

CSCA08 TUTORIAL WEEK 3

TUT0014

TA: Andrew Wang

E-mail: andrewpy.wang@mail.utoronto.ca

Website: <http://andrewwang.ca/teaching/17fall/csca08/>

THIS WEEK

Review

(Lots of) memory models

Python Tutor

DATA TYPES

TYPE CONVERSION

```
>>> float(1)
```

```
>>> int(2.9)
```

```
>>> str(2.9)
```

```
>>> int("1")
```

```
>>> float("2.9")
```

OPERATORS

```
>>> 34 + 8
```

```
>>> 3 ** 2
```

```
>>> 14 * (3 + 2)
```

```
>>> 29 / 3
```

```
>>> 5.5 * 4.0
```

```
>>> "abc" + "def"
```

```
>>> 29 // 3
```

```
>>> 28 / 3.0
```

```
>>> "2.9" + "2.9"
```

```
>>> 18 - 2.9
```

```
>>> 28 // 3.0
```

```
>>> 2.9 + float("2.9")
```

```
>>> 24 % 5
```

```
>>> 14 * 3 + 2
```

```
>>> 1 + "2.9"
```

FUNCTIONS

```
def function_name(parameters):
```

```
    # chunk of code
```

```
    return ...
```

SCOPE

Global variables:

- Created outside a function
- Can be accessed, but cannot be changed inside a function

Local variables:

- Created inside a function
- Get “lost” after the function is executed

MEMORY MODEL

Tedious, but really really important

Will be at least 50% on term test 1

Be very careful!


```
x = 7
y = 10
x = 8
x = 10
x = 8
x = y
y = 15
z = x + y
print(z)
x = "Hello"
y = 5
z = x + y
print(x * y)
```

```
x = 7
```

```
y = 10
```

```
x = 8
```

```
x = 10
```

```
x = 8
```

```
x = y
```

```
y = 15
```

```
z = x + y
```

```
print(z)
```

```
x = "Hello"
```

```
y = 5
```

```
z = x + y
```

```
print(x * y)
```

```
#now let's try functions
```

```
def my_function(x):
```

```
    y = x + 7
```

```
    print(y)
```

```
    return "Hello"
```

```
y = my_function(3)
```

```
print(y)
```

```
#now more functions
def func_a():
    x = 7
def func_b():
    x = 7
    return x
def func_c(x):
    x = 7
def func_d(x):
    return x
y = func_a()
print(y)
y = func_b()
print(y)
x = 10
y = func_c(x)
print(x, y)
y = func_d(x)
print(x, y)
```

```
#and functions within functions
def func_a(x):
    x = x + 7
    print(x)
    return x
def func_b(x):
    x = x + func_a(x)
    print(x)
    return x
def func_c(x):
    x = x + func_b(x)
    print(x)
    return x
y = func_c(3)
print(y)
```

```
def func1(my_var):  
    print(my_var)  
    my_var = "A"  
    print(my_var)  
    return my_var  
  
def func2(my_var,my_var2):  
    print(my_var)  
    my_var = "B"  
    my_var2 = "C"  
    my_var2 = func1(my_var)  
    print(my_var)  
    print(my_var2)  
    my_var = my_var2  
    return my_var  
  
my_var = "D"  
print(my_var)  
my_var = func1(my_var)  
print(my_var)  
my_var = func2(my_var,my_var)  
print(my_var)
```

PYTHON TUTOR

<http://www.pythontutor.com/>

You need to use the following settings:

- Python 3.6
- Hide exited frames [default]
- Render all objects on the heap
- Use text labels for pointers

REMINDERS

<http://individual.utoronto.ca/andrewwang/a08/>

Exercise 1: Testing code on Piazza

Exercise 1 dues Friday @5pm