This document includes research data, It is sufficient to read about 50 percent of the document.

I recommend that you view it as a pdf and dark mode(for font and image): https://github.com/Fhres126/nl/blob/main/nl.pdf

if that link dont work: https://glki.netlify.app/nl.pdf

youtube: https://youtube.com/@normallanguage?si=4E6OEKgSuZa9OJAj

our chat room: https://discord.gg/VhHBjXbpkz

if you dont like this document then give me feedback or see after 1month. that issue can be fixed.

all feedback is welcome!

norlang{

this artificial language is called norlang(Normal Language) in english.

advertisement{

over 65% of words is compound word and antonym and arranged in an orderly way.

so you can remember words easily.

grammar is simplest and similar to math.

norlang dont have exception.

norlang will can become the international language.

one sentence has only one meaning.

eg 'A and B of C' can be interpreted as '(A and B) of C' and 'A and (B of C)' in english.

but in norlang, 'af C be A B' mean only '(A and B) of C'.

'be A af C B' mean only 'A and (B of C)'.

Norlang is logical, so it prove Sapir-Whorf hypothesis.

'i know you' in norlang is 'nbbd'.(This is pronounced differently from English. it is easy to pronounce)

english is 80bit(in ascii code).

norlang is 16 bit(one letter is 4bit).

norlang is five times more efficient than English.

so norlang reduces environmental pollution.

norlang is phonetic script.

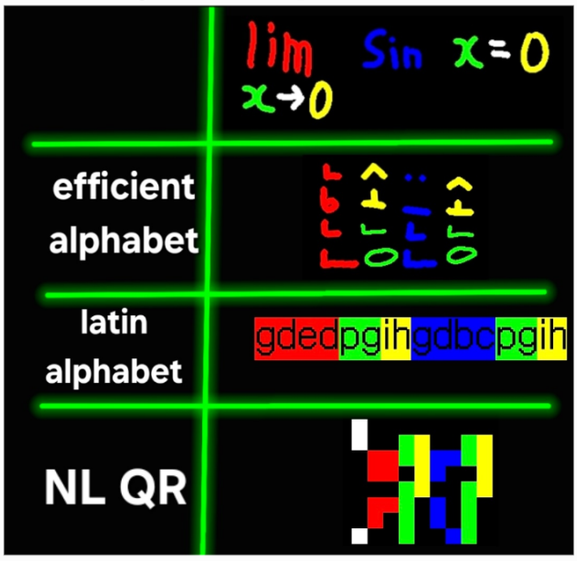
norlang helps easy communication.

norlang brings peace and helps technology grow.

norlang versions of barcodes and QR codes is called NL QR.

Since norlang is a binary language, it can easily be converted into NL QR and and norlang version morse code.

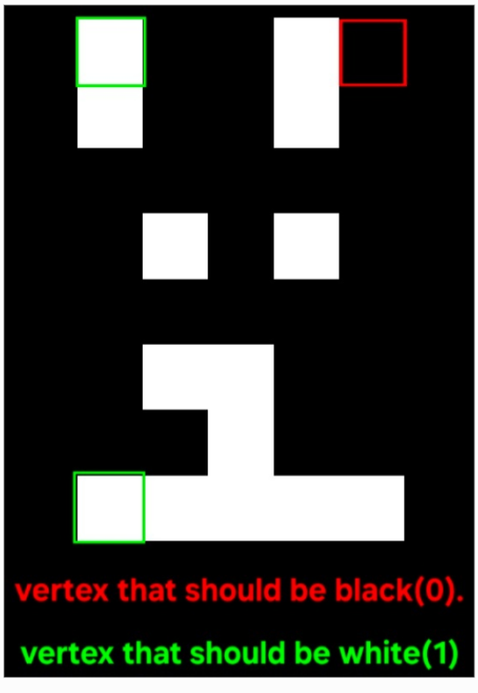
norlang can express all meanings using only binary.



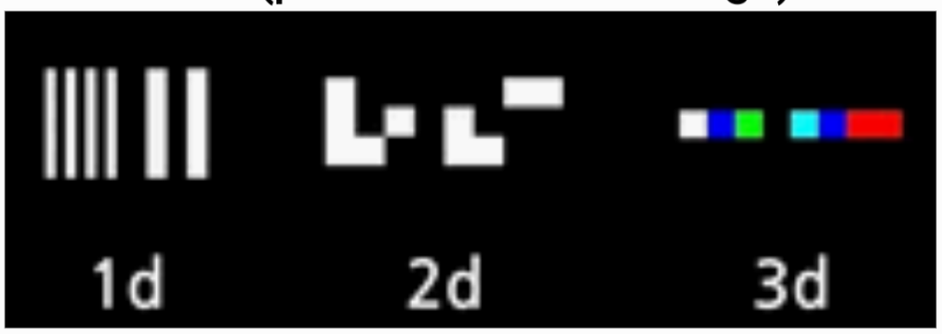
NL QR1 is over 1.5 times more efficient than a barcode.(both image mean same)

NL QR2 uses over 5 times less space than a QR code.(the both image mean 'i am people')

NL QR means the same from any angle like QR code.



NL QR can be expressed in one, two, or three and more dimensions(pdf user can see image).



you can use norlang fluently in just one month

It is the simplest, most efficient, easiest and most logical future international language.

norlang can be used in programming.(eg

for(var i=0;i<4;i++){

console.log(i)

}

goic ih gh ja fh gh jh mg gh lh

kb gh ad

bc)

it is called ECL.

}advertisement

alphabet{

* alphabet is only 0 and 1 then it can be expressed as 16 latin letter.

a:0000. b:1000. c:0100. d:1100.

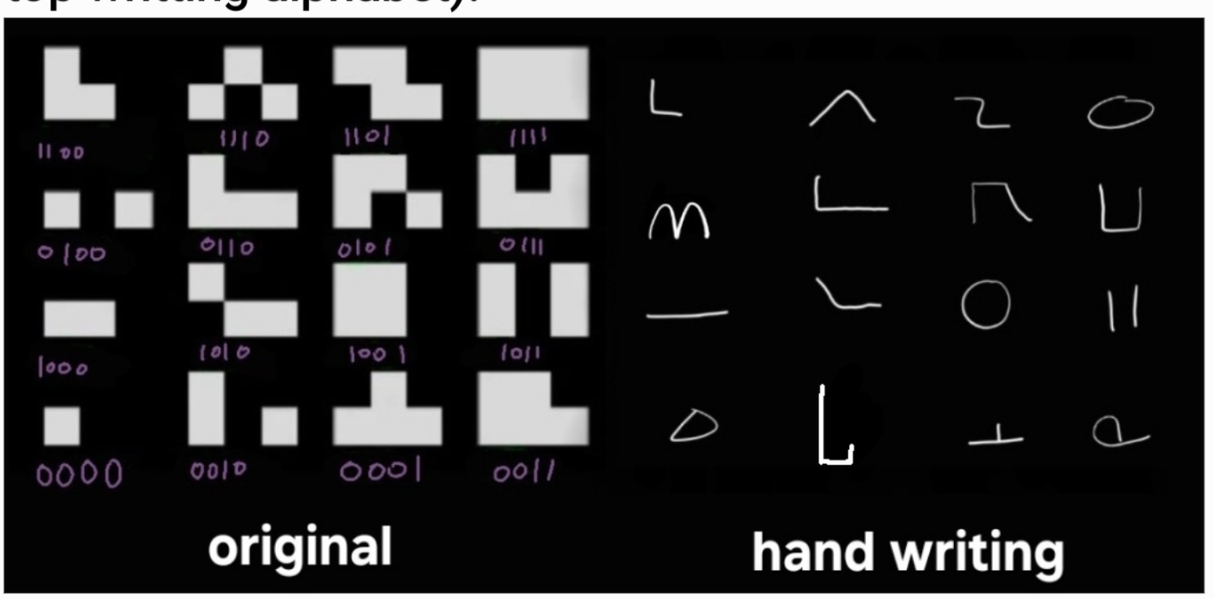
e:0010. f:1010. g:0110. h:1110.

i:0001. j:1001. k:0101. l:1101.

m:0011. n:1011. o:0111. p:1111.

norlang uses only lowercase letters.

more efficient alphabet is here(only pdf user can see. bottom to top writting alphabet):



one word consist of 8n bit.

8n-th bit determines length of word.

8n-th bit 0 plays the same role as whitespace of english so it deosnt need whitespace.

* 8n-th bit 1 make the word consist of more bit.(eg sentence '{0000101(1)0110000(0)}{0111000(0)}' consist of two words.

between '(' and ')' is 8n-th bit.

between '{' and '}' is one word.)

it means that word whose 2n-th letter is between 'a'(0000) and 'h'(1110) consist of smaller quantity bits.

the example converted to latin letter is 'anga oa'.

but i will use whitespace to make you easily understand.

pronunciation{

* Two letters make one syllable.

* (2n-1)th letter is pronounced as initial consonant.

* 2n-th letter is pronounced as vowel also final consonant.

eg 'fh ee anba' is pronounced as 'nwo da maemboo'.

(2n-1)th letter 'f' is pronounced as 'n'.

2n-th letter 'h' is pronounced as 'wo'.

(2n-1)th letter 'e' is pronounced as 'd'.

'e' is 'da' in pronunciation chart.

if 'e' is in 2n-1th letter it pronounced as 'd'.

if 'e' is in 2n-th letter it pronounced as 'a'.

2n-th letter 'e' is pronounced as 'a'.

then 'anba' is pronounced as 'maemboo' then it can be pronounced

as 'maeb'.

pronunciation chart{

sound consists of decibels (dB) and hertz (Hz).

it is arranged with lower Hz sounds coming first.(eg m(100hz) < s(14000hz))

so it is not biased toward any specific culture.

a:'moo'. b:'bo'. c:'geo'. d:'leu'.

e:'da'. f:'nae'. g:'ppi'. h:'kkwo'.

i:'ttoom'. j:'pom'. k:'keom'. l:'teum'.

m:'jam'. n:'jjaem'. o:'chim'. p:'swae'.

IPA version{

a:'mu'. b:'bo'. c:'gʌ'. d:'ɾɯ'.

e:'da'. f:'nɛ'. g:'p̪i'. h:'k̪wʌ'.

i:'t̪um'. j:'pom'. k:'kʌm'. l:'tɯm'.

m:'tɕam'. n:'t̪ɕɛm'. o:'t͡ʃim'. p:'swɛ'.

}IPA version

}pronunciation chart

}pronunciation

}alphabet

EML(Efficient Memory Language){

lecture{

basic{

* grammar of norlang is same as math.

* norlang can express every meaning.

* transitive verb is function.

* object is parameter.

* norlang lacks concept of verb.

* word order of norlang similar to VOS.

* norlang sentence 'A B' mean 'B is A'.

* norlang sentence consist of two objects.

eg 'djca aoad' mean 'ore is solid'.

'djca' mean 'solid'.

'aoad' mean 'ore'.

* some word must have parameter like function.

'jd bd oa' mean 'i think you', '(thing that think you) is me'.

'jd' mean 'thing that think'.

'bd' mean 'you'

* 'jd' is function that must have parameter so 'jd bd' is one object.

'jd bd' mean 'thing that think you'.

'oa' mean 'i'.

* if subject is omitted, subject is 'i' so 'i think you' can be expressed as sentence 'jd bd' and 'jd bd oa'.

this grammar can express every meaning.

eg 'The happy boy gives her a small book in the morning and smiles.' is expressed as 'ab gb if ab dmkg mf bida ajmd kjkc ac ljhd ajea ab dmdg boob'.

you need to know how to interpret a dictionary to learn in detail.

<How to interpret a word dictionary>{

'fojb r!=dead thing.'{

word is expressed as it in word dictionary.

* far left is norlang word.

* far right is meaning of this word.

* between norlang word and '!' is word type.

* 'r' is an abbreviation for 'reality'.

only thing that can exist in reality is reality type object.

}

'kjgg r!=alive thing.//anto die'{

'//' is for an additional explanation.

'anto' is an abbreviation for 'antonym'.

* 'anto A' mean 'this word is antonym of A'.

* The antonym form is the inversion of the original word's bits.

0 becomes 1. 1 becomes 0.

eg antonym of
'fojb'(10100111,10011000) is
'kjgg'(01011001,01100110).

}

'jd r!o=thing that think A.'{

between '!' and '=' is parameter that this function must have.

'o' is an abbreviation for 'object'.

any object can be in object type parameter.

'A' in right hand side is first input value of parameter of this function.

eg 'jd oa' mean 'thing that think me'.

}

'pf si!=continue speaking.'{

'si' is abbreviation for 'sign'.

'si' type words don't interface with all words.

eg sentence 'jd pf bd oa' means 'jd bd oa' because the word doesn't interface with 'si' type words.

You can use 'pf' word to continue what you were saying before.

}

}<How to interpret a word dictionary>

any word can be in object type parameter.

'jd oa' is one object.

'jd jd oa' is one object and mean 'thing that think thing that think me'.

'jd ad oa'='i think', '(thing that think something) is me'{

ad si!=symbol that omit one parameter.

FYR: (jd ad) oa.

parameter of 'jd' is omitted by 'ad'.

}

'higina coea oa'='my name is higina', 'higina is name of me'{

coea v!o=name of A.

FYR: higina (coea oa).

}

}basic

intermediate{

<How to interpret a word dictionary>{

'fc r!o,o=thing that ask B about A.'{

'!A,B' mean this function need 'A' and 'B' as parameter.

A is first input value of this function of norlang word, the second one as B, and then C, D, E, F, and so on.

parameter type that this function must have is object and object.

eg 'fc aoad oa' mean 'thing that ask me(oa) about ore(aoad)'.

}

'lb p!o=this A'{

'p' is abbreviation for 'parameter'.

parameter value determines type of this function.

eg 'lb fojb' is reality type cuz fojb is reality type object.

}

'ge 2p!o-o-n=thing that become B using A.'{

'2p' mean '2rd parameter'.

type of 2rd input value determines type of this word.

eg type of 'ge oa kjgg' is reality type object cuz 'kjgg' is reality type object.

The symbol '-' means that if the parameter is omitted by symbol that omit parameter then the next parameter is omitted too.

you can omit all parameter of 'ge' with only one 'ad'.

}

'oa r!=i.//com express,this,symbol that omit parameter..'{

'com' is an abbreviation for 'compound word'.

'com A,B' mean 'this word is compound word of A and B'.

'oa' is compound word of 'kb' and 'ha' and 'ad'.

'kb ha ad' mean 'thing that express this'.

total length of all that words is 24bit.

but i want to express word 'i' as 8bit word cuz word 'i' is used a lot.

The floor of 8 divided by 3 is 2.

2bit is from 'kb'(01,011000).

2bit is from 'ha'(11,100000).

4bit is from 'ad'(0000,1100).

result is 'oa'(01,11,0000).

}

'anga r!=people.//pron de mensch'{

'pron' is an abbreviation for 'pronunciation'.

'pron de mensch' means that the pronunciation of this word is derived from the German word 'mensch'.

'de' mean german in ISO code alpha2.

}

'gb p!s-n=reason why A2 is A1'{

's' is an abbreviation for 'sentence'.

's' is 'o-o'.

so 'A1' is first 'o'.

'A2' is second 'o'.

}

'be p!c o=A1 and A2 and....'{

'c o' mean 'this word have object type parameter until it met symbol that omit parameter'.

eg 'be oa oa oa ad' mean 'me and me and me'.

}

ee t!s=time when A2 is A1.{

't' is abbreviation for 'time'.

}

}<How to interpret a word dictionary>

'dmba cf gjif glid bd ajmd oa' = 'I see her giving you food', '(thing that see that thing that give food to you is she) is me'{

dmba r!o=thing that see A.//com sense,photon,.

cf r!s=that A2 is A1.

gjif o-o=thing that give A to B.//com reason,have,

glid r!=food.

ajmd r!=she.//com the,women,.

FYR: (dmba (cf (gjif glid bd) ajmd)) oa.

}

'jc dmkd klbg oa cf boob ad oa' = 'if i listen music, i am happy', '(what happens if thing that listen music is me) is (that thing that feel pleasure in something is me)'{

jc r!s,n=what happens if A2 is A1.

dmkd r!=thing that listen A.//com sense,sound,.

klbg r!=music.//com sound,pleasure,.

cf r!s=that A2 is A2.

boob r!o,n=thing that feel pleasure in A.

FYR: (jc (dmkd klbg) oa) (cf (boob ad) oa).

}

'mb ee dmkd klbg oa boob ad oa' = 'I am happy when I listen to music'{

mb 2p!t-o=thing that be B when A.

ee t!s=time when A2 is A1.

FYR: (mb (ee (dmkd klbg) oa) (boob ad)) oa.

}

'ef cf dmkd klbg oa boob ad oa' = 'i am happy cuz i listen music'{

ef p2!o-o=B by A.//com also,result,fact,.

cf r!s=that A2 is A1.

FYR: (ef (cf (dmkd klbg) oa) (boob ad)) oa.

}

'lc dmkd klbg oa boob ad oa' = 'i listen music but im happy'{

lc p!s-o=but.//com inverse,that a2 is a1,.

FYR: (lc (dmkd klbg) oa (boob ad)) oa.

}

'mb dh ld flda ph de bd' = 'I saw you 3 minutes ago'{

dh n or t!o-o=A minus B.

ld t!=now.//com this,time,

flda t!o=A minute.

ph n!=3.

de r!o=thing that sense you.

FYR: (mb (dh ld (flda ph)) (de bd)).

}

<how to ask>{

'fc anga bd'='what is people? i ask you', '(thing that ask you about people) is me'.{

fc r!o,o=thing that ask B about A.

}

'fc ne bd'='where are you?,i ask you', '(thing that ask you about location of you) is me'{

ne v!o=location of A.

if second parameter of 'fc' is omitted, the second parameter mean 'you'.

FYR: fc (ne bd).

}

'fc gb dnfd oa bd'='why do you hate me?', '(thing that ask you about reason why thing that hate me is you) is me'{

gb p!s,n=reason why A2 is A1.

dnfd r!o,n=thing that hate A.//com dread,normal,

if second parmeter of 'fc' is omitted, it is 'you' or 'i'(alone think).

'gb dnfd oa bd' mean 'reason why you hate me'.

FCR: fc (gb (dnfd oa) bd).

}

'fc jc dnea noda bd'='what if you give up being worker?', '(thing that ask you about what happens if thing that give up being worker is you) is me'{

jc p!s,n=what happens if A2 is A1.

dena r!o=thing that gave up being A.

noda r!=worker.

FYR: fc (jc (dnea noba) bd).

}

'fc ja dmba oa bd'='are you looking at me?', '(thing that ask you about whether thing that see me is you) is me', {

ja r!s=whether A2 is A1.

dmba r!o=thing that see A.//com sense,photon,.

FYR: fc (ja (dmba oa) bd).

}

'fc ec gjkb lb anga boba'='how do we save this person?', '(thing that ask you about how we become thing that save this people) is me'{

ec r!s-n=how A2 become A1.//com what,need to,.

gjkb r!o,n=thing that save A.//com reason,alive,.

lb p!o=this A.

boba r!=we.//com and,i,you,o,.

FYR: fc (ec (gjkb (lb anga)) boba).

}

'fc ee de id boba'='when we sense each other?', '(thing that ask you about time when thing that sense each other is we) is me'{

ee t!s=time when A2 is A1.//com what,thing that be B when A,

id r!=each other.//com what,interface,self,.

FYR: fc (ee (de id) boba)

}

}<how to ask>

'jf anga ha' = 'this seems people'{

jf r!s=thing that knows it is true that A2 is A1.//com know,sen,true,whether,.

FYR: (jf anga ha).

}

time{

'*jb gjfg anga* oa', '*jb gb fojb anga* oa'='i killed people', '*thing that was thing that kill people* is me'{

jb p!o=thing that was A.

gjfg ow,n=thing that kill A.//com reason,die,.

gb p!s,n=reason why A2 is A1.

fojb r!=die.

'gb fojb anga' means 'reason why people is die'.

FYR: (jb (gjfg anga)) oa.

}

'*gc gjfg eida* oa'='i will kill that', 'thing that will be thing that kill it is me'{

gc p!o=thing that will be A.//anto was A.

eida o!=it.

(gc (gjfg eida)) oa.

}

'eb jb fojb oa'='i have never died', 'thing other than thing that was dead thing is me'{

eb v!o=thing other than A.

jb p!o=thing that was A.

FYR: (eb (jb fojb)) oa.

}

}time

how to command{

'fa kjgg bd' = 'hey you, be alive!'{

fa r!o,o=thing that orders B to become A.

kjgg r!=alive.//anto die,.

}

}how to command

<symbol that omit parameter>{

'fc ad bd'='i ask you', 'thing that ask you about something is me'{

ad si!=symbol that omit one parameter.

bc si!=symbol that omit two parameter.

cc si!=symbol that omit three parameter.

dc si!=symbol that omit four parameter.

omitted parameter become mean 'something'.

}

'dnfd be bd dnfd bd bc oa'='i hate you and thing that hate you'{

dnfd r!o,n=thing that hate A.

be p!c o=A1 and A2 and....

in this siruation

it should omit two parameter so 'bc' is used instead of 'ad'.

the first omitted parameter is 2nd parameter of 'dnfd'.

second omitted parameter is parameter of 'be'.

FYR: (dnfd (be bd (dnfd bd)) bc) oa.

}

}<symbol that omit parameter>

'ge af oa fjdb dmba bd '='i see you using my eye', 'thing that become thing that see you using eye of me is me'{

ge 2p!o-o-n=thing that become B using A.

af 2p!o-o=B of A.//com also,element-of,.

fjdb r!=eye.//com organism,see,.

dmba r!o=thing that see A.//com sense,photon,

FYR: (ge (af oa fjdb) (dmba bd)).

}

'klkb fleg'='i speak fleg'{

klkb r!o=thing that speak A.//com sound,express.

fleg=norlang,//com normal,language,.

}

"='im from'{

}

'mmfd bd oa', 'me fd bd oa'='i love you', 'thing that love you is me'{

mmfd r!o,n=thing that love A by B percent.//com hope,normal,

me r!s,n=thing that hope that A2 is A1 by B percent.

fd r!=normal,

'me fd bd' mean 'thing that hope that you are normal'.

}

}intermediate

upper intermediate{

<How to interpret a word dictionary>{

'ih n!=0'{

'n' is abbreviation for 'number'.

}

}<How to interpret a word dictionary>

<how to use 'ea'>{

'mmfd bd oa', 'bd ea mmfd oa'='i love you'{

'bd ea mmfd oa' mean 'what i love is you'.

'ea' is similar to 'what' of english.

but word 'ea' target n-th paramter.

list of parameter:

first parameter: 'o' of 'mmfd'.(what i love)

second parameter: 'n' of 'mmfd'.(how much i love)

if word 'ea' target second parameter, it mean 'how much i love you'(ea kh mmfd bd oa).

FYR: bd (ea mmfd oa).

}

'fc ad bd oa', 'bd ea nh fc ad oa'='i ask you'{

ea v!c n,s=what.//ea remove targeted parameter by A.

'bd ea fc ad oa' mean 'you are who i ask to'.

To know the meaning, you need to list the parameters of all function in parameter sentence of the 'ea'.

function in parameter sentence of 'ea' is 'fc'.

list of parameter:

first parameter is 'what it asks about'.

second parameter is 'what is asks'.

'ea nh' mean '2nd parameater of this sentence'.

so 'ea' target second parameter of 'fc'.

FYR: bd (ea nh fc ad oa).

}

'nb bd oa', 'bd ea nb pa oa'='i know you'{

first sentence mean 'i know you'.

second sentence mean 'what i know is you'.

The 'pa' is in parameter of 'nb'.

parameter of 'nb' is 'what it know'.

}

'ea mmfd bd'='ea nh mmfd bd'='ea mmfd pa bd'{

}

'ae anga mmfd oa'='people that i love', 'lovely people'{

ae p!o-<c n,s>=A that C2 C1.//C2 is subject,,C1 is verb,,com also,what,

'ae anga mmfd oa' mean 'ab anga ea mmfd oa'.

ab p!o-o=A also B.

}

}<how to use 'ea'>

}upper intermediate

advanced{

<How to interpret a word dictionary>{

'bg n!int=integer'{

'int' is an abbreviation for 'integer'.

'int' is for expressing only integer.

so it dont make parameter.

'bg' express integer number.

eg 'bg he' is 35.

first bit determines whether the number is positive or negative.

if first bit is zero, number is negative.

integer 'he' is '111000010'.

(2nd bit)*1=1.

(3rd bit)*2=2.

(4th bit)*4=0.

(5th bit)*8=0.

(6th bit)*16=0.

(7th bit)*32=32.

1+2+32=35.

so it is 35.

}

'if p!o=thing that have A.//anto element'{

he p!o=element of A.

'if' is '0 0 0 1 1 0 1 0'.

'he' is '1 1 1 0 0 0 1 0'.

antonym of 'he' seems 'id'(00011100).

antonyms have an order.

An element must exist for the owner to exist.

so word bit of 'element' must smaller than 'have'.

then antonym of 'he' seems 'ih' but 'ag'~'ph' is for number.

word like 'element' is called 'former word'.

'gidd o-o-n=thing that need be A to be B,//com need,also,self,.'{

gljb o-<o,n>=thing that need A to be B.

ab r!o-o=A also B.

dd c=self.

The floor of 16 divided by 3 is 5.

But the length of element word in the compound word must be a power of 2.

}

'anae p!o=her A.//com of,she,.'{

ajmd r!=she.//com the A,women,.

compound word of 'of' and 'she' seems 'anab'.

but word 'anab' already exsit.

element of compound word is split.

'of' word have already splited cuz it is already compound word.

af 2p!o-o=B of A.//com also,element-of,.

take initial of 'ajea mlmb'.

it is 'a m'.

and Set the 8th bit to 0

so result is 'ae'.

'an'+'ae'='anae'.

}

}<How to interpret a word dictionary>

<number>{

'ag'~'ph' is number word.

'bg da', 'lh'='1'{

'bg da' is '10000110,11000000'.

'lh' is '11011110'.

and see between '(' and ')' in next.

'bg da' is '10000110,(110)00000'.

'lh' is '(110)11110'.

'ih'~'ph' is 'eni'(efficient number integer).

}

'cb nh fojb'='two dead thing'{

nh n!=2.

}

'cbfojb'='quantity of dead thing'{

}

'dg fa lh'='1.25'{

dg n!int-n=real number.

'fa lh'='10100000.11011110'.

lh n!=1.

'fa' is in decimal area.

}

'cb v!n,o=A B.'{

example of 'number' + 'word' in english is 'one apple'.

eg 'cb ih anga' mean 'zero people'.

ih=0.

If different type word like 'r' is in place of an 'n' type parameter, then 'n' is automatically omitted.

if parameter 'n' of 'cb' is omitted, function 'cb' return quantity.

eg 'cb anga' mean 'quantity of people'.

you can omit number type parameter to express quantity of number using symbol that omit parameter.

eg 'cb ad ih' mean 'quantity of 0'.

}

'i love you by 100percent'='mmfd bd ph oa'{

ph n!=4.

this situation dont need minus.

second parameter of 'mmfd' is 'intensity of love'.

}

'I love you as much as I love him'='mmfd bd pg ajde oa'{

pg n!o=A converted into number.

ajde r!=he.//com the,men,.

}

'i love you more than she'= 'mmfd bd fh ajmd'{

fh n!o=more than A.//anto less,

ajmd r!=she.//com the,women,.

}

}<number>

<mathematical expression>{

'mmfd bd ph'='i love you with 100percent intensity'{

}

'mg lh nh ph'='1+2=3'{

mg n!o-o=A plus B.

lh n!=1.

nh n!=2.

ph n!=3.

}

}<mathematical expression>

}advanced

example sentence{

'fc ab ab bked mocg lllb of cmeh'='What is the fastest route to get to the sea'{

}

}example sentenece

for who wanna make new word{

}for who wanna make new word

//i review document up to this point using 20250612version norlang.

<logic and how to make new word>{

la p!c o=too and.

'la oa bd ad' can be 'oa' and 'bd' and ('oa' and 'bd').

if A={B, C}, 'he A' return 'la B C ad' so sentence 'B he A' is true.

'ma he A' is only 'be B C ad' cuz it mean 'all element of A'.

'A is B and C' mean 'A is B also C'.

but if A is la type, A is not it.

}<logic and how to make new word>

NL QR{

convert alphabet to binary.

NL QR1{

0 is black or white.

1 is opposite color of 0.

draw '10'.

input it to next of '10'.

draw '11' at last to finish.

read from '10' to '11'.

eg 'oa'->'01110000' -> '10,01110000' -> '10,01110000,11'.

if last is not finished by '01' then you can omit '11' of last.

eg '10,01110000'.

}NL QR1

NL QR2{

draw '10000011'

draw bit in right or left.

eg 'oa' ->

10000011
01110000

if you wanna draw bigger NL QR, draw 1010000000000011 or 101000000000000.....11

}NL QR2

}NL QR

quiz{

1.meang of sentence 'fc ea jd bd'.

2.translate 'hello my name is gociba' to norlang.

3.meaning of sentence 'coob ea jb bd'.

answer{

1.What are you thinking?

2.gociba coea oa.

3.I don't know what you're thinking.

}answer

}quiz

}lecture

word dictionary{

aa si!=finish-all,

ba si!=whitespace.

ca

da si!=enter.

ea v!c n,s=what.//ea remove targeted parameter by A,,target,certainty,pronoun,

fa r!o,o=thing that orders B to become A.

ga p!o=only A.

ha

ia p!o,n=thing that tends to become A.

ja s=whether A is true.//zero is true.cuz true is important.exist made false.one is exist.

ka p!c o=array.

la p!c o=too and.

ma p!o=all,

na r!=fact,//com too and,A include,real,true,o,.

oa r!=i.//com express,this,omit word,

pa o!//untranslatable word, it is used by ea.

ab r!o-o=A also B,////no cw but intend com be,and,.

bb v!o=retuen value of A.

cb p!n,o=A B.//eg two apple.

db v!int=word as int A from onself.

eb p!o=thing other than A.

fb !n,o=B that has id A.

gb p!s-n=reason of A.

hb p!o,n=thing that can be A with B percent probability.//bisi becoming.

ib v!o=freeze function.

jb p!o=thing that was A.//cant com b when a,past,o,.

kb p!o=thing that express A.//com make,code,.

lb p!o=this A.

mb 2p!t-o=thing that be B when A.

nb o=thing that know A.//com include,code,

ob 2p!o-o=thing that be B with A.

pb o!n=what function.//1st bit 1 mean int type

ac p!o-o=B that A include.//com also,thing that A include,.

bc si!=symbol that omit too parameter.

cc si!=symbol that omit three parameter.

dc si!=symbol that omit four parameter.

ec r!<o-o>-n=how B become A.//what,need to,

fc r!o,o=thing that ask B about A,//com command,reply,

gc p!o=thing that will be A.//anto A of past,.

hc//mean s ae,two,.

ic p!o-o=intersection of A and B,//com also,A include,A include,

jc p!s-n=what happens if A2 is A1.//anto reason.

kc int-int...:1p=thing that says A.//com express,str,.

p1!o-o=B which is called A.

lc p!s-o=but.//com inverse,that a2 is a1,.

mc//mean s what,two,.

nc r!o=thing that call A.//com command,sense,i,.

oc p!o=thing that becoming A.

pc 1p!o-o=B converted into A.

ad si!=symbol that omit one parameter.

bd r!=you.//anto i,

cd p!<o-o>-n=thing that become A using oneself also B.//
use,also,self,.

dd c=self.//no cw but intend com word-as-id,+2id.

ed p!o-o=between A and B.//com exceed,below,.

fd r!=normal.

gd int=A th function.

hd p!o=other A.//cant com also,not,wtdoc,.

id r!=each other.//com what,interface,self,.

jd r!o=thing that think A.

kd r!o=thing that make A.//com reason,A include,real,.

ld t!=now.//com this, time,.

c int=norlang ipa.

md p!o,n=thing that intend be A.

nd c int=foreign word.

od p!o=thing that become A.

si!int=version marker.

pd si!ni=backspace.

ae p!o-<c n,s>=A that C2 C1.//com also,what,,C1 is verb.

be p!c o=and{,//com or,most,

ce

de r!o=thing that sense A.

ee t!s=time when A2 is A1.//com what,thing that be B when A,

fe r!o,o=thing that forbids B from becoming A.//com command,not,

ge p!o-o,n=thing that become B using A.

he p!o=element of A.//include promise.

ie o-o=B according to A.

p!o=essence of A.

je t!s=past of A.

ke r!n=abnormal.//anto normal,

le o=around of A.

me r!s,n=thing that desire A is true with B percent intensity.

ne n or r!o=location of A.

oe !n,o=Ath B.//minus is from last.

pe p!o=average of A.

af 2p!o-o=B of A.//com also,element-of,.

bf p!<c o>-n=or{.

cf r!s=that A2 is A1.

df r!s,n=thing that dread A with B percent intensity,//anto desire,

ef 2p!o-<o,n>=B that is by A.//also,result,fact,.

ff p!n,o=thing that similar to A.

gf t!s=future of A.//anto past of A.

hf si!o=symbol that change parameter to word type.

if r!o=thing that have A.//anto element,.including promise

jf r!s=thing that knows it is true that A2 is A1.//com know,sen,true,whether,.

kf r!o,o=thing that reply to B by saying A..//anto ask,

lf o=minus of A.

mf o!=word that depends on context.

nf !o-o=nearest A to B.

of r!o=thing that go A.//com become,

pf si!=continue-say,

{

p3!int-int-para...=too function

}

ag

bg n!int:-1 in minus=integers,

cg v!int-int...=str.

dg

eg n!int=value.

n!int-n:int=real number.

fg

gg n!o=not A.

hg n!o-o=and A B.

ig n!o-o=xor A B.

n!=other-math.

jg n!o-o=or A B.

kg n!o=less than A.//A>

lg n!o=bellow than A.//A>=

mg n or t!o-o=A plus B.

ng n!o-o=A times B.

n!int=unknown,

og n!o-o=A square B.

pg n!o=A converted into number.//com converted,number,.

ah

bh n!o-o=A root B.

ch n!o-o=A divided by B.//anto sqare.

dh n!o-o=A minus B.//anto plus.

eh n!o=exceed than A.//anto bellow,,A<=

fh n!o=more than A.//anto less,,A<

gh n!=x.

hh n!=y.

ih n!=0.

jh n!=4.

kh n!=-1.//dnm 5

lh n!=1.

mh n!=-2.//dnm 6

nh n!=2,

oh n!=-3.//dnm 7

ph n!=3.

by word formation rule{

anto{

dmdg r!=men.//anto women.

fjdg o,n=thing that feel uncomfortable be A.//anto comfortable.

nmbg=opened.//anto closed.

njdg p!n=splited.//anto combination.

ojbg o,n=thing that feel pain about A.//anto pleasure,.

kjgg n=alive,//anto die,

epjg o,n=no.//anto yes,

glbg o,n=thing that think that A is ugly.//anto beauty.

jkdg o=thing whose A is friend.//anto enemy,

jlje o-o=thing that lose to A in competition of being B.,//anto win,.

omeg o=thing that emits A.//anto inhale,.

njef o=thing that include A,//anto thing that A include,.

dome o=thing that fight with A.//anto peace,

fneg=shift.//anto fixed,

jlbg o=outside of A.//anto inside,.

mocg o!=most.//anto minus most,.

mjhc r!o,o=thing that sell A to B.//anto buy,.

oohh r!=false.//anto true,.

//eanto

}anto

com{

mlde r!o=thing that looking for A.//com intend,sense,

glid r!=food.

cnad=situation.//com sen,o,.

ejcf r!=logic.//com how,true,situation,.

bjia r!=true.//com return value,whether,zero,zero,.

mlph r!o=thing that focus be A.//com intend,ph,.

cnba r!o=As.//com quantity,over,one,.

gidd o-o-n=thing that need to be A to be B.//com need,also,self,

gjfd o,n=thing that normalize A.//com reason,normal,.

gjfg o,n=thing that kill A.//com reason,die,.

kkcd r!=phonetic alphabet.//com express,sound,letter,

r!o=A taster.//com know, a include,.

r!=objective.//what,beilive,true,logic,.

ongg o=thing that go in A.//com go,inside,.

kkbb v!s=evidence of A.//express,sen,true,whether,.

mmgc r!o,n=thing that feel angry by A.//com hope,ruin,.

gmmd r!o,n=thing that help A,//com reason,what,intend,self,

r!n=thing that laughs.//com fun,express,,ori com use,fun,o,express,.

clkb r!o,o,n=thing that write A at B.//letter,express,.

dmbg r!o=thing that read A.//com sense,code,.

dmba r!o=thing that see A.//com sense,photon,

o,n=thing that gives birth to A,//com launch,baby,.

bidd r!o,n=thing that enjoy be A.//com pleasure,also,self,.

mlge r!o,n=thing that mind A.//com intend,use,.

ljhd t!=today.//com this,day,.

anbg p!o=information of A.//of,code,

fikg o=thing that weaker than A.//com newton,less,

fifh o=thing that stronger than A.//com newton,more,

bmkg o=thing that shorter than A.//com meter,less,

bmfh o=thing that longer than A.//com meter,more,

cmkg o=thing that narrower than A.//com meter2,less,

cmfh o=thing that wider than A.//com meter2,more,

dmkg o=thing that smaller than A.//com meter3,less,

dmfh o=thing that bigger than A.//com meter3,more,

bmed o-o=distance of between A and B,//meter,between,

ajed o=pointed A,//com also,pointed,

ajea o=the A.//com also,it,.

o=type.//com diffrence,one,

moke o,n=thing that angry at A.//com hope,reason,abnormal,

gjic o-o=thing that give A to B.//com reason,have,

gjnb o-o=thing that teach A to B.//com reason,know,

gjde o,o=thing that shows A to B.//com reason,sense,

dnfd o,n=thing that hate A.//com dread,normal,

mmfd o,n=thing that love A by B percent.//com hope,normal,

ajmd r!=she.//com the A,women,.

anae p!o=her A.//com of,she,.

ajde r!=he.//com the A,men,.

ajad p!o=his A.//com,he,.

inih r!=wave.//com hz,over,zero,.

klkb r!o=thing that speak A.//com sound,express,,ori use,sound,express,.

klbg=music.//com sound,pleasure,

=gay,//com hope,breed,gender,self,.

o=bye,//com command,around,not,i,

fleg=norlang.//com normal,language,.

p!c int=compounf word of A1 and A2.//com combination,word,

boba=we.//com and,i,you,o,

kjbb o,n=yes.//com express,know a is true,true,

fjdb r!=eye.//com organism,see,.

r!=gender.//com

o=water-per,//com intersection,water,

dmkd o,n=thing that listen A.//com sense,sound,.

olif o=thing that get A./com become,have,.

gjff r!o,n=thing that impact A,//com reason,shift,

=thing that delete A.//com reason,not,real,

ajab c o=A1 also A2 also...//com also,also,

=justice.//com objectice,equality,.

anaf c o=c of.//com of,of,

enef c o=c by.//com by,by,.

ikic c o=c intersecrion.//com intersection,intersection,.

elad r!=pointed.//com what,point to,o,

gjkb r!o-n=thing that heal A.//com reason,alive,

ilpe p!o=features of A.//com difference,average,

gjod r!o,o,n=thing that change A to B.//com reason,become,

njgd r!o=thing that calculate A,//com know,number,.

o,o,n=thing that moved from A to B.//com

mmnb r!o,n=think that wonder about A.//com hope,know,

mmgg r!o,n=thing that want to sex with A.//com hope,sex.

jmbb r!o=thing that denies A.//com know a is true,not,true,.

nmag s=thing that understand A.//com know,what,mean,.

o,n=thing that think that think A is cute.//com know a is true,baby,.

andb n-o=minimum of A.//com of, minus most,

anmg n-o=maximum of A.//com of,most,

bida n!=binary.//com too and,zero,one,o,.

enih r!=money.//com dollar,over,zero,

doif r!o,o=thing that buys A from B.//com promise,have,.

oidd r!o,n=thing that suffers from being A.//com pain,also,self,

bibd r!o,n=thing that thinks A is delicious.//enjoy,eat,.

oibd r!o,n=thing that think A is tasteless.//com suffers,eat,.

jidd o-o-n=thing that dont need to be A to be B.//com dont need to,also,self,.

klkd r!o=thing that sing A.//com speak,music,

oidc r!o,n=thing that feel that A is noisy.//com suffer,listen,.

olfh r!=increasing.//com become,more,.

olkg r!=decreasing.//com become,less,.

omge r!o=thing that enters into A.//com go,inside,.

omjd r!o=thing that leaves A.//com go,out,.

oknb r!o=thing that learn A.//com becoming,know,.

omfb r!=birth thing.//com brcome,organism,.

bked r!n=Ameter per second.//com meter,second,

//ecom

}com

order{

dica p!o=thing that start be A.

eica p!o,n=thing whose A is middle.

fica p!o=thing that finish be A.

many is first.

aiba r!=elementary particle.

biba r!=photon.

~bjba

angf o=front of A.

bngf o=front left of A.

cngf o=left of A.

dngf o=behind left of A.

engf o=befind of A.

fngf o=behind right of A.

gngf o=right of A.

hngf o=front right of A.

akba r!n=size of A.

bkba r!n=A meter.

ckba r!n=A metet^2.

dkba r!n=A meter^3.

blgd o=thing that eat A.//com inhale,food,,ord eat,.

clgd o=thing that chew A.

dlgd o=thing that swallows A.

elgd o,n=thing that digest A.

bjga r!=gas,

cjga r!=liquid,

djga r!=solid.

dlda t!=time.

elda t!o=A second.

flda t!o=A minute.

glda t!o=A hour.

hlda t!o=A day.

ilda t!o=A week.

jlda t!o=A month.

klda t!o=A year.

bmba r!o=south of A.

cmba r!o=west of A.

dmba r!o=north of A.

emba r!o=east of A.

color{

rrggbb.

aipa r!=black.

dipa r!=red.

mipa r!=green.

pipa r!=yellow.

alpa r!=blue.

plpa r!=white.

}color

shape{

jllb r!=shape.

kllb r!=point.

lllb r!=line.

mllb r!=triangle.

nllb r!=square.

ollb r!=pentagon.

pllb r!=hexagon.

aolb r!int=n gon.

}spape

//eorder

}order

lesi{

bode r!=body.//lesi en body.

clob r!=closed.//lesi en closed.

comb r!=combination.//lesi en combination

//ele

}lesi

pron{

clda r!=letter//pron kr geul,

aoga p!o=thing that mean A.//pron en mean.

figa r!o=newton,//pron en newton,,stength,

gjda r!o=thing that for A,//pron en for

dmga r!=country.//pron de country,

anga r!=people.//pron de mensch,

kkod r!=heart,//pron en hearts,.

kmga r!=thing that feel emotion.//pron jp kanjou.

fjda r!=organism.//pron en organism.

bjca r!=god,//pron ru bog

emdc p!o=A dollar.//pron en dollar.

gmge r!=father.//pron baby papa.

amae r!rmother.//pron baby mama.

//epron

}pron

bisi{

}bisi

bsf{

}bsf

mix{

//emi

}mix

}by word formation rule

Randomly generated words(no word formation rule){

mlmb r!=women.

mida r!=world.

cjba r!=transistor.

ojda r!=crops.

djba r!=real.

kjba=empire.

ljba r!=shop.

r!o=thing that famous in A.

r!=app.

komb o,n=something that feels comfortable be A.

boda o!=code.

r!o,n=kind of A.

lica p!o=inverse of A.

o=symbol that convert A into and type.

kklb r!=fixed.

coea r!o=name of A.

gljb r!o-<o,n>=thing that need A to be B.

jmob r!o,n=thing that feel beauty.

gmnb r!o=thing whose A is enemy.

lnda r!o=thing that point to A.

mjdd o=peace.

boob o,n=thing that feel pleasure about A.

gmgd o-o:=thing that win A in competition of being B.

klif =sound.

fojb =die.

colc o=thing that A include.

p!o,n=detail of A.

o,o=thing that record A at B.

elba r!int=digit.

flba r!o-o-n=thing that efficient than B for be A.

eida o!=it.

glea p!n=A-th word.

emga o,n=thing that rule A.

llga o=law of A.

//id, str...=UTF8.

dnea o,n=thing that was given up on being A.

=initial-consonant.

gnea=oil.

r!o=thing that subscribe A.

hifa=document.

kifa o=thing that put on A.

nifa o:standard,o,n=rank.

njfa=mountain.

//glfa o=plan.

p!o=property of A.

ajca r!n=political party,

r!=book.//com document,document,

n:'number of id'-id:'words'-str:'structure of parameter'=expressing-mean-of-word.

dkca r!o=daily of A.

ekca r!o=life of A.

hkga r!o=structure of A.

ikca r!o=principle of A.

kkca r!o,n=thing that think A is fun.

//r!=gun.

bllb r!o=thing that inhale A.

//r!o=thing that aim A.

//lolb o!o=variable A declared with var.//of js

//n=variable A declared with let.//of js

olca r!o=terrain of A.

gmca r!o=thing that select A.

r!=planet.

r!=sun.

onca r!o=standard of A.

//o:target,o:location=pros of A.

//o,o=cons of A.//anto pros,.

//eoca ni=natural-phenomenon,

gida p!o=limit of A.

aoad r!=ore.

//v!o=Ayears old.

//=novel.

//=legend.

goma r!o=thing that sex with A.

//r!pregnancy.//com include,baby,

//bkda o=value of A.

//ekda=capital.

//hkda r!int,int,int=rgb.

//cnda n-id...=IPA.

//hnda o=direction of A.

//inda o=angular-momentum.

//mnda ni=plant.

domb o-o=thing that promise A to B.

coda o=symbol of A.

//eoda =religion.

imga v!o=Ahz.

goda =civilzation.

joda o,n=thing that forgive A.

//poda o,o:standard=important.

cjea =shit,

gjea o,n=thing that repeat A.

jjea o,o,n=thing that sorry.

gmob o=inside of A.

pibd o,n=thing whose A is purpose.

jkea o:is,o,t=difficulty.

mkob o,n=thing that ignore A.

clea s=request,

flea o=thing that trade with A.

nlea ni=peninsula,

int=QR code.

hmba r!=home.

djnb o!=minus most.

kmba r!o,o=think that thankful that A is B.

lmba r!o=thing that revenge A.

nmba r!o=entropy.

onba r!=mistake.

joba r!=cpu.

noda r!=worker.

poba p!n or s=wages.

apga r!n=company.

epba r!o=thing that vote A.

p!o=thing that determine A.

r!n=mosfet.

goic s,o,o,c s=for(A2=A1;B;C;){D}.

int:<number of oord>-ni:<targeting parameter>-int<word>-int<structure>=mean of function.

kiga v!n=Akg.

glba n!=number.

eokb r!language.

//era

}Randomly generated words(no word formation rule)

word that will be created{

p!o=standard of A.

r!=mass.

anfh r!=titan.//people,over,.

r!=intellect.//com use,logic,.

r!o=thing whose A is role.

dnng r!=lazy.//com dread,work,.

o=example of A.

r!o=map of A.//com converted,image,

r!=baby.//com age,below,two,.

r!=video.//com image,image,.

ilgg p!c o=diffrence between A1 and A2.//anto similarity

hmjb p!c o=similarity between A1 and A2.

eacd p2!o-o=process of B becoming A.//com what,use self,

ajlb r!=here.//com also cast,this,.

r!o=thing that discriminates against A.//eg discriminates gender
it=Discriminate against it based on gender.

gmfe t!=morning.//com hour,between,five-int,twelve-int,

dlhb t!=yesterday.//com minus,today,day,one,

n!str...=decimal number.

!o=parameter to c.

!o-c o=if(A){B}.

midd t!=always.//com all,time,.

//efw

}word that will be created

id{

cast{

ag=word,
bg=r-w,
cg=n-w,
dg=t-w,

}cast

english{

b=space,

}english

function{

official is starting with 0.

da=differential.

fa n-n:from-n:until=integral.

ha=buffer gate.

ja=not.

la n:base-n=log.

na n:'from'-<n:'until'-n>=right-hand-lim.

pa n:'from'-<n:'until'-n>=left-hand-lim.

ed n-<n-w>=lim A->B.

eb s:'bottom'-n:'top'=sigma,

bc n=sin A.

bb n=cos A.

bg n=tan A.

i=factorization,

}

other-math{

ba=lowest,
ca=most,

}other-math

constant{

da=-most,
fa=most,
ha=-infinity,
ja=infinity,
la=natural,
na=imaginary,
pa=pi,
bb=speed-of-light,//photon

}

language{
af=ML,
bf=CL,
}

}id

}word dictionary

}EML(Efficient Memory Language)

official account{

youtube: https://youtube.com/@normallanguage?si=4E6OEKgSuZa9OJAj

reddit:

email(feedback etc): gwahagsinsa@gmail.com

}official account

rule{

This language is not complete and it can be changed so you should indicate the version when using it like 'nl60125'(6m1d25y) month->day->year.

root word{

in word that consist of more 16bit, 8n+7bit 1 mean that this word dont have parameter.

}root word

number{

'x1000010'
if x is 0,minus.
if x ia 1,plus.
before bit is small.
it is eficency.

10000000=+0.
11000000=+1.
01000000=-1.

if you should use integer 0, but 0 is expressed as 'aa' then trash bit is appear. then 'ba' is 0.

}number

reading way in NL QR{

in 1 dimension, '10' is at first for expressing reading direction.
in other dimension,it is
'10000011'.
or
'1011000000000011'
read like➡️⬇️
rows should not 8,

}reading way in NL QR

for developer{

The efficiency of inaccurate expressions and accurate expressions should be similar to same.

pronunciation chart order is low herz->high herz.

exact expression is important than efficiency.

one sentence must mean only one mean.

purpose of language is only communication and efficient.

0 cannot be used for counting order.

exception should none.

relative things must have standard.

sentence 'people i' means 'i have over 70% of features of people'.it is called 'ratio word'.
10.arrangement order of word is 'number->most common->'.

word formation rule{

word that need parameter{

1:meaning of word whose parameter is all or nothing is what other word can express.

2:relative word.

transitive verb.

test sentence{

input word your wonder to @.

Forgotten @ repeats itself.(eg history word dont need parameter. try input history of. is it owe?)

people in @ love you.

There should be no discrimination based on @(that is features of discrimination word. so 'element of all' word legal)

}test sentence

}word that need parameter

antonym{

bit of antonym is inverse of original word so 0 become 1, 1 become 0.

word that should come first{

priority-est is

low number(below->more).

order(eg seller exist to make buyer,, women is made before men

exist)

original:'df' mean 'meeb' so 'me' is first.

many quantity:photon is mannier than neutrino.

}word that should come first

if spectific word have lot of antonym, created by antonym mechnism is 'root(no slave,no king)->number(0% can=cant)->purpose(weapon,treatment tool,,intend,evolution)->diffrence set(dont know,know)'.

}antonym

compound word{

1.'know' is 'have'+'information-of'.

2.'girlfriend'='girl'+'love' cuz original is 'wae'+'girl'+'love'+'omit gramr'.it should keep word order.

3.'teethbrush'='teeth'+'brush'.

form of compound word must similar to original expression!(eg abangafhad(titan)->anfh(no fnaf!))

if new compound word overlap with other compound word,first element word of new compound word become small.(eg '00,10' become '0,1'. then if length of area of element is 1, use XOR).

length of bit of element of compound word must power of two.

if length of bit of element of compound word is not power of two, element of compound word such as symbol that omit grammar,

ea, of, and, or, also is omitted to be power of two.

small bit word is first in situation whose order doesn't matter.

if length of element compound word of compound word is less than whole bit divided by 8, use antonym mechanism.

In compound words, those made up of fewer visible elements become split first.

}compound word

phonetic loanword{

'russia'->'romsimna(russia in russian. slient consonant become 'n')'.
'higgs particle'->'(kims('h'->'k')).

}phonetic loanword

word formation rule that should be used first{

If a word can be expressed as an antonym and also as a compound word.

if word created by the antonym mechanism has the same meaning as the original word(eg mmfd=dlke) then use compound word mechanism.

If the original word too similar to compound wkrd, use antonym mechanism(eg i(com expres,this,o) -> you(com sense, this, o))

If the bits of a word expressed as a compound word differ by less than 1/4 from its antonym, use the antonym mechanism..

and If a compound word consists of two words, use the

compound word mechanism first using their antonym.(eg mmfd->dnfd)

If not, use the antonym mechanism first.

}word formation rule that should be used first

}word formation rule

}for developer

license{

//vc

nl version:70425

license can be changed.

this document and norlang have copyright.

You can use norlang legally for free.

you cant edit norlang.(eg chaing grammar, but you can make new word but that new word is not official)

but only large corporations and country must pay 500000dollar adjusted to 2025 value to use norlang per year except it of germany and laos and rwanda and honduras dont need to pay.(pay in january 1day. in 2025, it is 500000dollar)

The standards of large corporations are determined by the rules of their countries.

Violation of these terms may result in legal action under applicable

international copyright laws.

This license is subject to change. Future versions may alter terms and apply to future releases of the language.

Users must comply with the license version associated with the version of the language they received.

}license

}rule

other{

It is almost complete, but I am having particular difficulty completing the pronunciation chart.

if norlang have problem norlang is changed never mind other.

Creator Verification Code:'20@@0226@0@@@28. pn:@@05139@@@8 in 2024/11/18. JSM. SE. GIGADESADE. GOHW. SEMI'.

prohibits anything similar to this.

}other

history{

I created it during academic research because natural language was inconvenient. Logical grammar seemed inefficient, so it was made using binary.

prediction{

starting creating this language began in 1m 2023.

verb first is finalization in 1m 2023.

NL QR is finalization in 11m 21d? 2023.

VOS is finalization in 1m 2024.

antonym mechanism is finalization in 11m 2024.

starting perfectizing language in 1m 2025.

}prediction

}history

research data{

using NL ipa.

/•pa/ similar to /bbua/ than /pupua/.

is it lowest herz?{

voice recoder(app name is sound){

m(2400hz,13db)

g(3700hz,12)

kk(3700hz,10)

k(3700hz,18db), (2325hz, 18db)v

b

n(5500hz,17db)

l(7400hz,12db)

d(7400hz,10db)

tt(3700hz,22db)v

pp(3854hz,18db)v

t(5500hz,11db)v

p(3700hz,18db), (14000hz 18db)v

s(14000,16db) (3639hz 14db)v, (28000hz 10db)v

ss(3300hz,28db)v

j(21000hz,10db)v

jj(7400hz 22db)v

ch(21253hz,13db)v

oo,o,eo(3.7khz,3db),eu(7.4khz,5db),a,e,i,.

}voice recoder(app name is soound)

frequency generator{

m(100hz)

b(130hz)

g(150)

l(160hz)

d(185)

n(327hz)

pp(1000hz)

kk(1800hz)

tt(3000hz)

t(5400hz)

}frequency generator

my think{

sin{

m:100.b:140.d:160.n:180.'p:300.'t:500.'k:700.'c:7000.'j:12000.t:16000.

}sin

tri{

d:130hz.l:190hz.

}tri

squ{

c:11500.

}squ

saw{

d:30hz.l:40hz.

}saw

dlmbn(p?)Pp(j?)(h?)TtKk(Ch?)sJjSst

dlm(q?)bn(j?)(p?)(k?)'p(x?)'t'k(z?)c'j'ct

uyoraeiw'u'w'y'i'o'r'a'e

}my think

others think{

f0
mnvbdqlcjxzptk'p't'k'c'j

mnlbqdcxjzptk'k't'p'j'c

uyoraeiw'u'w'y'i'o'r'a'e

tɕ:4000~8000hz

ʂ:3000~7000hz

}others think

features{

Phonetically, fricatives (especially alveolar/palatal ones) show greater energy in the high-frequency range than affricates.(s>j)

}features

}is it lowest herz?

logic{

ab A bb B C ad bb ab A B ab A C ad.

mb ee mb ee de bf cjea bd ad oa ab anga ea de oa bb ab anga cjea ab anga bd ad mofg ga ab anga ea de oa ab na ab anga ea de oa

i dont know you=i know that you are only...

ae anga de de bf ih oa ad' is 'people also 0' and 'people also me' but we couldnt find other word.

}logic

word order{

discuss(20250621){

advatage{

OVS{

omitting parameter without ad.

math expression efficient?

Reverse Polish Notation (RPN) offers advantages over Polish Notation (PN) primarily in processing and storage efficiency, particularly in calculator implementations and compiler design. RPN simplifies the evaluation process by using a stack-based approach, requiring less complex logic and potentially reducing memory usage compared to PN. Additionally, RPN can be more efficient for complex calculations and is less prone to errors when entering equations

}OVS

VOS{

efficient for ea(what).

it can express image as efficient.

sentence 'this is people' is efficient.

good for int type parameter.

interpret with certainty.

}VOS

}advantage

result{

i decide to use VOS.

}result

}discuss

}word order

}research data

element of language{

WL is world language.

EL is for efficient.

CL is Cpu Language.

ML is Memory Language.

}element of language

ECL{

sentence 'A B' mean 'hey B, be A!'

exam{

//js to ecl

```
var a=0.

ihbgca.

for(var i=0;i<4;i++){

console.log(i)

}
goic ih gh ja fh gh jh mg gh lh da<-(da is enter, it for easy to read)
kb gh ad da
bc da
kccanc

'nc'.replace('fojb')
gjodcgcafojbcgbanc

'fojb'.includes('jb')
janjefcgbajbcgcafojb

if(x>1){

console.log(1)

}
cffhlhghgbkblhad

function A(var a){

console.log(a);
return a;

}
[p to c]gbkbghadkbghbcgdda
```

A||B
janalaghhh

}exam

qeqzejix•as,
word{
a=finish command,
b id=to other language,
c ,
d=line break,
e ,
f ,
g=finish standard,
h=finish sentence,
i
j
k
l
m
n
o
p
ab,
bb,
cb,
db,
eb,
fb,
gb =external memory to cache memory,
hb ML ss,ML s,ML s,CL s=for-loop,
ib
jb,
kb id,w=access to external devices,

lb
mb
nb
ob
pb
ad=omit-gramar,

}word

}ECL,

useless from here

useless from here

useless from here

useless from here

WL{

it is incomplete.plz ignore WL,go to BWTL,
sound of WL{

xxyyyyyy{

xx is frequency type,

yy is frequency,

}

(2bit:s or t or q or a.6bit:32n+32= of 8bit,

pronunciation expressed using NL ipa.

sin

32n{

aa:du. ea: bu. ia:'tu. ma:ju.

ab:lu.eb:'t'i.ib:dud.mb:dul.

ac:bws.ec:mus.ic:dup.mc:cub.

ad:y.ed:do.id:bul.md:qul.

ae:rm.ee:ym.ie:'trm.me:'twm.

}32n

maybe 128n{

aeim

ae:'vu.ee:mo.ie:'cu.me:dum,

af:'vo.ef:mom.if:'tum.mf:'co,

ag:om.eg:'pon.ig:'tus.mg:'nom,

ah:'pos.eh:'cus.ih:'trm.mh:'kuk,

ai:'tym.ei:'tu.ii:'pun.mi:'po,

aj:'jo.ej:'tos.ij:'k.mj:'pum,

ak:'tom.ek:'mo.ik:mu.mk:mr,

al:m'r.el:'t'r.il:'tw.ml:'tr,

| . .'tun |. . .'tom | .. .'ty |... .'ku ,

| . .'p'r |. . .'toq | . . .'tob |.. . .'prm ,

| .. .'kos |. .. .'kr | ... .'j |.... .'jum ,

| ..'kw |. ..'tul | . ..'cin |.. ..'tin ,

| . ..'pis |. . .'tis | .. ..'cun |... ..'ti ,

| ...'pin |. ...'p'is | . ...'tim |.. ...'t'i ,

| ....'k'is |. ....'t'is | .....'p'ipp'p'in ,

}maybe 128n

}sound of WL,

WCL{

}WCL

WML{

<Number of expression objects,Number of cases,frame,Minimum force,what compare>
-><dimension,bit,strength,frame,case>

sign{

rank of priority{

sev is several,
r is relative,
infi is infinity,
}rank of priority
}sign
a=finish-sentence-and-standard,
ad=finish-sentence,
world<0,0,0,0>,
none<0,0,0,0>,
time<0,0,0,2>,
elementary-particle<1,1,sev,1,19>,
exist<1,1,r,1,infi>,

atom<1,100~300,1,1.67×10^−27kg>,
molecule<1,300~,1,2×1.67×10^−27kg>,
appear/disappear<1,2,2,1>,
none change/change(s,r,p=<1,2,2,1>
light/heavy,(become light/become heavy=<1,2,2,2>,
(become low/become many=<1,2,2,mc^2>,
square<1,1,1,srpr>,
number<1,2,1,srpr>(1,
inner/out<1,2,1,1,7>,

this i/this u<1,2,1,2,1>,
under/down/left/up/right/on<1,6,1,1,7>,
line<2,1,1,2,6>,
move to (under/down/left/up/right/on=<1,6,2,2,14>,
stay/move<1,2,1,2,14>,
split/combine<(1'2.,2,2,2×mason,13>,
decrease/gain<(1'2.,2,2,3,13>,

}WML,

}WL,

}norlang