



# Programming for Social Scientists

---

Johan A. Dornschneider-Elkink

## Variables and functions

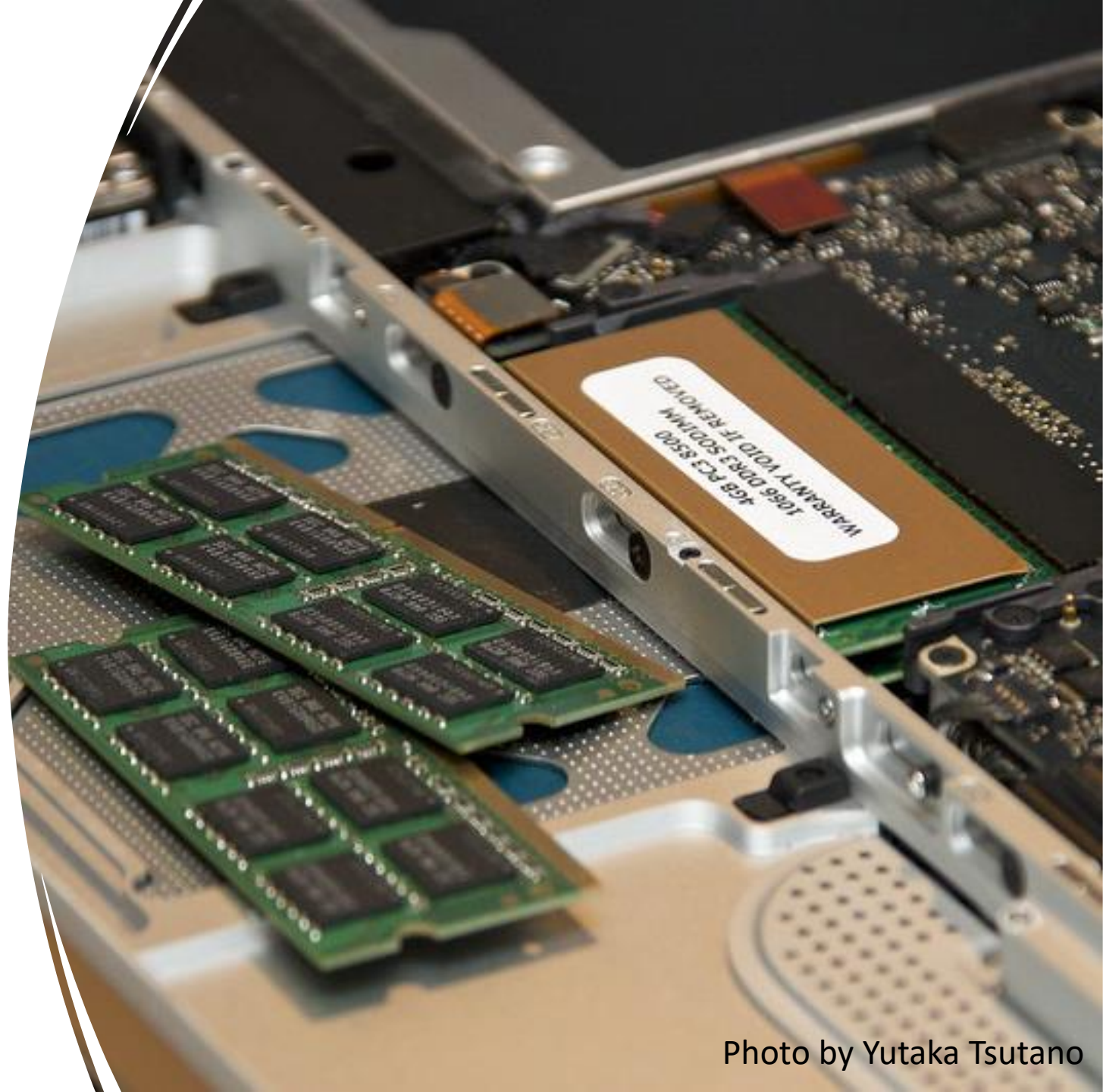
# Variables

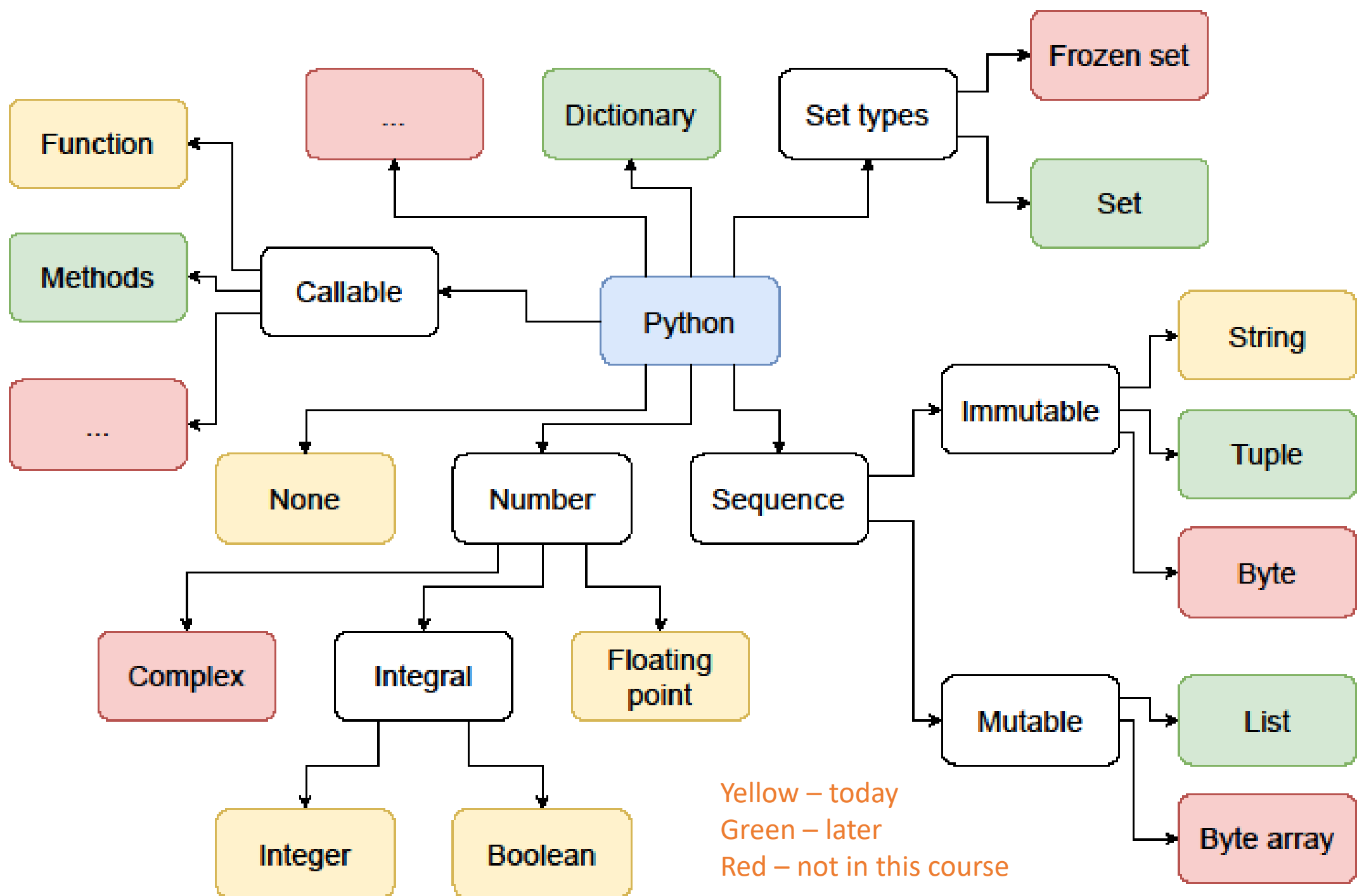
---

Data storage in computer memory

Not to be confused with a variable in:

- statistics
- a mathematical function







# Numerical types

<b>int</b>	Integer number, i.e. no decimal point or fraction. E.g. 100, 356, etc.
<b>float</b>	Floating point number, i.e. has a decimal point. E.g. 4.0, 3.1428, etc.
<b>bool</b>	Variable that only stores two values, True or False
<b>complex</b>	Stores complex numbers, which have a real and an imaginary part

# String (text) types

**str**

Sequence of letters, forming a string of text.



Photo by Ivan Radic

# None type

**NoneType** Contains no value.

# Variables in python

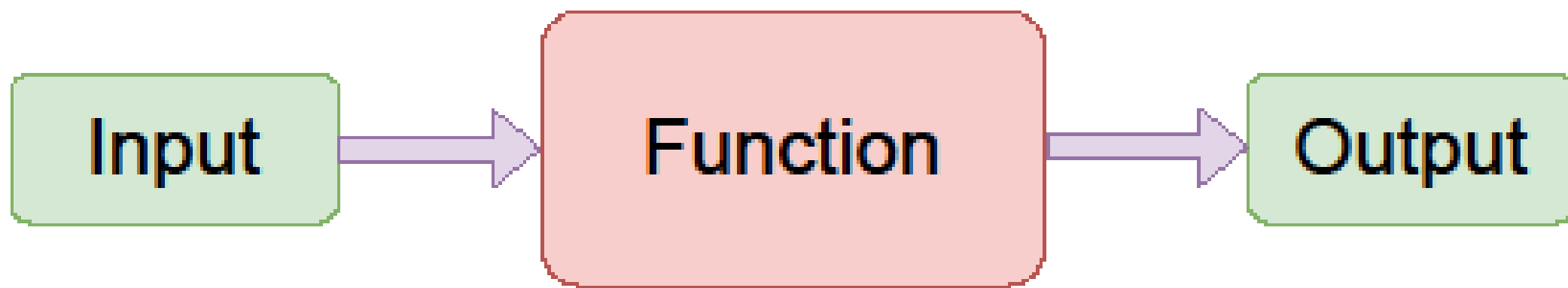
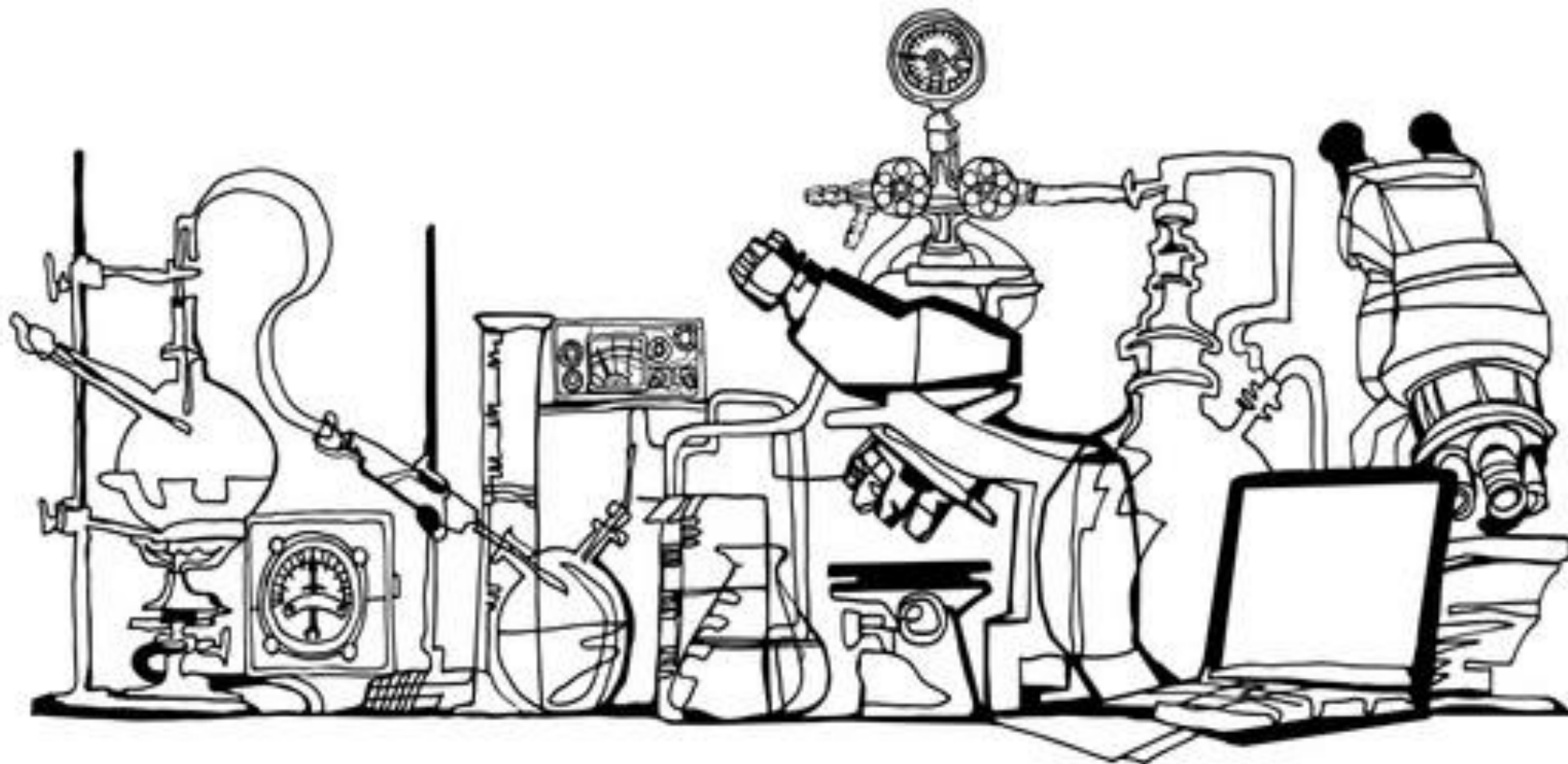


**DYNAMICALLY TYPED**



**GARBAGE-COLLECTED**







$$f(x, y) = x^y$$

$$f(x, y) = x^y$$

```
def power(x, y):  
    return(x ** y)
```

```
power(2, 4)
```

$$f(x, y) = x^y$$

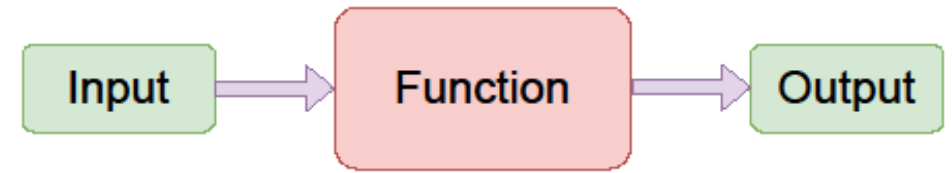
Function definition  
and implementation

```
def power(x, y):  
    return(x ** y)
```

```
power(2, 4)
```

Function call

$$f(x, y) = x^y$$



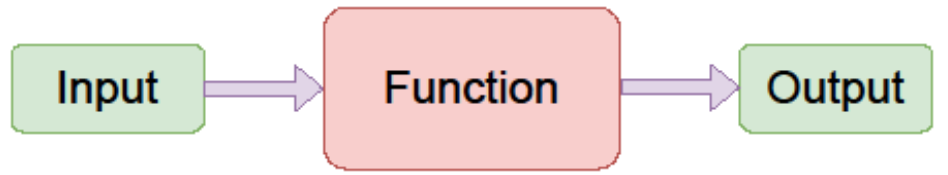
Input

```
def power(x, y):  
    return x ** y
```

```
power(2, 4)
```



$$f(x, y) = x^y$$



```
def power(x, y):  
    return x ** y  
  
power(2, 4)
```

Input

Output

```
s = "Pythons are scary!"
```

```
print(len(s))
```

```
print(s.upper())
```

Using built-in  
functions

```
s = "Pythons are scary!"
```

```
print(len(s))
```

```
print(s.upper())
```

Using the output of the len() function as the input of the print() function

```
s = "Pythons are scary!"
```

```
print(len(s))
```

```
print(s.upper())
```

Method call



```
fs = 20
```

```
def upper_firstname(name):  
    fs = name.find(" ")  
    return s[0:fs].upper() + s[fs:]
```

```
upper_firstname("Jos Elkink")  
'JOS Elkink'
```

```
fs  
20
```

*Variable scope*: the part of the code where the variable is valid.

Using a variable name in a function that also occurs elsewhere in the program does not change that other variable.

```
def upper_firstname(name):  
    nf = name.find(" ")  
    return s[0:nf].upper() + s[nf:]
```

```
upper_firstname("Jos Elkink")  
'JOS Elkink'
```

nf

**NameError: name 'nf' is not defined**

**Variable scope:** the part of the code where the variable is valid.

Using a variable name in a function that also occurs elsewhere in the program does not change that other variable.

Parameters and variables defined inside a function are only accessible inside the function itself.