



Fruit - dpl

Programming language presentation

Motivation

- This language was created so you can code your food recipes (as long as it have a lot of fruits haha)





Features

This language is designed to create fruit salads, desserts with fruits, juices... Anything that you can do with fruit.

That being said:

- In your program all your variables and function names must be named after a fruit.
- The name of the fruit should be represented in Portuguese
- All the accentuation must be removed.
- For composite fruit names, we assumed CamelCase naming convention.
- You are only allowed to create variables and functions with names that doesn't follow the above rules 5 times.
- If you use different names, more times than the limit, your program will crash.

List of allowed words in the fruit language

```
FRUITS = {
    'abacate',      'abacaxi',  'abiu',      'abricó',      'abrunho',    'acai',      'acerola',    'akee',        'alfarroba',  'ameixa',     'amendoa',
    'amora',        'ananás',   'anona',    'araca',       'arando',     'araticum',  'ata',        'atemoia',     'avela',      'babaco',     'babacu',
    'bacaba',       'bacuri',   'bacupari', 'banana',     'baru',       'bergamota', 'biriba',     'buriti',      'butia',      'cabeludinha', 'cacau',
    'cagaita',      'caimito',  'caja',     'caju',        'calabaca',   'calabura',  'calamondin', 'cambuca',     'cambuci',    'camuCamu',   'caqui',
    'carambola',    'carnauba', 'castanha', 'castanhaDoPara', 'cereja',     'ciriguela', 'ciruela',    'coco',        'cranberry',  'cupuacu',
    'damasco',      'dekopon',  'dende',    'diospiro',    'dovyalis',   'duriao',    'embauba',    'embaubarana', 'engkala',    'escropari',
    'esfregadinha', 'figo',     'framboesa', 'frutaDoConde', 'frutaPao',   'feijoa',    'frutaDeCedro', 'frutaDeLobo', 'frutaDoMilagre', 'frutaDeTatu',
    'gabirola',     'glicosmis', 'goiaba',   'granadilla',  'gravata',    'graviola',  'groselha',   'grumixama',   'guabiju',    'guabirola',
    'guaraná',      'hawthorn', 'heisteria', 'hilocéreo',   'ibacurupari', 'ilama',     'imbe',       'imbu',        'inaja',      'inga',
    'inhare',       'jabuticaba', 'jaca',     'jambo',       'jambolao',   'jamelao',   'jaracatia',  'jatoba',     'jenipapo',   'jeriva',
    'jua',          'jujuba',   'kiwi',     'kumquat',     'kinkan',     'kino',      'kiwano',     'karite',      'laranja',
    'limao',        'lima',     'lichia',   'longan',      'lucuma',     'lacucha',   'lulo',       'lobeira',     'langsai',    'laranjaDePacu',
    'mabolo',       'maca',     'macadamia', 'macauba',     'mamao',      'mamey',     'mamoncillo', 'manaCubiu',   'manga',      'mangaba',
    'mangostao',    'maracuja', 'marang',   'marmelo',     'marolo',     'massala',   'melancia',   'melao',      'melo',       'nozPeca',
    'mexerica',     'mirtilo',  'morango',  'murici',      'naranjilla', 'nectarina', 'nespera',    'noni',        'noz',         'nozPeca',
    'nozMacadamia', 'oiti',     'oxicoco',  'orangel',     'pera',       'pessego',   'pitanga',    'pinha',      'pitaia',     'pitomba',
    'pitangatuba',  'pindaiba', 'pequi',    'pequia',     'physalis',   'pulasan',   'pomelo',     'pupunha',     'puca',       'pataua',
    'pajura',       'pixirica', 'pistache', 'quina',      'quiui',      'roma',      'rambutao',   'rukam',      'saguaraaji',
    'salak',        'santol',   'sapota',   'sapoti',      'sapucaia',   'saputa',    'seriguela',  'sorvinha',   'tangerina',  'tamarindo',
    'tamara',       'toranja',  'tucuma',   'taiuva',     'tapia',      'taruma',    'tangor',     'tucuja',      'uva',        'umbu',
    'uvaia',        'uchuva',   'ume',      'uxi',         'vacinio',    'veludo',    'vergamota',  'wampi',      'xixa',       'yamamomo',
    'yuzu',         'zimbro'
}

MIX = { 'leite', 'acucar', 'sal', 'agua', 'manteiga', 'leiteCondensado' }

FUNCTIONS = { 'Main', 'mistura', 'tempera', 'junta' }
```



Curiosities

- More than 216 different brazilian fruits were include in this language.
- There are 10 extra non fruit names available in this language
- The language allow you to name more 10 variables that aren't included in the above group
- That means that in this language, one program can only have a MAXIMUM of 236 different names (used for function names, function args, and variable names).



Example 1 - Code that runs

```
// t2.fruit
receita Main() {
    ingrediente banana, acucar, leite, abacate : inteiro;
    ingrediente copo : inteiro;

    acucar recebe 200;
    banana recebe 100;
    leite recebe 200;
    abacate recebe 100;

    copo recebe fazVitamina(banana, acucar, leite, abacate);

    se (copo temMaisQue 600) {
        mostra("Deu ruim, tinha mais vitamina do que o limite do liquidificador!");
    }
    casoContrario {
        mostra("Muito bom, a vitaminha ficou uma delicia!");
    }
}

receita fazVitamina(i1: inteiro, i2: inteiro, i3: inteiro, i4: inteiro) -> inteiro {
    resultado (i1 com i2 com i3 com i4);
}
```

Output

```
└─ python3 main.py t2.fruit
>> Muito bom, a vitaminha ficou uma delicia!
```



Example 1 - Code that runs

```
// t2.fruit
receita Main() {
    ingrediente banana, acucar, leite, abacate : inteiro;
    ingrediente copo : inteiro;

    acucar recebe 200;
    banana recebe 100;
    leite recebe 200;
    abacate recebe 100;

    copo recebe fazVitamina(banana, acucar, leite, abacate);

    se (copo temMaisQue 600) {
        mostra("Deu ruim, tinha mais vitamina do que o limite do liquidificador!");
    }
    casoContrario {
        mostra("Muito bom, a vitaminha ficou uma delicia!");
    }
}

receita fazVitamina(i1: inteiro, i2: inteiro, i3: inteiro, i4: inteiro) -> inteiro {
    resultado (i1 com i2 com i3 com i4);
}
```

Output

```
└─ python3 main.py t2.fruit
>> Muito bom, a vitaminha ficou uma delicia!
```

Example 2 - Code that doesn't runs

```
receita Main() {  
    ingrediente banana, acucar, leite, abacate : inteiro;  
    ingrediente copo : inteiro;  
    ingredientes invalidName1, invalidName2, invalidName3, invalidName4, invalidName5, invalidName6: inteiro;  
  
    mostra("Esse programa não deve rodar, pois tem mais variáveis com nome invalidos, do que o permitido");  
}  
  
receita fazBolo(i1: inteiro, i2: inteiro, i3: inteiro, i4: inteiro) -> inteiro {  
    resultado return i1 com i2 sem (i3 multiplicadoPor i4);  
}
```

Output

```
└─ python3 main.py t3.fruit ─┐  
Traceback (most recent call last):  
  File "/home/daniel/Desktop/Insper/logcomp/dpl/main.py", line 1363, in <module>  
    main()  
  File "/home/daniel/Desktop/Insper/logcomp/dpl/main.py", line 1360, in main  
    res = run(source_code)  
  File "/home/daniel/Desktop/Insper/logcomp/dpl/main.py", line 1338, in run  
    tokens = lexer.make_tokens()  
  File "/home/daniel/Desktop/Insper/logcomp/dpl/main.py", line 215, in make_tokens  
    tokens.append(self.make_keyword_or_identifier())  
  File "/home/daniel/Desktop/Insper/logcomp/dpl/main.py", line 174, in make_keyword_or_identifier  
    raise Exception(f"More than {self.MAX_ALLOWED_DIFFERENT_VARIABLES} were found in the source code.\nVars: {self.not_allowed_name_ocurrences}")  
Exception: More than 10 were found in the source code.  
Vars: {'ingredientes', 'invalidName6', 'i1', 'invalidName1', 'invalidName3', 'invalidName5', 'fazBolo', 'invalidName4', 'copo', 'invalidName2'}
```




Example 3 - Complex code that runs

```

receita Main() {
    // All bool and int operations
    ingrediente uva, limao : inteiro;

    ingrediente copo1, copo2 : inteiro;
    ingrediente conclusao : texto;

    conclusao recebe "Agua com limao é bom";

    uva recebe 99;
    limao recebe (uva ehIgualzinho 99);

    mostra(uva);
    mostra(limao);

    mostra(uva com limao); // 100
    mostra(uva sem limao); // 98
    mostra(uva multiplicadoPor limao);
    mostra(uva divididoPor limao);
    mostra(uva ehIgualzinho limao);
    mostra(uva temMenosQue limao);
    mostra(uva temMaisQue limao);
    mostra(conclusao);

    copo1 recebe 0;
    copo2 recebe 10;

    enquanto (copo1 temMenosQue copo2) {
        mostra(copo1);
        copo1 recebe copo1 com 1;
    }

    ingrediente laranja : inteiro;
    laranja recebe entrada();

    se (laranja temMaisQue 10) {
        mostra("Laranja > 10. Bolo de banana:");
        mostra(boloDeBanana(10));
    }

    casoContrario {
        mostra("Laranja < 10. Salada de frutas:");
        mostra(saladaDeFrutas(70, 7));
    }
}

receita boloDeBanana(banana: inteiro) -> inteiro {
    resultado 8000;
}

receita saladadeFrutas(banana: inteiro, maca: inteiro) -> inteiro {
    resultado banana com maca;
}

```

Output

```

└─ python3 main.py t1.fruit
>> 99
1
100
98
99
99
0
0
1
Agua com limao é bom
0
1
2
3
4
5
6
7
8
9
300
Laranja > 10. Bolo de banana:
8000

```