

## Fondamentaux du Web (WEB'17)

### TD 2 : Anatomy of programming languages

The goal of every exercise is to write the code given in Python using a language the you do not know already and that you would like to learn. Not all the code has an equivalent in every language, or a code that looks equivalent may in fact execute differently. Preference is for structured dynamically-typed, object oriented languages such as JavaScript, Python, Ruby, PHP, or even Java (the later is however nominatively-typed).

#### Exercise 0 : Hello world

```
print "Hello World!"
```

### Data Types

Here, we explore data values of various types

#### Exercise 1 : Basic data types (int, float, string, bool, null)

```
print 1
print 1.0
print 'hello'
print True
print False
print None
```

#### Exercise 2 : Composite data types (tuple, list, map)

```
t = (1,2,3)
l = (1,2,3)
m = {1:'a', 2:'b', 3:'c'}
print t[1]
print l[-1]
print m[2]
print len(l)
print len(m)
```

### **Exercise 3 : Higher-order types**

```
def f(x):  
    def f_aux(y):  
        return x + y  
    return f_aux
```

```
g = f(3)  
print g(5)
```

```
def f(x):  
    return 2 * x
```

```
def g(x):  
    return x + 1
```

```
def compose(f,g):  
    def res(x):  
        return f(g(x))  
    return res
```

```
h = compose(f,g)
```

```
print h(2)
```

$\lambda$ -expressions

```
f = lambda x : (lambda y : x + y)  
g = f(3)  
print g(5)
```

```
f = lambda x : 2 * x
```

```
g = lambda x : x + 1
```

```
compose = lambda f,g : lambda x : f(g(x))
```

```
h = compose(f,g)
```

```
print h(2)
```

### **Exercise 3 Modularity** How is the code divided into modules/packages and how to access them

```
import random  
from math import ceil  
  
print random.randint(0,10)  
print ceil(10.1)
```

## Data Flow

Here, we explore data manipulation operations as well as concepts of scope, mutability of data structures, and call by reference.

### Exercise 4 : Name scope

```
a = 0

def f(b):
    a = b

def g(b):
    global a
    a = b

print a
f(2)
print a
g(2)
print a
```

### Exercise 5 : Arithmetic expressions and default type conversion

```
print 1 + (2 * 3) / 0.25
print int(1 + (2 * 3) / 0.25)
print 14 / 5
print 14 % 5
print 2 ** 10
```

### Exercise 6 : String operations

```
print 'ab' + 'c'
print 'ab' * 2
print "this %s and that %d" % ('1', 2)
print "this {0} and that {1}".format('1', 2)
print "this {first} and that {second}".format(first='1', second=2)
```

### Exercise 7 : Conditional expression

```
import random

x = 0 if random.choice([False, True]) else 1

print x
```

### **Exercise 8 : Boolean expressions**

```
def f():
    print "f"
    return False
def t():
    print "t"
    return True
print "F AND T"
print f() and t()
print "T AND F"
print t() and f()
print "F OR T"
print f() or t()
print "T OR F"
print t() or f()
print "NOT T"
print not t()
```

```
x = None
y = [1,1]
y_ = y
z = [1]*2
```

```
print x == y
print y == y_
print y == z
print x is None
print x == None
print y is None
print y == None
print y is y_
print y is z
```

### **Exercise 9 : Immutability**

```
l=[1,2,3]
print l[1]
l[1] = 4
print l
```

```
t=(1,2,3)
print t[1]
t[1] = 4
print t
```

### **Exercise 10 : Call by reference**

```
def f(l):
    l = l * 2
    print "inside", l

x = 1
f(x)
print "outside", x

y = [1,1]
f(y)
print "outside", y
```

## **Control Flow**

### **Exercise 11 : Conditional instruction**

```
import random

if random.choice([True,False]):
    print "it worked :)"

if random.choice([True,False]):
    print "it worked :)"
else:
    print "it didn't work :)"
```

### **Exercise 12 : While loop**

```
import random

l = ['a','b','c','d','e','f','g','h']

i = 0
while i < len(l) :
    if random.choice([False, False, False, False, True]):
        print "I give up"
        break
    if random.choice([False, False, True]):
        print "I'll skip this one"
        i += 1
        continue
    print "the secret letter is", l[i]
    i += 1
```

### **Exercise 13 : For loop**

```
import random

l = ['a','b','c','d','e','f','g','h']

for c in l:
    if random.choice([False, False, False, False, True]):
        print "I give up"
        break
    if random.choice([False, False, True]):
        print "I'll skip this one"
        continue
    print "the secret letter is", c
```

### **Exercise 12 : Invoking methods**

```
s = 'abcdeabcde'
print s.index('abc',3)
```