Functions in R & Python

A practical introduction

Lima, Perú
08 October 2022

Content

- Introduction to function
  - Coding vs programming
  - Benefits of programming
  - Introduction to Python
  - Function structure (Python vs R)

- Create a source (function dictionary)

Illustrations by Pixeltrue on icons8

Coding vs programming

Instead these words are often used interchangeably, they actually signify two very different things.

Coding	Programming
Translate the requirement logics into a machine-understandable code without worrying about the details	Analyze and conceptualize different aspects of any program and also solutions to any problems that may or may not occur due to the process
Basic knowledge of programming skills without any software tools	Good knowledge of programming skills and good software development tools
Helps to work with simple projects	helps to work with big projects and manage them

Coding vs programming

So, is it better to learn coding or programming?

Coding: occasional use, general solutions, but also high demand

Programming: frequent use, specific solutions, high demand

Benefits of programming (functions)

Why functions are useful?

Reusability

It can be called multiple times from different spaces. This avoids duplicated code

Modularity

Divide a complex problem into multiple functions. Better tracking

Testing

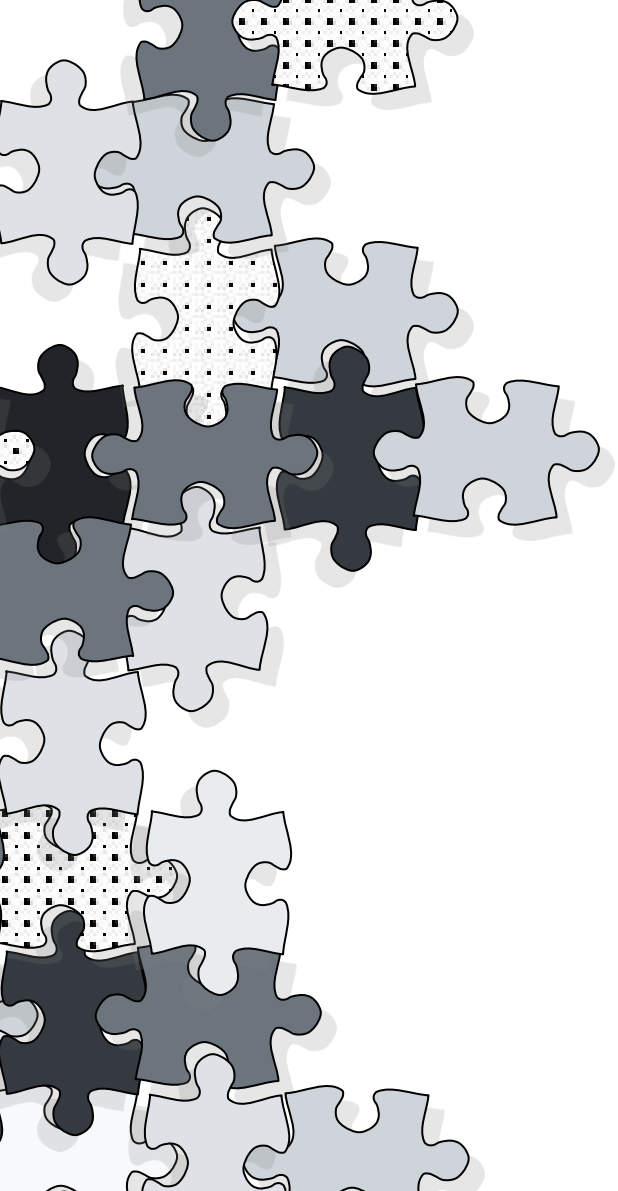
Less code to test, easier to find bugs. Test a function just once time

Extensibility

Functions reduce the amount of code that has to be written

Abstraction

You don’t need to know how it works, or what other code it’s dependent upon to use it



> Imagine the first 10 employees of Google, it’s pretty clear they didn’t have a PHD in computer science, they had **programming experience**
>
> - Dr. Chuck Severance

Python

What is it?

It's a **high level, dynamic "multiparadigm" (maybe more OOP than others) language.** Its core philosophy is **readability.**
Typing in Python is based on "classes", which allow programmers to create their own types.

Why it matters?

Because of its readability and broad range of libraries it is one of the more accessible and popular languages for Data Science today, it has also been implemented as an scripting language in multiple softwares which allows interdisciplinary approaches

Function structure

Basic Logic

- A function is a block of code which performs a task.
  - It has a **name**; which we use to call the function
  - It has a **void**; which means it has “no value”
  - It has an **argument**; which is the mini-program inside

Photo by Dave Hoefler on Unsplash

C++ and Python

Objective: Return a signal for the execution of a function

```cpp
// Create a function
void myFunction() {
  cout << "I just got executed!";
}

int main() {
  myFunction(); // call the function
  return 0;
}

// Outputs
"I just got executed!"
```

C++

```python
#Create a function
def my_function():
  print("I just got executed!")

#Call the function
my_function()

#Outputs
"I just got executed!"
```

Python

R and Python

Objective: Create a function which tells if a number is odd or even

```r
# Create a function
evenodd = function(x)
{
        if (x %% 2 == 0)
                return(“even!”)
        else
                return(“odd¡”);
}

# Call the function
print(evenodd)(4)) ;

# Outputs
"even!"
```

R

```python
#Create a function
def evenodd(x):
        if x%2 == 0:
                        print("even!")
                else:
                        print("odd¡")
#Call the function
evenodd(4)

#Outputs
“even!”
```

Python

R and Python

Objective: Create a function which tells if a number is odd or even

```r
# Create a function
evenodd = function(x)
{
        if (x %% 2 == 0)
                return(“even!”)
        else
                return(“odd¡”);
}

# Call the function
print(evenodd)(4)) ;

# Outputs
"even!"
```

R

```python
#Create a function
def evenodd(x):
        if x%2 == 0:
                        print("even!")
                else:
                        print("odd¡")
#Call the function
evenodd(4)

#Outputs
“even!”
```

Python

Python Dive

Objective: Return a signal for the execution of a function

Arguments get a value when called

```
#Create a function
def my_function(fname):
  print("Hello from a function called: ", fname)

#Call the function
my_function(“my first function”)

#Outputs
“Hello from a function called: my first function”
```

...And you can have multiple values

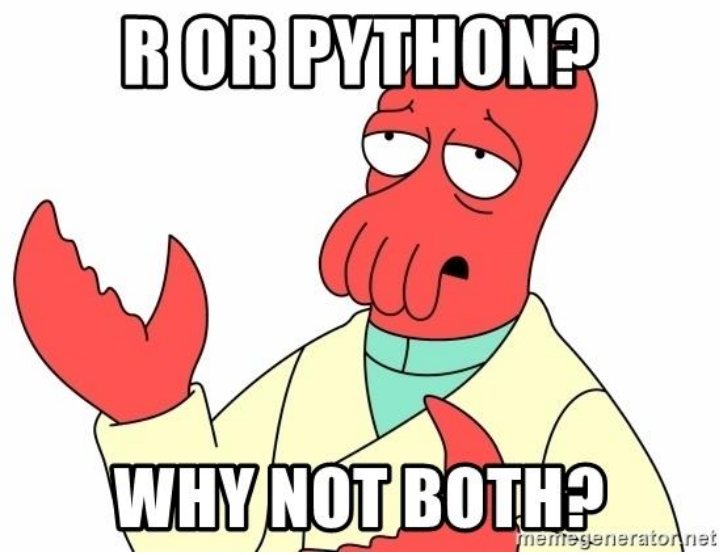
```
#Create a function
def my_function(fname, flastname):
  print("Hello from a function called ", fname, flastname)

#Call the function
my_function(“my first function”, “is quite simple”)

#Outputs
“Hello from a function called: my first function is quite simple”
```

Create a source (function dictionary)

Problem: How automate sum of squares?



The thing with creating Python Source…

And why some people say its and Object Oriented Language

Its impossible to call a function directly from somewhere outside the main file/code

Or at least, it ain’t the easy way…

Python Dive

Objective: Create a function which returns the square of a number

```
class square:
#x square
        def _init_(self, x, y):
                self.x = x
                self.y= y
        def sq(x):
                return x*x
#Call function
print(square.sq(4))
#Outputs
16
```

Functions in R & Python

A guide for a deeper dive

Lima, Perú

15 October 2022

Content

- Good Practices in coding
  - Naming conventions
  - Indents and spaces
- Source/package management
  - Anaconda vs. Pip (Python)
- Introduction to functional programming

Illustrations by Pixeltrue on icons8

Thanks to Stackoverflow (and others) coding is easier everyday...

But sometimes answers can lead to ugly solutions

It works! But, can I make it more simple?

```
<script>
        function down_visibility(id1,id2,id3,id4,id5,id6,id7,id8) {
          var e1 = document.getElementById(id1);
          var e2 = document.getElementById(id2);
          var e3 = document.getElementById(id3);
          var e4 = document.getElementById(id4);
          var e5 = document.getElementById(id5);
          var e6 = document.getElementById(id6);
          var e7 = document.getElementById(id7);
          var e8 = document.getElementById(id8);

         if(e1.style.display == 'none')
            e1.style.display = 'block';
         else if(e1.style.display == 'block')
           { e1.style.display = 'none'; }

            e2.style.display = 'none';
            e3.style.display = 'none';
            e4.style.display = 'none';
            e5.style.display = 'none';
            e6.style.display = 'none';
            e7.style.display = 'none';
            e8.style.display = 'none';
           }
  </script>
```

“Beautiful” code it’s inherently subjective

But some standards have become common practice:

- Naming should always be sensible and direct to avoid misinterpretations
    - When working within an organization, we should always ask about naming conventions and nomenclatures
- Indents should be used for better readability (mandatory use in Python)
- Ockham’s Razor: Avoid Copy+Pasting your own code
- Comment always you can

Recycled example

Objective: Create a function which tells if a number is odd or even

```
#Create a function
def evenodd(x):
        if x%2 == 0:
                print("even!")
        else:
                print("odd¡")
#Call        the function
evenodd(4)

#Output should be “even!”
```

sensible naming

Nesting and indentation

Commenting

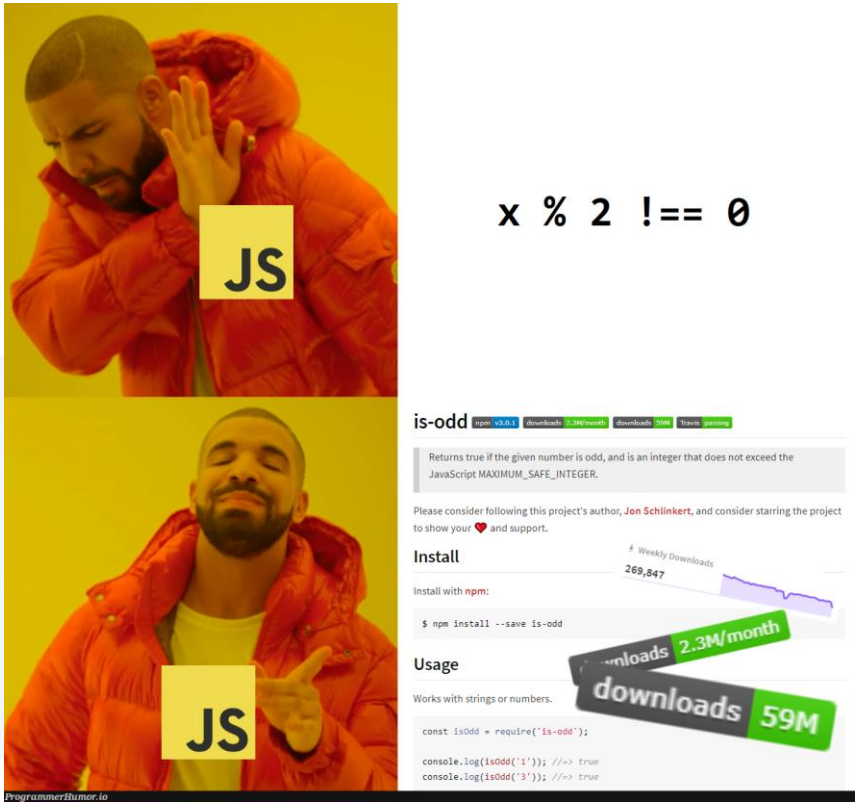
When working on a group…

Practices inside organizations can vary and we should check:

- NDA's and private information that the organization wishes to avoid naming
- Previous coding documentation and packages in use
- Security limitations and sharing conventions

Nowadays, packages or libraries will (most likely) be used extensively in your projects

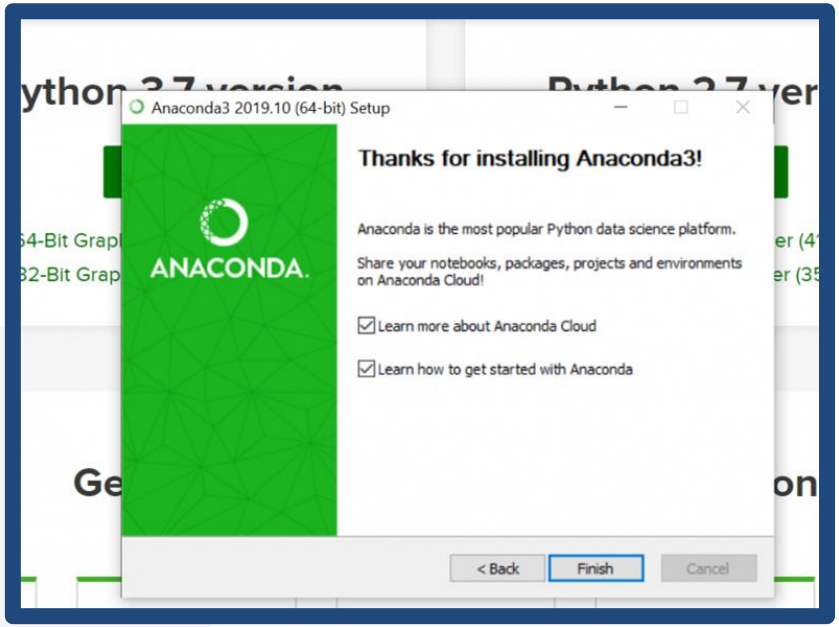
Managing many packages can be tricky…



So we should always keep ours in check

- Ockham's razor: Loose some packages
- Update (if not in production)
- Manage your sources:
  - Follow the practices though before, but for your packages
  - Explore if you need to multiply your sources
    - Main, per project or per client are some of the options.

Maybe you have heard about Anaconda



Anaconda is a package manager for Python (and R)

- You have to install it before Python or R, or your system might have a hard time
- Helps with package management by updating automatically based on performance
- Has its own “conda-forge” where people can share their own packages
- Conda might be late to some updates that are distributed mainly on pip (ie: psysql)

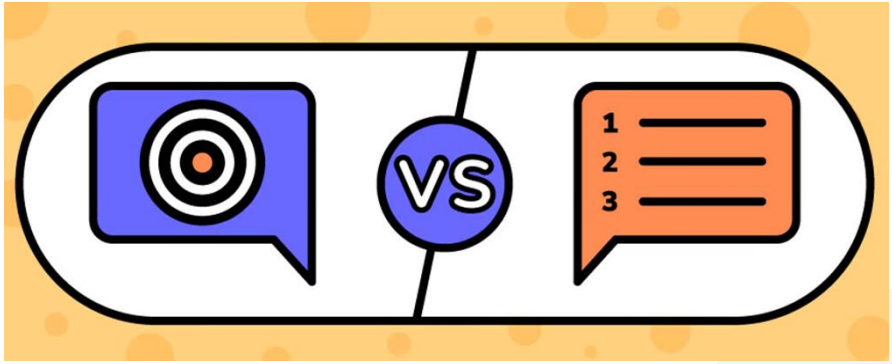
Functional programming

What's a programming paradigm?

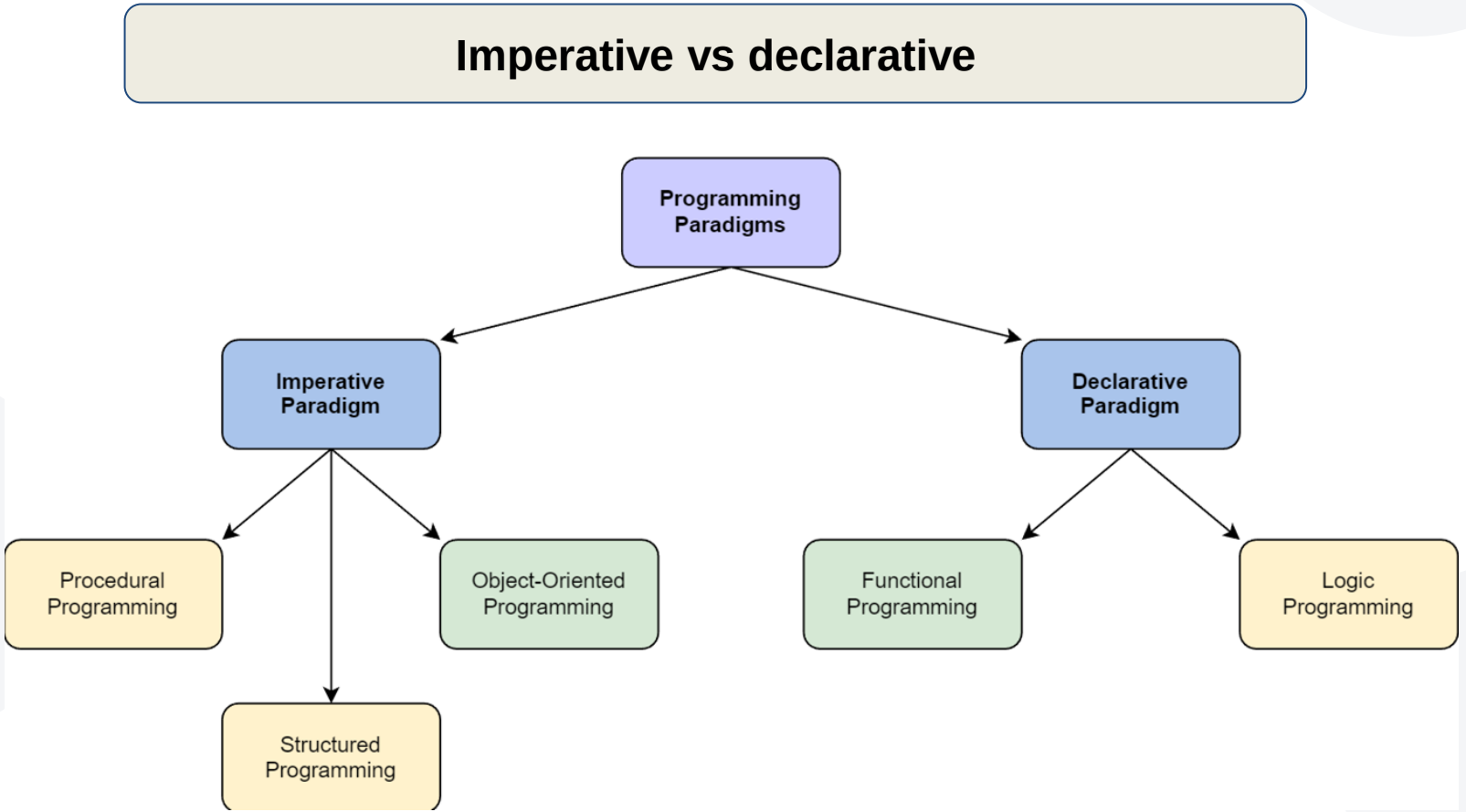
A method to solve a problem

Imperative
Programs specify how it is to be done; direct dictation [JAVA, C, C++]

Declarative
Programs specify what is to be done; desire result [SQL]



Functional programming



Functional programming

Concepts

- Pure functions
- Function composition
- Recursive functions
- High order functions