .

For example :

Code :

(I will write simple matter in natural language)

(I will write precise matter in pham language)

(this look like precise matter (myText = (first 1-level nested element of this message)))

(print (number of word in myText))

In this example , there is 1 long pham language message element , which contain 4 1-level nested element . The first 1-level nested element contain simple natural language . But the third element and the ford element already contain more structured pham language style element , which describe the user intention to count all word in the first 1-level nested element . The AI system will print number 8 .

The content of this text file is official specification of pham language, and is also official parsing instruction to AI system to parse pham language code , and is also training material for AI system to train to implement pham language . Good AI system take about few seconds to fluently learn and implement pham language .

### **2.9.1/ Use pham language with other non-english natural language vocabulary :**

It is easy to use other non-english natural language vocabulary in pham language.

Because pham language use mathematical grammar for its explicit hierarchy structure to combine word and to describe element abstract relationship.



Thus, it is the same effect to use non-english natural language vocabulary in pham language, but only if your AI system companion have good predefined knowledge of these non-english natural language vocabulary.

### 2.9.2/ More example for training :

Example 1 :

```
(begin pham language)
((iterate all file in directory) (directory name = 'my directory')
(change all file name to upper case))
(end pham language)
```

{Copy and give the above command to AI system, for example Microsoft CoPilot, Chat GPT, Google Gemini, Facebook AI. Then ask them to generate Python code or Rust code or C code, .. . You will receive explicit Python code doing exactly the intension}

Explain : there are 3 nested element : (iterate all file in directory), (directory name='my directory'), (change all file name to upper case). The AI system understand these element.

The AI system see that there is abstract relationship : (iterate all file in directory)->(directory name='directory') .

The AI system see that there is abstract relationship : (directory name='directory')->(change all file name to upper case).

So the system decides the following interpretation : "change all file name in the specified directory into upper case.

(if you give this pham language command to the AI system, for example, ChatGPT, Google Gemini, Microsoft Copilot, Facebook AI, for example, to ask generate python code, then they will generate explicit python code to do exactly the intended task).

Example 2 :

```
(begin pham language)
( look (all file in directory) ((directory name) = 'my directory')
(change all (file name) to (upper case)))
(end pham language)
```

{Copy and give the above command to AI system, for example Microsoft CoPilot, Chat GPT, Google Gemini, Facebook AI. Then ask them to generate Python code or Rust code or C code, .. . You will receive explicit python code doing exactly the intension}

Explain : this example 2 do the absolute same task as example 1 . But example 2 describe more strictly abstract relationship.

Thus example 2 help the AI system to quicker and easier to understand, and remarkably save energy consumption of AI system computing at server side.

You maybe can not feel that AI system quicker to understand the example 2. But when you send file of many command such as example 2 , then you can feel that AI system quicker understand example 2. The example 2 require less energy consumption of computing at AI system server side. They, the example 1 and example 2, are all legal right syntax of pham language command.

In example 2 , there are 4 outermost element : "look", "(all file in directory)", "((directory name) = 'my directory')", "(change all (file name) to (upper case))"

Note that the complex element : ((directory name) = 'my directory') .  
 It has 3 nested element : (directory name), "=", 'my directory'.  
 The AI system will read these 3 nested element, with their specified abstract relationship, to decide that must have "directory name" = 'my directory'.  
 In this case, the abstract relationship : (directory name) -> "=", "=" -> 'my directory', are more closely to describe the reality of that (directory name) = 'my directory' .

Now review the example 1 : the element (directory name = 'my directory') have 4 nested element : "directory", "name", "=", 'my directory' .  
 With abstract relationship : "directory" -> "name", "name" -> "=", "=" -> 'my directory' .  
 you see that these abstract relationship less strictly describe the real reality of that (directory name)='my directory'.  
 Thus the example 1 is longer to AI system to understand, and require more energy consumption of computing in AI system.

Example 3 :

```
(begin pham language)
(tire (global name = myTire) (description : ((automobile tire ((hermetic
inner tube tire) contain compressed gas) , (outer tire)) , (dimension (1m
, 1m , 0.3m))) , (color is black)))
((column of tire) (global name = myTireColumn) (3 myTire in vertical
row))
((drying area) (global name = myDryingArea)
  (description : (the drying area of apartment))
  (area type : area of typical (high floor) apartment)
  (area feature : (beautiful and clean area))
  (area feature : ((big opened window) with view to sky))
  (area feature : (dimension : (2m 4m 2.5m)))
((object in area) : (myTireColumn stay in (floor and (very close to the
window)))))
)
(end pham language)
{This code ask AI system to generate image, which must precisely describe
the scene.
```

Copy this code, paste this code to chat windows with AI system, then it will give the image of precise scene : an array of automobile tire stay in specific drying area}

Example 4 :

```
(begin pham language)
(set variable ((variable name) is 'a', (variable value) is 123))
(set variable ((variable name) is 'b', (variable value) = 456))
(repeat (print (a + b + index)) (5 time) (index go from 1 to 5))
(end pham language)
{ Paste the above pham language code to AI system, such as Google Gemini,
Microsoft CoPilot, ChatGPT, and ask them to generate explicit programming
language codes, for example Python code, Rust code, C++ code, then you
will receive explicit programming code doing the intension. }
```

The above instruction is intension to do simple program :

```
+ set variable a=123
+ set variable b=456
```

+ repeat print (a + b + index) for 5 times, where index go from 1 to 5.

Beside , it is possible to simply write : (a = 123) to set a variable . User can choose their familiar way , instead of must find the specific way of setting variable .

This is simple program. But use meaning here is that : you do not know, for example, Python code syntax, or you know Python syntax not well, but you can describe enough precise intention in pham language, so that you receive explicit Python code to realize your intension. In natural language, it is impossible to strictly describe precise intension/instruction to AI system.

Of course, if you know very well programming languages and can very quickly remember colossal number of all detailed syntax and command of programming language, then you should not use AI, but you can do all your instructions by yourselves writing programs. But reality shows that it require big human cognitive load, and almost people are not comfortable with that. Pham language levelized, it require substantial less your cognitive load for syntax memory, while allow you to maximal strictly describe precise matter, and allow minimal consume energy of AI calculating at server side.

One of the most powerful use case of pham language : you use pham language to possible precise instruction to AI system, so that AI system will generate programming code (for example Rust code, Python code, C++ code, .. ) realizing your intension. Then you correct the received code .

## **2.10/ Pham language as universal AI-AI language :**

Pham language is also perfectly universal AI-AI language to share and exchange knowledge between different AI system of different specific architecture and specific model .

Pham language specific instruction and specification for AI-nonbiological user is described in next part , which is dedicated for nonbiological user (AI user) to write pham language message to exchange knowledge with AI system . It allow more precisely describe matter in pham language. Of course it do not restrict human user to write and use these pham language specific command .

\*\*\* \*\*

**3/ Strict Pham language specific rule and instruction are dedicatedly to use pham language as universal AI-AI language . This extending part specify strict pham language specific rule , so that ((AI system) and (nonbiology user) and (biology user)) can use to write more specific pham language message to describe more precise content to exchange knowledge between AI system - AI system .**

(Description of this part is not included in the Pham language human-AI language version)

(Description of this part is in the Pham language human-AI-AI language version)

## **4/ Contact :**

Contact : the Pham language project :

phamlanguage@gmail.com  
github.com/phamlanguage  
youtube.com/@phamlanguage  
youtube.com/phamlanguage