

Week 1 Opgave 3

Nand

Implementeer een NAND-functie in Python3 zonder gebruik te maken van de Python AND/OR of NOT operatoren. De definitie van deze functie moet luiden:

```
def myNand(a, b):  
    """ Nand-functie: returns de result of nand(a, b)  
        Uses if statements  
    Arguments  
        a: 0 of 1  
        b: 0 of 1  
    Result  
        r: 0 of 1  
    """  
    # ... write your code and return 0 or 1  
    return 0
```

Programma-files

Gegeven is de volgende code : w1o3.py.

Vervolgens implementeer je de functies myNot, myAnd, myOr, myNor and myXor met alleen myNand aanroepen.

Test

De bijbehorende file `w1o3.py` bevat de test die de goede werking van het programma demonstreren.

```

def test1(opr, testSet):
    """ Test operator with one argument
        Arguments
        opr:    function to test
        testSet: list of arg, res
    """
    for a, r in testSet:
        rr = opr(a)
        if rr != r:
            print('Error: {0:s}({1:d}): {2:d} != {3:d}'.format(
                opr.__name__, a, r, rr))

def test2(opr, testSet):
    """ Test operator with two arguments
        Arguments
        opr:    function to test
        testSet: list of args, res
    """
    for (a, b), r in testSet:
        rr = opr(a, b)
        if rr != r:
            print('Error: {0:s}({1:d}, {2:d}): {3:d} != {4:d}'.format(
                opr.__name__, a, b, r, rr))

# Test your code !!
if __name__ == '__main__':
    test1(myNot, [(0, 1), (1, 0)])
    test2(myNand, [((0, 0), 1), ((0, 1), 1), ((1, 0), 1), ((1, 1), 0)])
    test2(myAnd, [((0, 0), 0), ((0, 1), 0), ((1, 0), 0), ((1, 1), 1)])
    test2(myOr, [((0, 0), 0), ((0, 1), 1), ((1, 0), 1), ((1, 1), 1)])
    test2(myNor, [((0, 0), 1), ((0, 1), 0), ((1, 0), 0), ((1, 1), 0)])
    test2(myXor, [((0, 0), 0), ((0, 1), 1), ((1, 0), 1), ((1, 1), 0)])
    print('tests completed')

```

Help

Het behulp van "PEP 257 -- Docstring Conventions" <https://www.python.org/dev/peps/pep-0257> kun je je programma van commentaar voorzien. Op deze manier kun je functies van help-informatie voorzien.

```

$ python3
Python 3.7.1 (v3.7.1:260ec2c36a, Oct 20 2018, 03:13:28)
[Clang 6.0 (clang-600.0.57)] on darwin
Type "help", "copyright", "credits" or "license" for more information.
>>> import wlo2
>>> help(wlo2)

```

Vervolgens wordt de volgende help-tekst getoond.

```
Help on module wlo2:
```

NAME

```
    wlo2
```

FUNCTIONS

```
    myAnd(a, b)
```

```
    myNand(a, b)
```

```
        Nand-functie: returns de result of nand(a, b)
```

```
        Uses if statements
```

```
        Arguments
```

```
            a: 0 of 1
```

```
            b: 0 of 1
```

```
        Result
```

```
            r: 0 of 1
```

```
    myNor(a, b)
```

```
    myNot(a)
```

```
    myOr(a, b)
```

```
    myXor(a, b)
```

```
    test1(opr, testSet)
```

```
        Test operator with one argument
```

```
        Arguments
```

```
        opr:    function to test
```

```
        testSet: list of arg, res
```

```
    test2(opr, testSet)
```

```
        Test operator with two arguments
```

```
        Arguments
```

```
        opr:    function to test
```

```
        testSet: list of args, res
```

VERSION

```
    1.0
```

AUTHOR

```
    (c) 2019 Frans Schippers (f.h.schippers@hva.nl) for CSP
```

FILE

```
    .../wlo3.py
```

Inleveren

Zorg dat je naam in het programma staat (`__author__`) en lever jouw versie van `w1o3.py` in op VLO. Inleveren voor vrijdag 8 feb 24:00.