



Introduction to Computer programming with Python

What is computer programming

- *A computer program is a sequence of instructions written using a Computer Programming Language to perform a specified task by the computer.*

Note:

- 1. Sequence of instructions*
- 2. Computer programming language*

Sequence of instructions & Languages

Consider cooking the recipe for making egusi soup, There are steps that you need to take so that the soup would come out tasty and appeal to the sight.

1. In a large pot, heat the palm oil on medium for a minute and then add the Une.
2. Slowly add the stock and set on low heat to simmer.
3. Scoop teaspoon size balls of the egusi paste mixture into the stock. Be sure to keep ball shape.
4. Leave to simmer for 20 – 30 minutes so the balls cook through.
5. Add the meat and fish and other bits which you'd like to use.
6. Add cut-up pumpkin leaves.
7. Add the waterleaf.
8. Stir and put a lid on the pot and allow cook for 7–10 minutes, till the leaves wilt.
9. Add the bitter leaf. Leave the lid off while the cooking finishes for another 5-10 minutes.
Stir, check seasoning and adjust accordingly.



Here we used English language to convey the steps that should be taken in preparing egusi soup.

Now, try to map the situation with a computer program. The above sequence of instructions is actually a **Human Program** written in **English Language**, which instructs on how to prepare. This same sequence could have been given in Igbo, Yoruba, Hausa, or any other human language, provided the person cooking knows any of these languages.

Now, let's go back and try to understand a computer program, which is a sequence of instructions written in a Computer Language to perform a specified task by the computer. Following is a simple program written in **Python** programming Language

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

The above computer program instructs the computer to print "Hello, World!" on the computer screen.

Programming Languages

A programming language is a **computer language** that is used by **programmers (developers)** to **communicate with computers**.

There are lots of programming languages such as Java, C, C++, Python, PHP, Perl, Ruby etc.

The programming language you will be learning is Python. Python is a high-level interpreted language that is dynamically typed.

Note:

1. High-level language
2. Interpreted language
3. Dynamically typed

High-level language

As you might infer from the name high-level language, there are also low-level languages. Loosely speaking, computers can only execute programs written in low-level languages. Thus, programs written in a high-level language have to be translated into something more suitable before they can run.

Example of low-level languages are assembly language and machine language.

Example of high-level languages are JavaScript, Python, C++ etc.

Interpreted Language

An interpreter is a computer program that directly executes instructions written in a programming or scripting language, without requiring them previously to have been compiled into a machine language program.

The engine that translates and runs Python is called the **Python Interpreter**.

Interpreter Vs Compiler

1. Translates program one statement at a time.
2. Considering it scans code one line at a time, errors are shown line by line.
3. Due to interpreters reading code line by line, the execution time is slower.
4. It does not convert source code into object code instead it scans it line by line.
5. It requires source code for later execution.

Python, Ruby, Perl, SNOBOL, MATLAB, etc.

1. Compiler scans the whole program in one go.
2. As it scans the code in one go, the errors (if any) are shown at the end together.
3. Main advantage of compilers is its execution time.
4. It converts the source code into object code.
5. It does not require source code for later execution
6. E.g. C, C++, C#, Java etc.

Dynamically Typed

There are generally two language types i.e. static languages and dynamic languages.

In statically typed programming languages, type checking occurs at compile time. This means that before source code is compiled, the type associated with each and every single variable must be known. Example of statically typed languages are Java, C, C++ etc.

```
int varA = 6
```

```
String varB = "Hello World!"
```

A language is said to be dynamically typed if a variable is checked during run-time. This means that variables are checked against types only when the program is executing. Example of dynamically typed languages are Python, JavaScript, Ruby etc.

```
varC = "Welcome to Loctech!"
```



Advantages of Python Language

- Easy to Read, Learn and write
- Improved Productivity
- Interpreted Language
- Dynamically Typed
- Free and Open-Source
- Vast Libraries Support

Portability



Disadvantages of Python

- Slow Speed
- Not Memory Efficient
- Weak in Mobile Computing

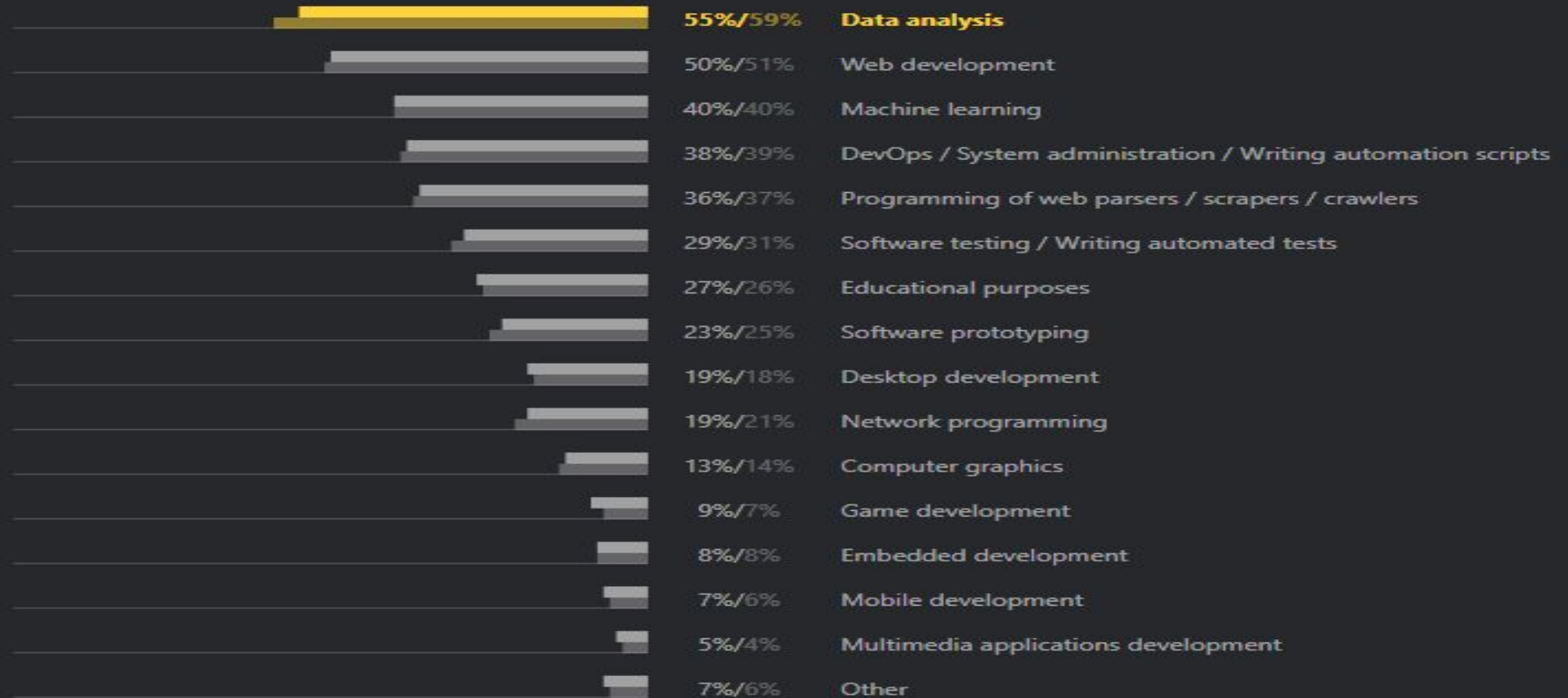
What do you use Python for?

> 100%

Main Secondary Combined

● 2020

● 2019



Values and Data Types

- A value is one of the fundamental things — like a letter or a number — that a program manipulates. The values we have seen so far are "Welcome to Loctech!" and 6.
- These values are classified into different classes, or data types: 6 is an integer, and "Welcome to Loctech" is a string, so-called because it contains a string of letters.

Data types in python

1. Strings `"Hello, World!", "23", 'Lion'`
2. Integers `98, 3426, 3`
3. Booleans `True or False`
4. Float `5.98`
5. Dictionary `{"name": "Uzoma", "company": "Loctech"}`
6. Tuple `(12, "Paris Hilton", True)`
7. List `[12, "Paris Hilton", True]`

Variable, Variable name and Keywords

- **Variable:** A name that refers to a value.
- **Variable name:** A name given to a variable. Variable names in Python consist of a sequence of letters (a..z, A..Z, and _) and digits (0..9) that begins with a letter. In best programming practice, variable names should be chosen so that they describe their use in the program, making the program self documenting.
- **Keyword:** A reserved word that is used by the compiler to parse program; you cannot use keywords like `if`, `def`, and `while` as variable names.

Variables

One of the most powerful features of a programming language is the ability to manipulate variables. A variable is a name that refers to a value.

The **assignment statement** gives a value to a variable:

```
>>> message = "What's up, Doc?"  
>>> n = 17  
>>> pi = 3.14159
```

This example makes three assignments. The first assigns the string value "What's up, Doc?" to a variable named `message`. The second gives the integer 17 to `n`, and the third assigns the floating-point number 3.14159 to a variable called `pi`.

The assignment token, =, should not be confused with equals, which uses the token ==. The assignment statement binds a name, on the left-hand side of the operator, to a value, on the right-hand side.

If you ask the interpreter to evaluate a variable, it will produce the value that is currently linked to the variable:

```
>>> message
'What's up, Doc?'
>>> n
17
>>> pi
3.14159
```


Rules for assigning variables

1. A variable name must start with a letter or the underscore character.
2. A variable name cannot start with a number.
3. A variable name can only contain alpha-numeric characters and underscores (A-z, 0-9, and _)
4. Variable names are case-sensitive (age, Age and AGE are three different variables)

Below

and	as	assert	break	class	continue
def	del	elif	else	except	exec
finally	for	from	global	if	import
in	is	lambda	nonlocal	not	or
pass	raise	return	try	while	with
yield	True	False	None		