Instant community
www.instant-ss.com

[i]NSTANT

AI Diploma R27

Session 1

AI What and why?

Prepared by/ Elzahraa Alaa Tag Eldein

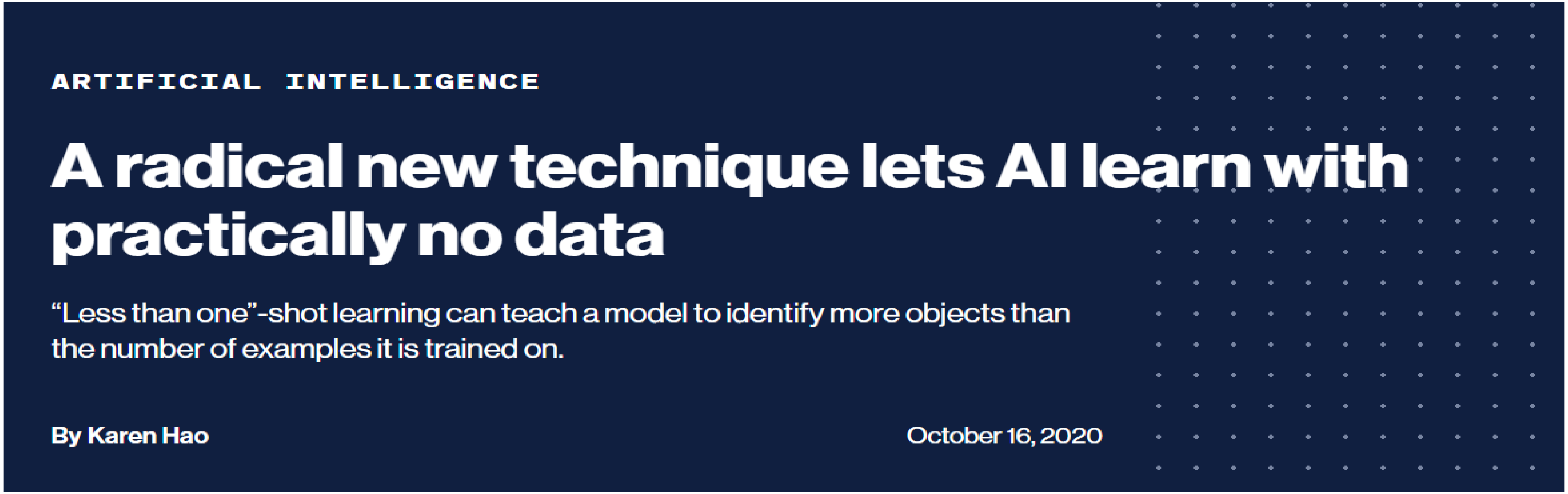
March 13, 2023

Task 1

How to use AI without Data?

[i]NSTANT

AI Diploma R27



ARTIFICIAL INTELLIGENCE

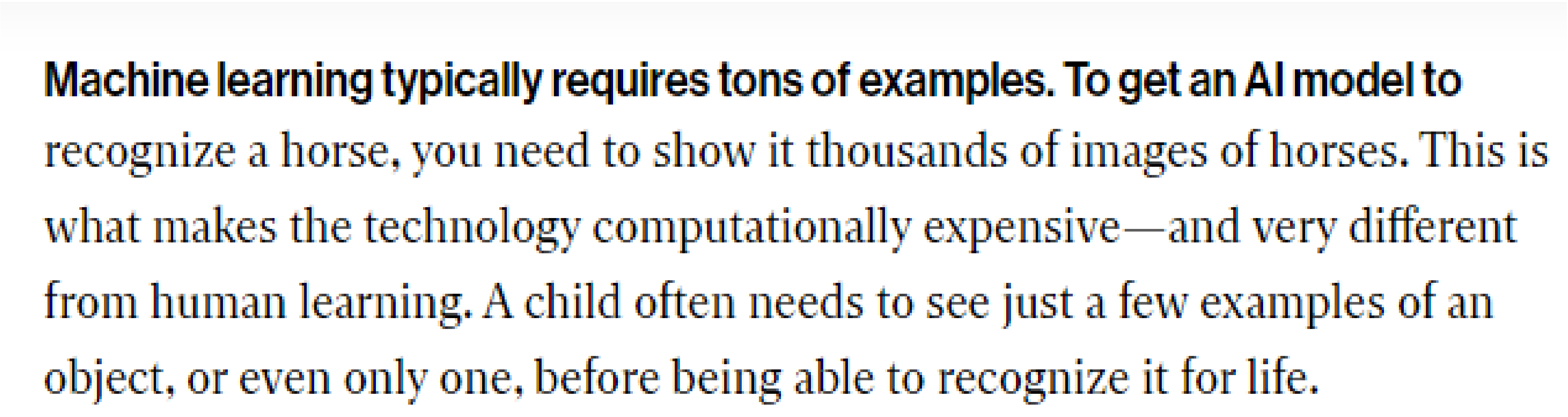
A radical new technique lets AI learn with practically no data

“Less than one”-shot learning can teach a model to identify more objects than the number of examples it is trained on.

By Karen Hao

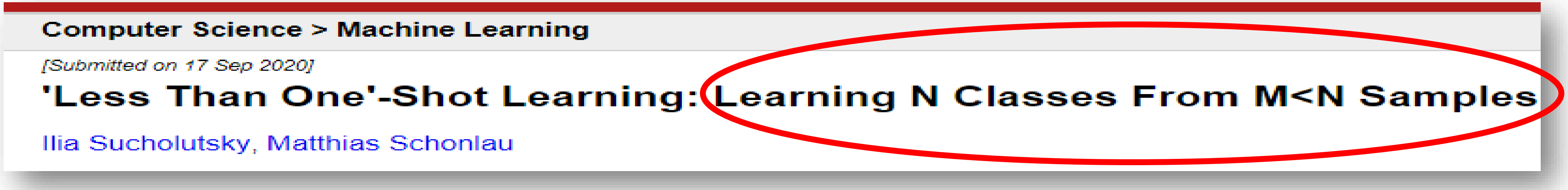
October 16, 2020

https://www.technologyreview.com/-ynti-htiw-gninrael-enihcam-ia/٢٠٢٠/١٠/١٦/١٠١٠٥٦٦
laicos_rt=muidem_mtu&tnemegagne.diapnu.rotisiv_etis=ngiapmac_mtu&rettiwT=ecruos_mtu؟/atad



Machine learning typically requires tons of examples. To get an AI model to recognize a horse, you need to show it thousands of images of horses. This is what makes the technology computationally expensive—and very different from human learning. A child often needs to see just a few examples of an object, or even only one, before being able to recognize it for life.

In fact, children sometimes don't need *any* examples to identify something. Shown photos of a horse and a rhino, and told a unicorn is something in between, they can recognize the mythical creature in a picture book the first time they see it.

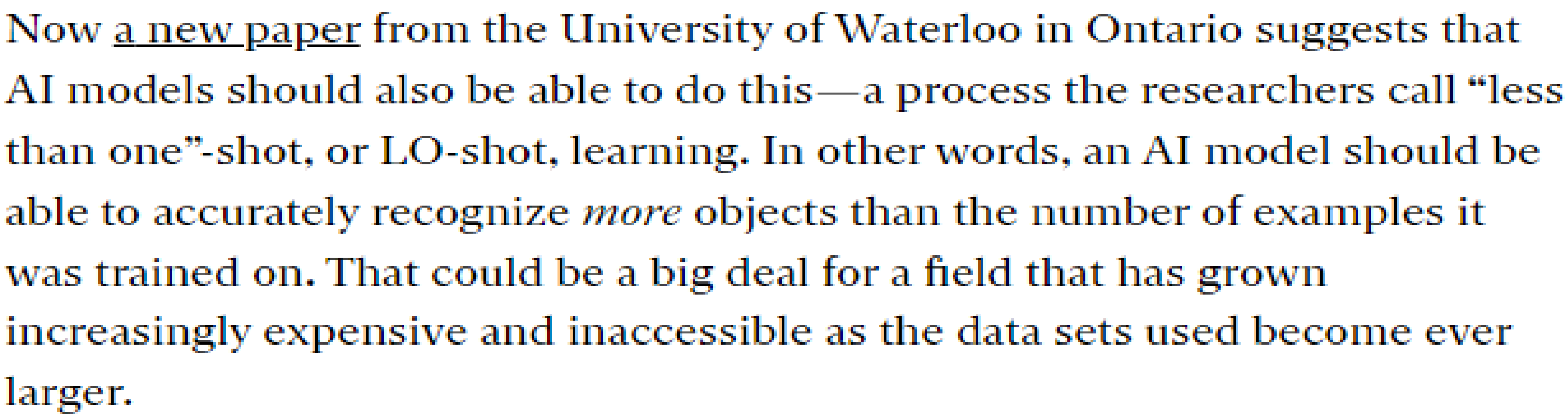


Computer Science > Machine Learning

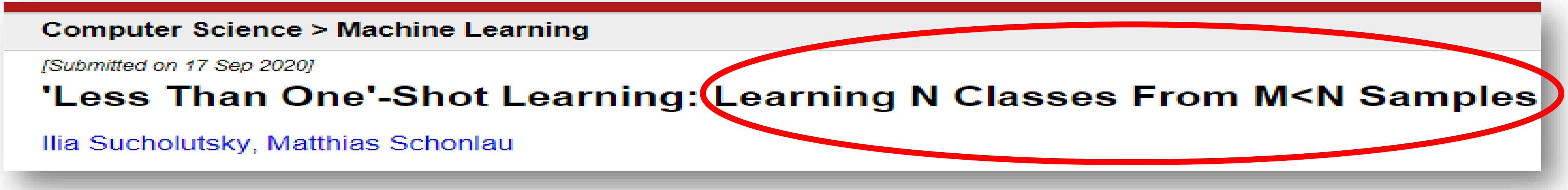
[Submitted on 17 Sep 2020]

'Less Than One'-Shot Learning: Learning N Classes From M<N Samples

Ilia Sucholutsky, Matthias Schonlau



Now a new paper from the University of Waterloo in Ontario suggests that AI models should also be able to do this—a process the researchers call "less than one"-shot, or LO-shot, learning. In other words, an AI model should be able to accurately recognize *more* objects than the number of examples it was trained on. That could be a big deal for a field that has grown increasingly expensive and inaccessible as the data sets used become ever larger.

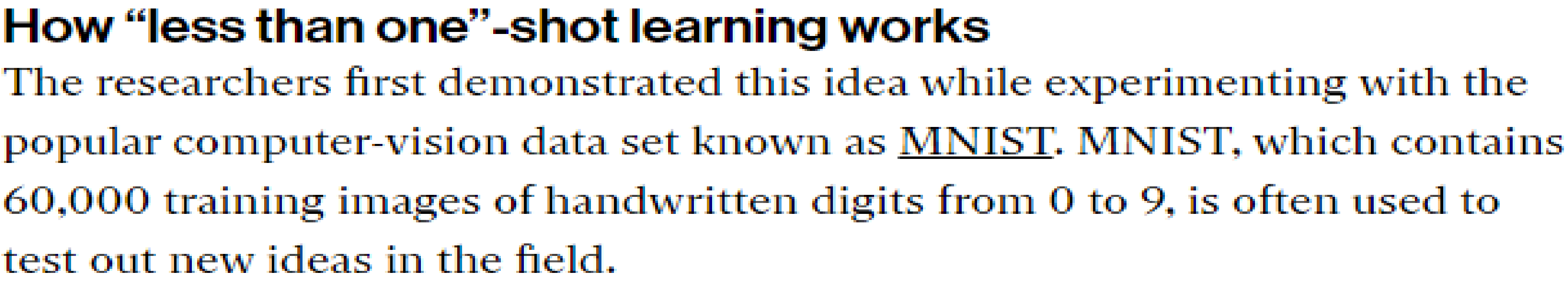


Computer Science > Machine Learning

[Submitted on 17 Sep 2020]

'Less Than One'-Shot Learning: Learning N Classes From M<N Samples

Ilia Sucholutsky, Matthias Schonlau



How “less than one”-shot learning works

The researchers first demonstrated this idea while experimenting with the popular computer-vision data set known as MNIST. MNIST, which contains 60,000 training images of handwritten digits from 0 to 9, is often used to test out new ideas in the field.

Task 2

List AI Companies and startups in Egypt

10 Top Artificial Intelligence Startups and Companies in Egypt

https://www.f6s.com/companies/artificial-intelligence/egypt/co

21 Most Innovative Egypt Based Artificial Intelligence Companies

https://futurology.life/21-most-innovative-egypt-based-artificial-intelligence-companies/

104 AI/ML Driven Companies In Egypt

https://github.com/harryadel/AI-ML-Driven-Companies-In-Egypt/blob/master/README.md#aiml-driven-companies-in-egypt

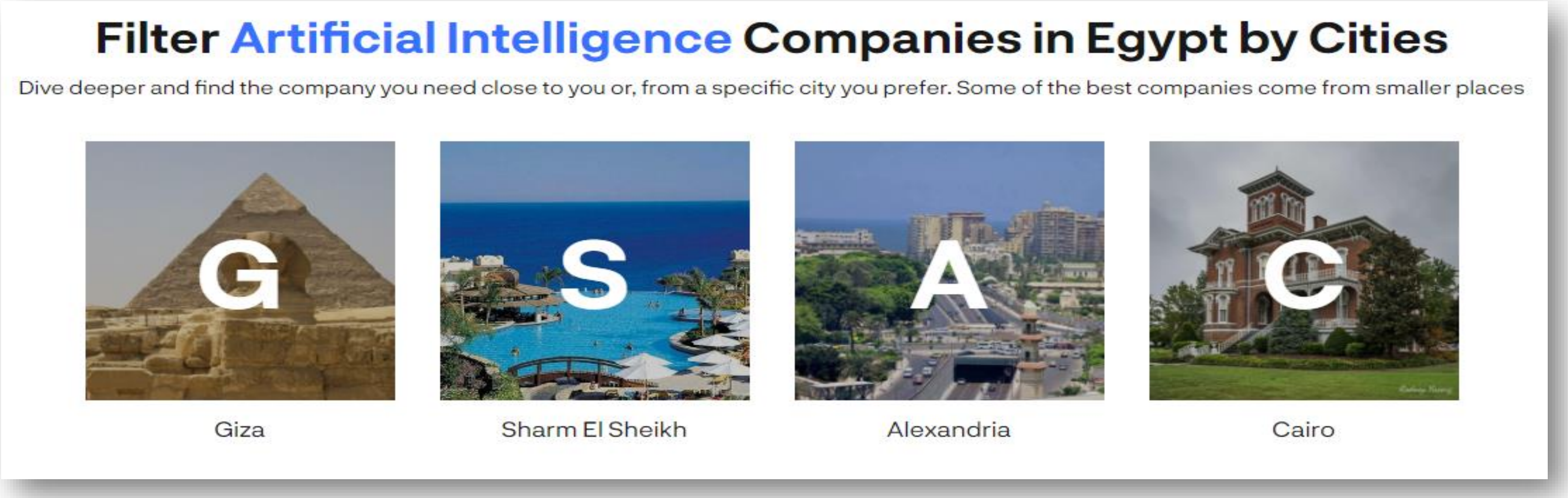
18 Most Innovative Machine Learning Startups & Companies (Egypt)

https://datamagazine.co.uk/-enihcam-evtiavonni-tsom-١٨
/tpyge-seinapmoc-sputrats-gninrael

[i]NSTANT

AI Diploma R27

Tech Behemoths



Filter Artificial Intelligence Companies in Egypt by Cities

Dive deeper and find the company you need close to you or, from a specific city you prefer. Some of the best companies come from smaller places

Giza

Sharm El Sheikh

Alexandria

Cairo

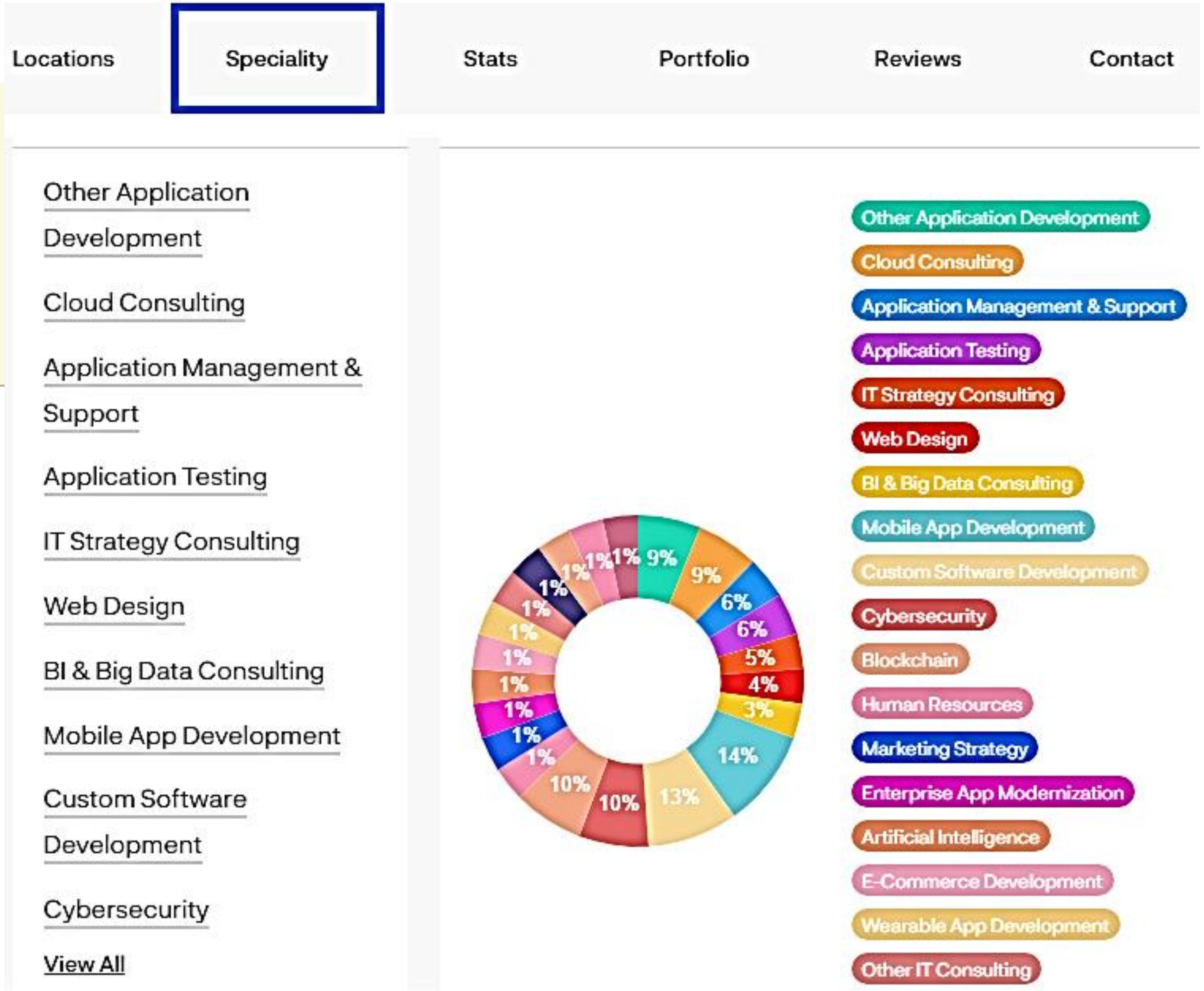
https://techbehemoths.com/companies/artificial-intelligence/egypt systems

Systems Limited

Giza, Egypt

Founding Year 1977

Team Size 1,000-9,999



Locations | Speciality | Stats | Portfolio | Reviews | Contact

Other Application Development

Cloud Consulting

Application Management & Support

Application Testing

IT Strategy Consulting

Web Design

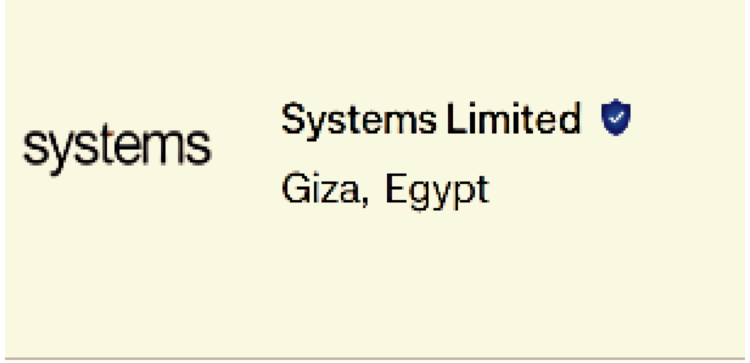
BI & Big Data Consulting

Mobile App Development

Custom Software Development

Cybersecurity

View All



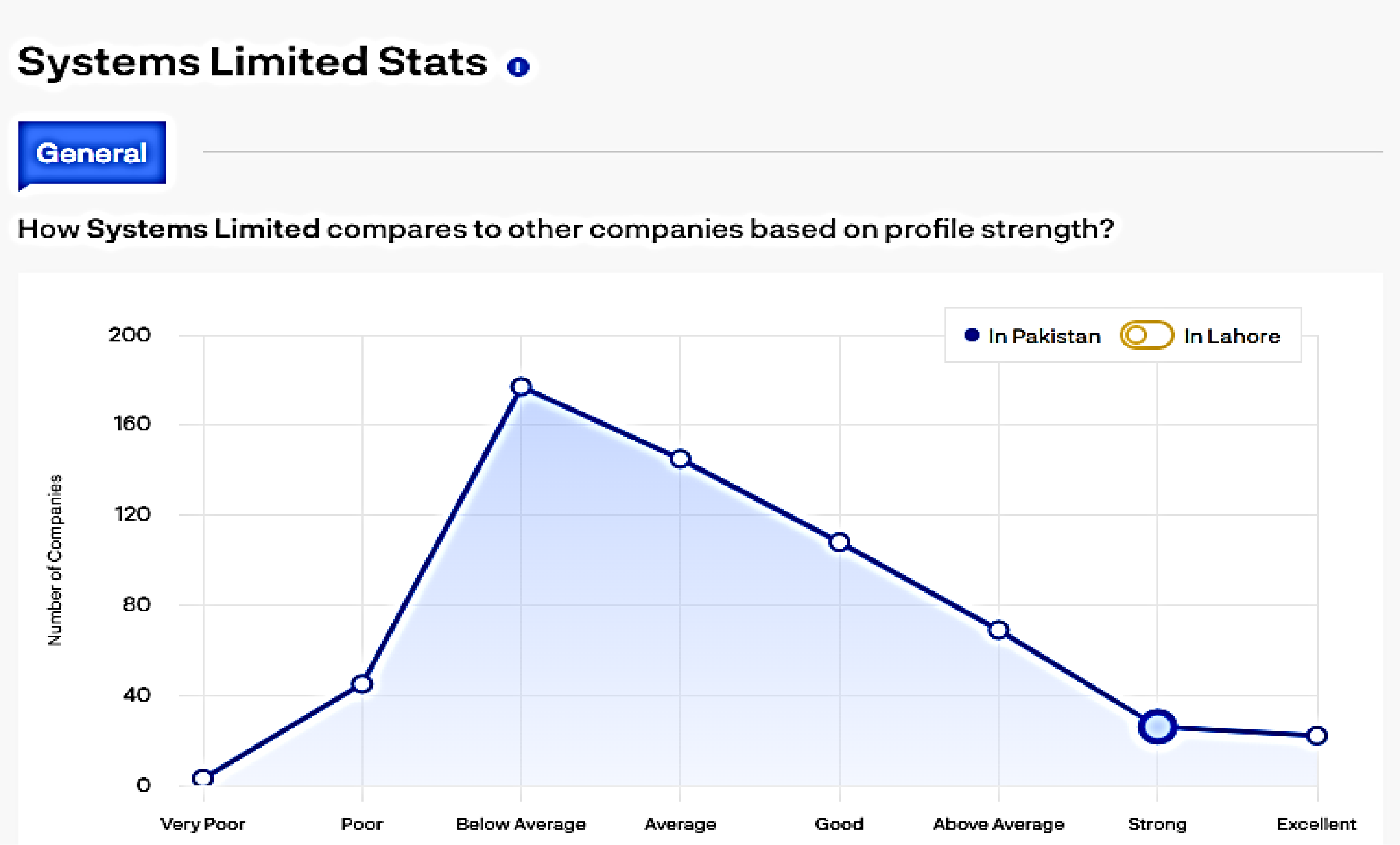
systems

Systems Limited

Giza, Egypt

Founding Year 1977

Team Size 1,000-9,999



Systems Limited Stats

General

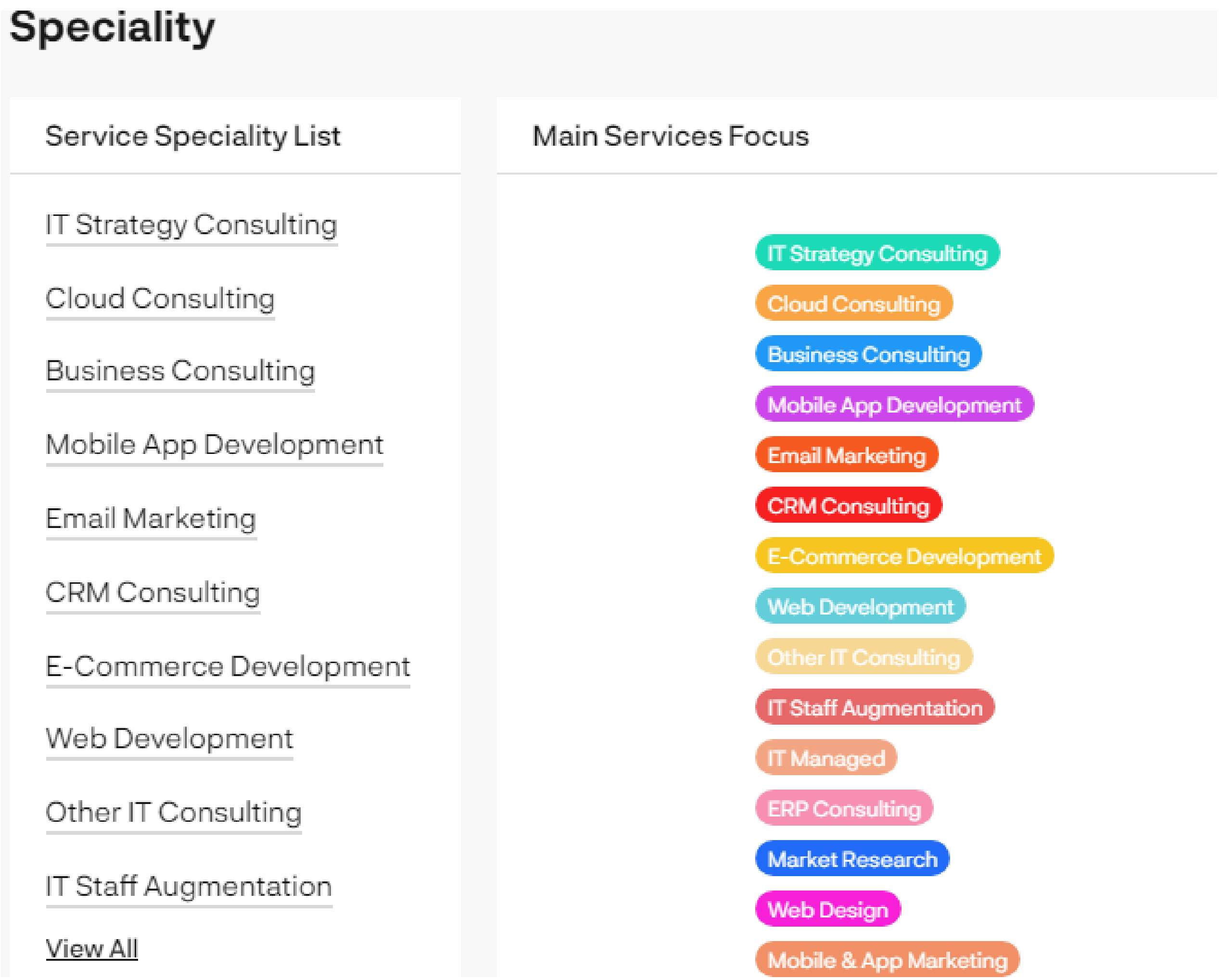
How **Systems Limited** compares to other companies based on profile strength?

Mars Consultation

Sharm El Sheikh, Egypt

Founding Year 2021

Team Size Freelancer



Speciality

Service Speciality List

- IT Strategy Consulting
- Cloud Consulting
- Business Consulting
- Mobile App Development
- Email Marketing
- CRM Consulting
- E-Commerce Development
- Web Development
- Other IT Consulting
- IT Staff Augmentation

View All

Main Services Focus

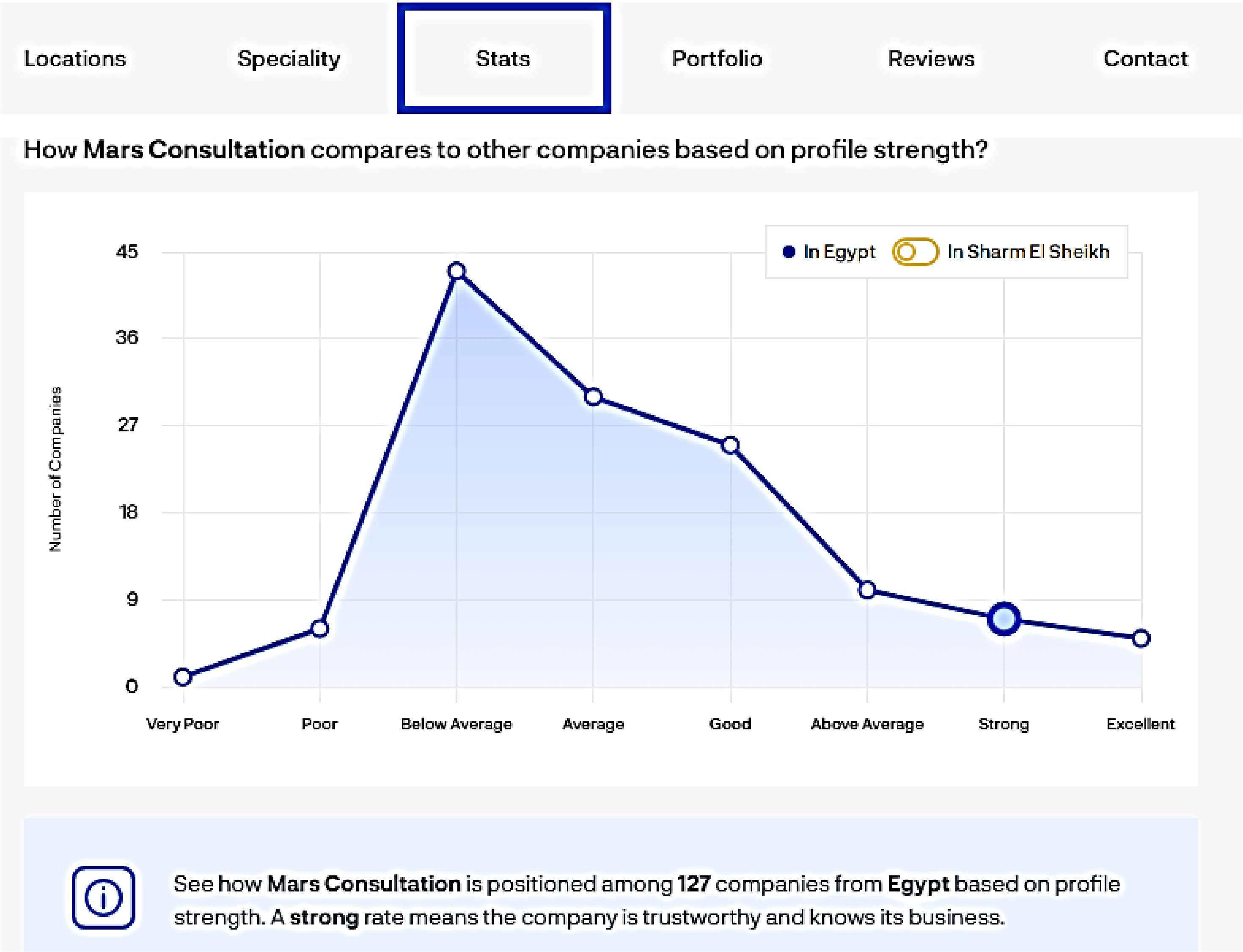
- IT Strategy Consulting
- Cloud Consulting
- Business Consulting
- Mobile App Development
- Email Marketing
- CRM Consulting
- E-Commerce Development
- Web Development
- Other IT Consulting
- IT Staff Augmentation
- IT Managed
- ERP Consulting
- Market Research
- Web Design
- Mobile & App Marketing

Mars Consultation

Sharm El Sheikh, Egypt

Founding Year 2021

Team Size Freelancer



How **Mars Consultation** compares to other companies based on profile strength?

See how **Mars Consultation** is positioned among **127** companies from **Egypt** based on profile strength. A **strong** rate means the company is trustworthy and knows its business.

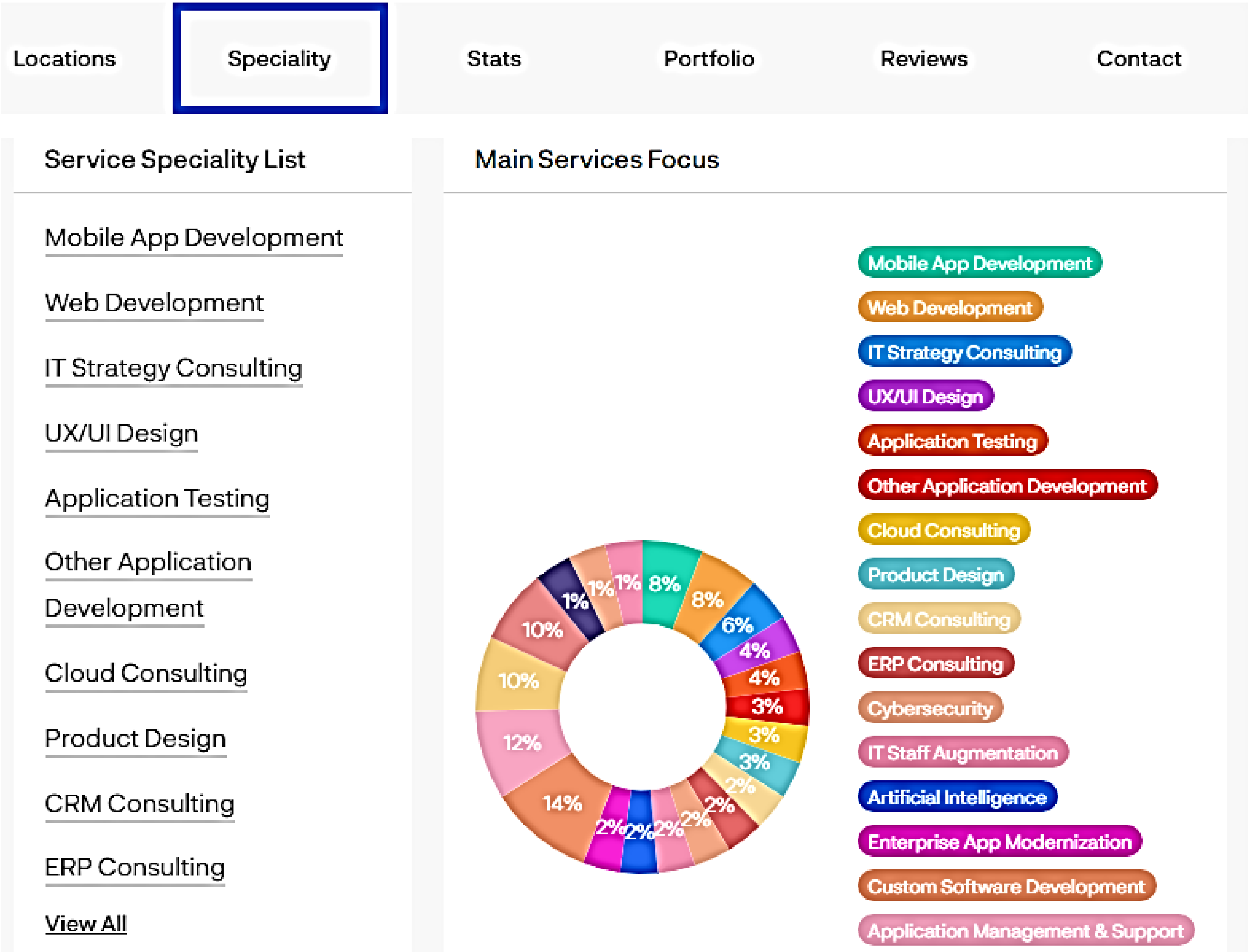
FLAIRSTECH
INNOVATION | TALENT | EXCELLENCE

FlairsTech

Cairo, Egypt

Founding Year 2018

Team Size 1,000-9,999



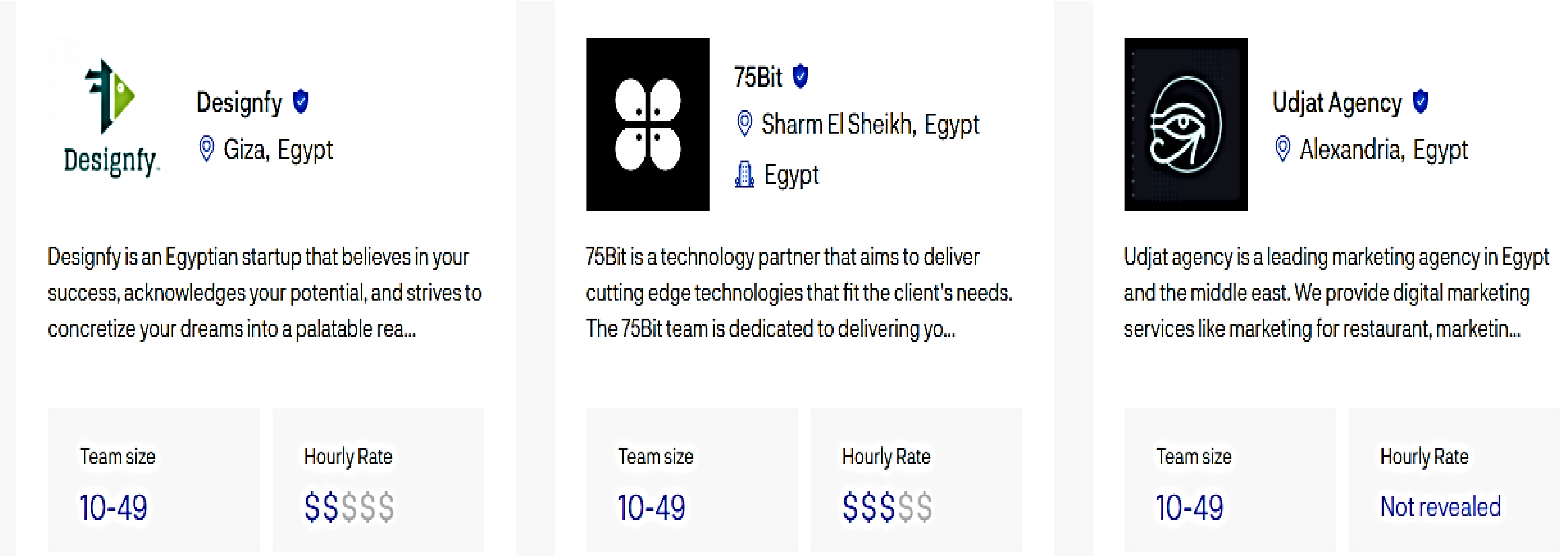
Service Speciality List

- Mobile App Development
- Web Development
- IT Strategy Consulting
- UX/UI Design
- Application Testing
- Other Application Development
- Cloud Consulting
- Product Design
- CRM Consulting
- ERP Consulting

View All

Main Services Focus

Tech Behemoths



Designfy

Giza, Egypt

Designfy is an Egyptian startup that believes in your success, acknowledges your potential, and strives to concretize your dreams into a palatable rea...

Team size	Hourly Rate
10-49	$$$$$

75Bit

Sharm El Sheikh, Egypt

Egypt

75Bit is a technology partner that aims to deliver cutting edge technologies that fit the client's needs. The 75Bit team is dedicated to delivering yo...

Team size	Hourly Rate
10-49	$$$$$

Udjat Agency

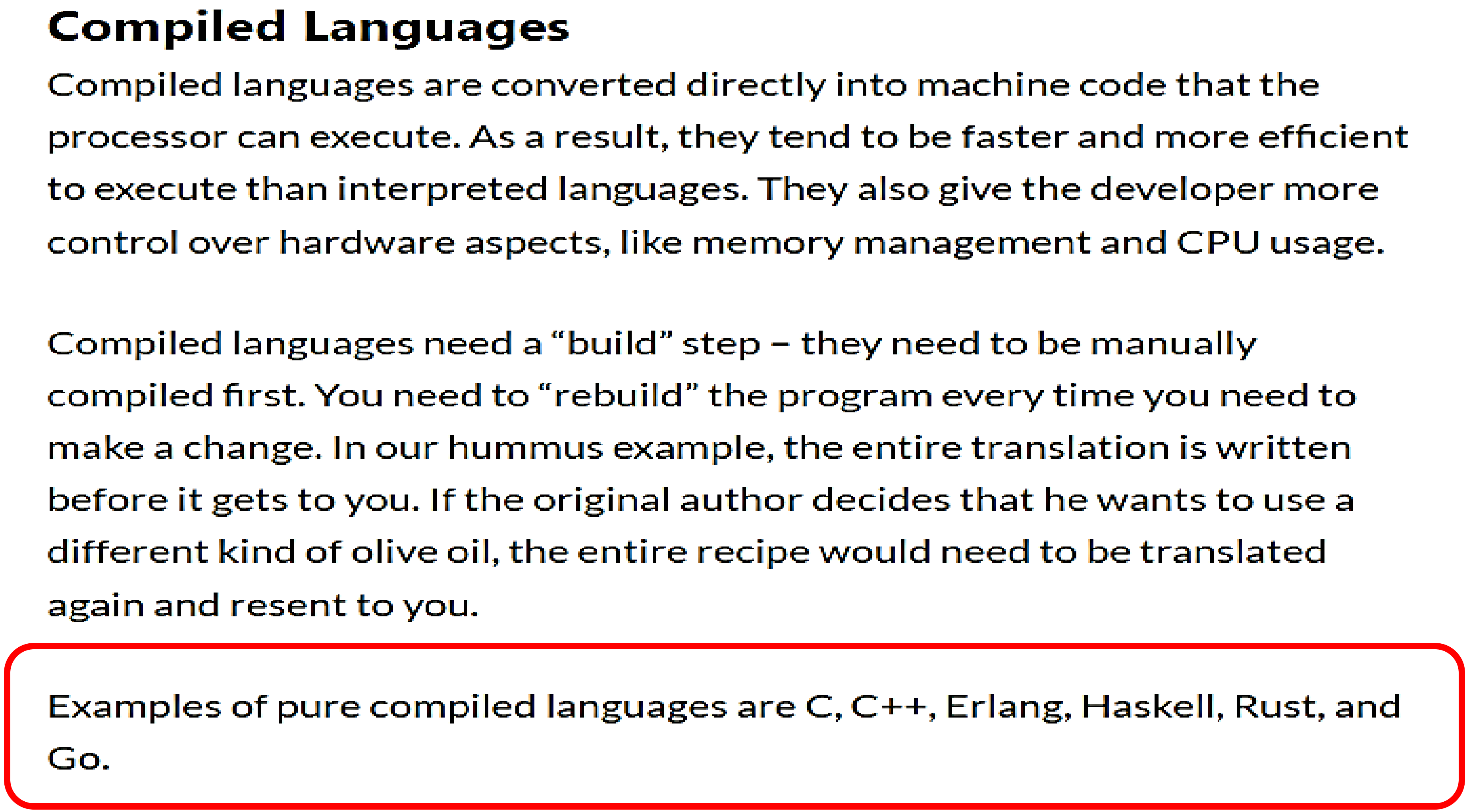
Alexandria, Egypt

Udjat agency is a leading marketing agency in Egypt and the middle east. We provide digital marketing services like marketing for restaurant, marketin...

Team size	Hourly Rate
10-49	Not revealed

Task 3

Complied vs interpreted programming Languages



Compiled Languages

Compiled languages are converted directly into machine code that the processor can execute. As a result, they tend to be faster and more efficient to execute than interpreted languages. They also give the developer more control over hardware aspects, like memory management and CPU usage.

Compiled languages need a “build” step – they need to be manually compiled first. You need to “rebuild” the program every time you need to make a change. In our hummus example, the entire translation is written before it gets to you. If the original author decides that he wants to use a different kind of olive oil, the entire recipe would need to be translated again and resent to you.

Examples of pure compiled languages are C, C++, Erlang, Haskell, Rust, and Go.

freeCodeCamp

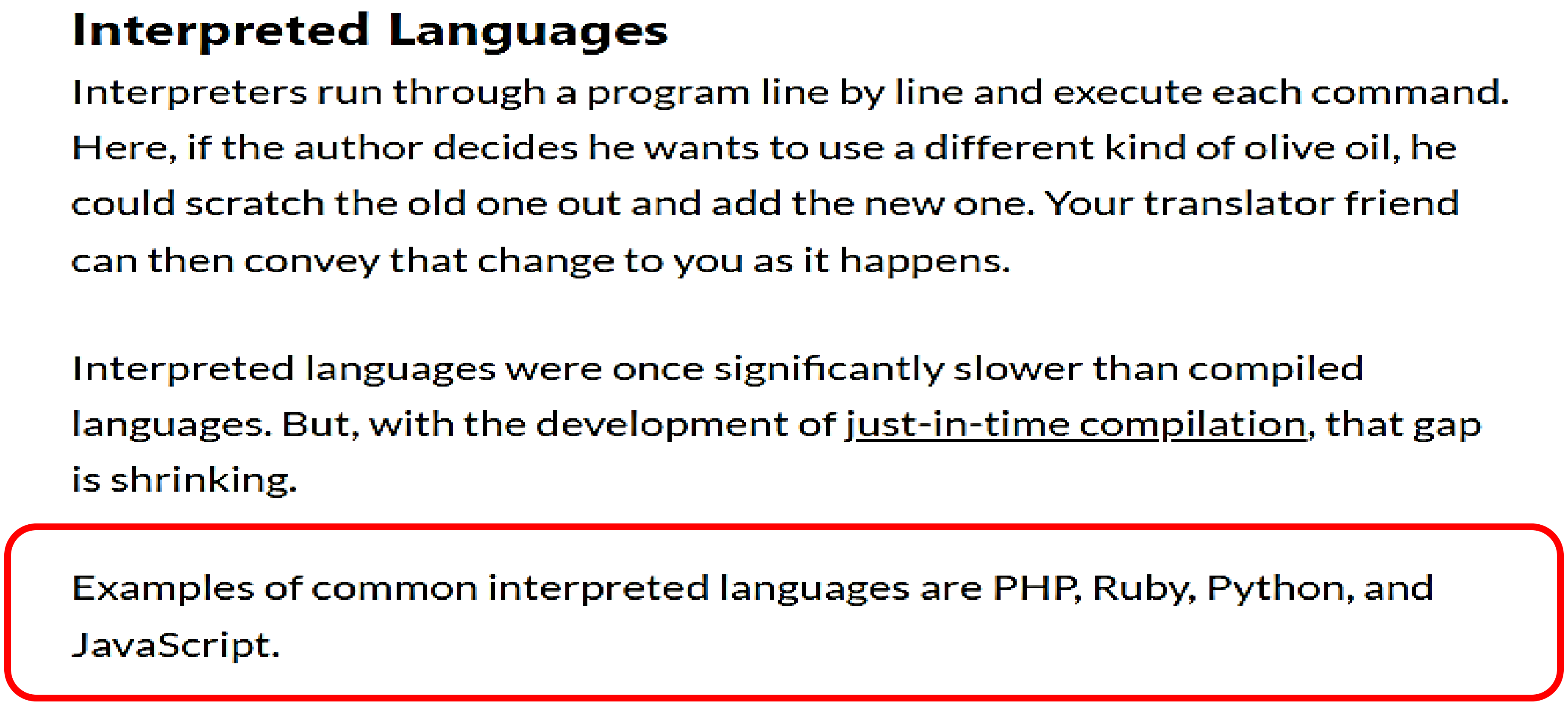
Advantages of compiled languages

Programs that are compiled into native machine code tend to be faster than interpreted code. This is because the process of translating code at run time adds to the overhead, and can cause the program to be slower overall.

Disadvantages of compiled languages

The most notable disadvantages are:

- Additional time needed to complete the entire compilation step before testing
- Platform dependence of the generated binary code



Interpreted Languages

Interpreters run through a program line by line and execute each command. Here, if the author decides he wants to use a different kind of olive oil, he could scratch the old one out and add the new one. Your translator friend can then convey that change to you as it happens.

Interpreted languages were once significantly slower than compiled languages. But, with the development of just-in-time compilation, that gap is shrinking.

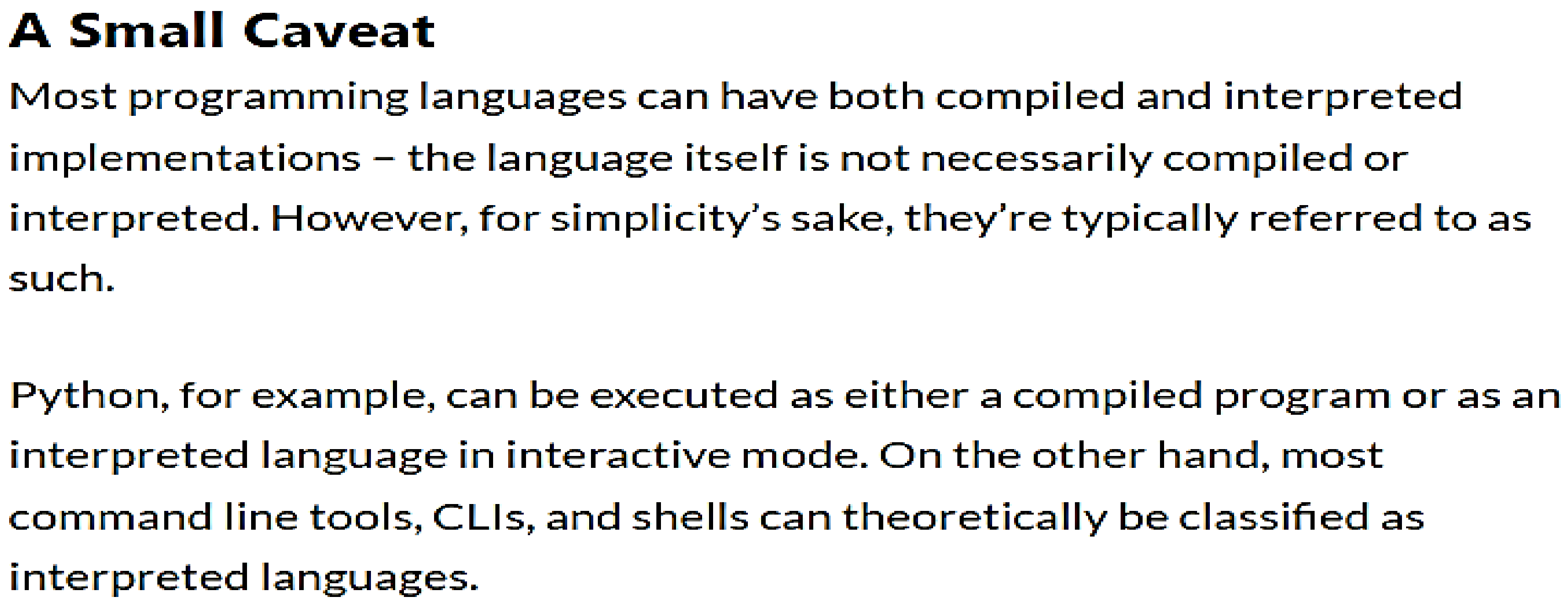
Examples of common interpreted languages are PHP, Ruby, Python, and JavaScript.

Advantages of interpreted languages

Interpreted languages tend to be more flexible, and often offer features like dynamic typing and smaller program size. Also, because interpreters execute the source program code themselves, the code itself is platform independent.

Disadvantages of interpreted languages

The most notable disadvantage is typical execution speed compared to compiled languages.



A Small Caveat

Most programming languages can have both compiled and interpreted implementations – the language itself is not necessarily compiled or interpreted. However, for simplicity's sake, they're typically referred to as such.

Python, for example, can be executed as either a compiled program or as an interpreted language in interactive mode. On the other hand, most command line tools, CLIs, and shells can theoretically be classified as interpreted languages.

Task 4

Open-source vs not open-source programming Languages

Top Open-Source Programming Languages

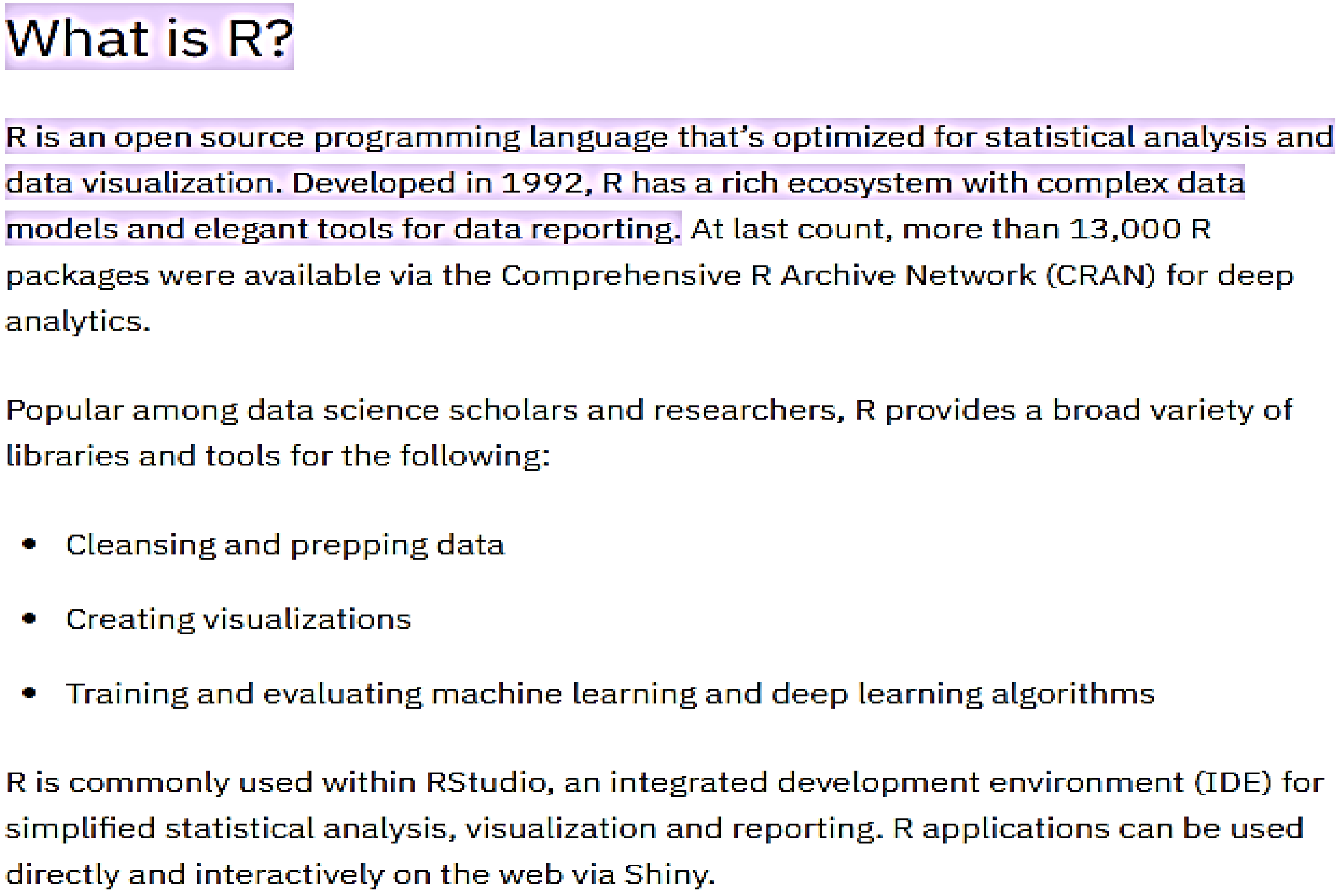
- Ruby
- Python
- R
- PHP
- Scala
- Kotlin

Not open-Programming Languages

- Java
- C
- C^{++}
- Swift
- Vyper
- Ada
- Delphi
- Visual Basic
- Microsoft’s .NET Framework
- Gravity
- Imba
- Morfa
- Objeck

Task 5

R ?



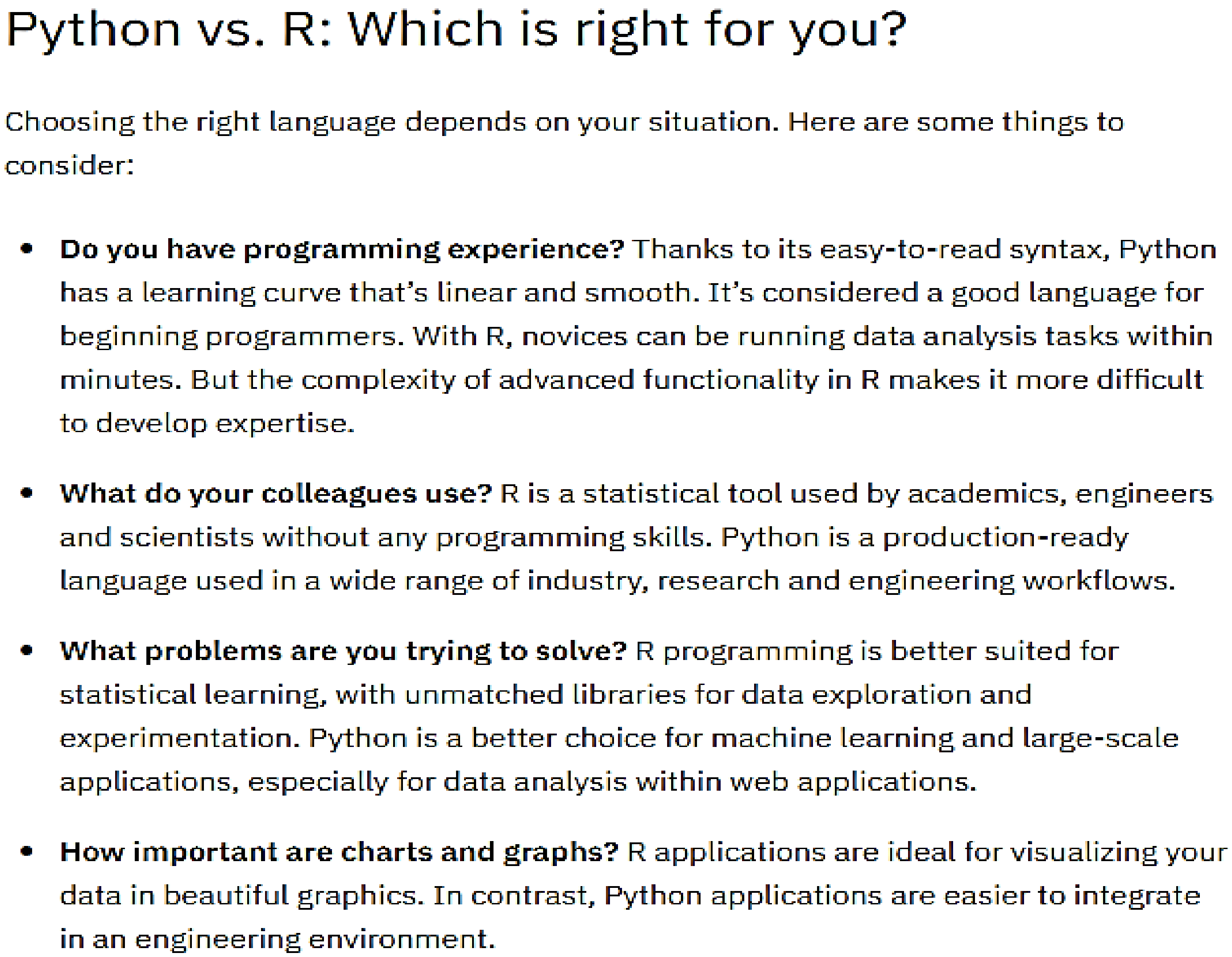
What is R?

R is an open source programming language that's optimized for statistical analysis and data visualization. Developed in 1992, R has a rich ecosystem with complex data models and elegant tools for data reporting. At last count, more than 13,000 R packages were available via the Comprehensive R Archive Network (CRAN) for deep analytics.

Popular among data science scholars and researchers, R provides a broad variety of libraries and tools for the following:

- Cleansing and prepping data
- Creating visualizations
- Training and evaluating machine learning and deep learning algorithms

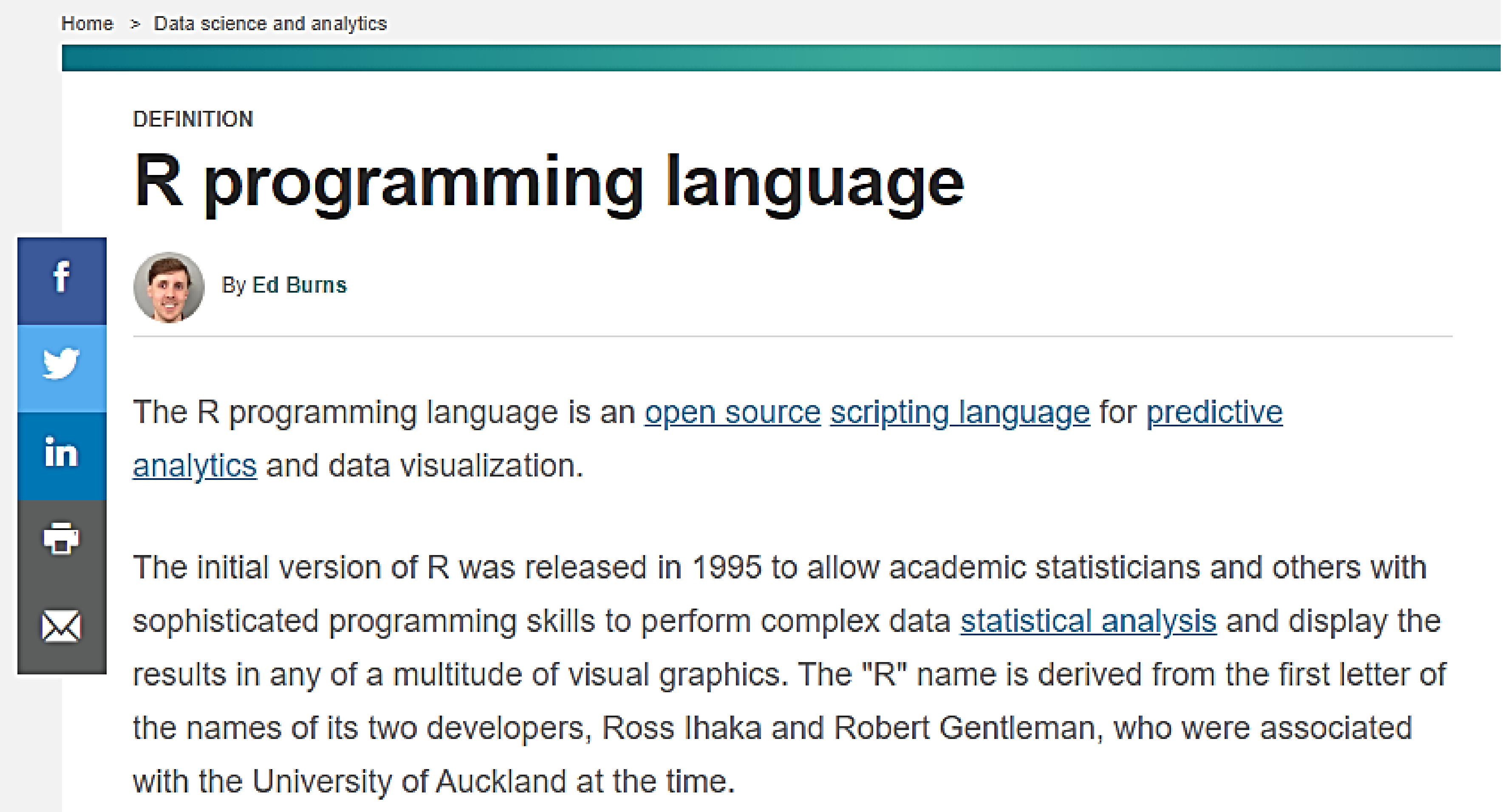
R is commonly used within RStudio, an integrated development environment (IDE) for simplified statistical analysis, visualization and reporting. R applications can be used directly and interactively on the web via Shiny.



Python vs. R: Which is right for you?

Choosing the right language depends on your situation. Here are some things to consider:

- **Do you have programming experience?** Thanks to its easy-to-read syntax, Python has a learning curve that's linear and smooth. It's considered a good language for beginning programmers. With R, novices can be running data analysis tasks within minutes. But the complexity of advanced functionality in R makes it more difficult to develop expertise.
- **What do your colleagues use?** R is a statistical tool used by academics, engineers and scientists without any programming skills. Python is a production-ready language used in a wide range of industry, research and engineering workflows.
- **What problems are you trying to solve?** R programming is better suited for statistical learning, with unmatched libraries for data exploration and experimentation. Python is a better choice for machine learning and large-scale applications, especially for data analysis within web applications.
- **How important are charts and graphs?** R applications are ideal for visualizing your data in beautiful graphics. In contrast, Python applications are easier to integrate in an engineering environment.



Home > Data science and analytics

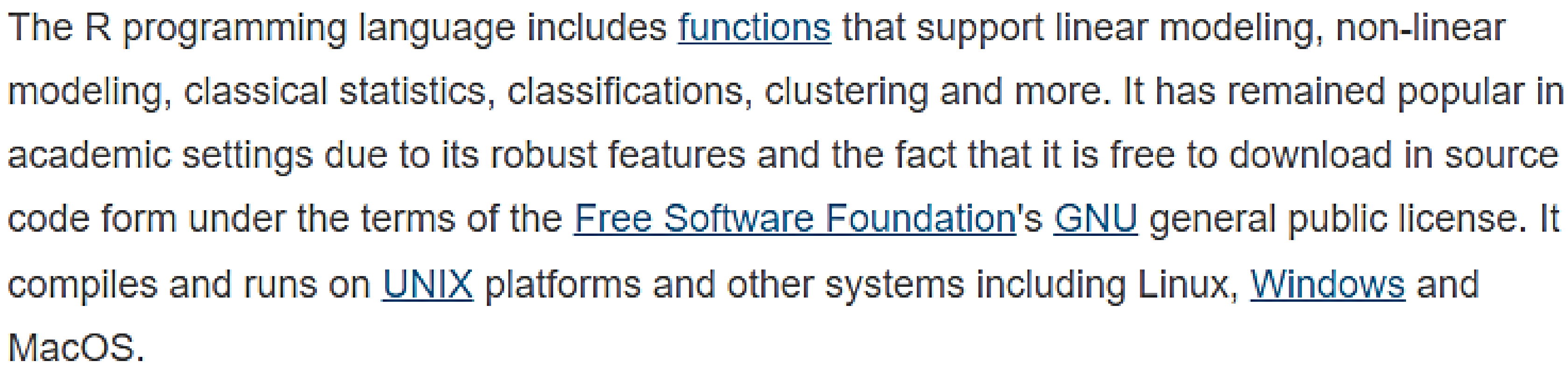
DEFINITION

R programming language

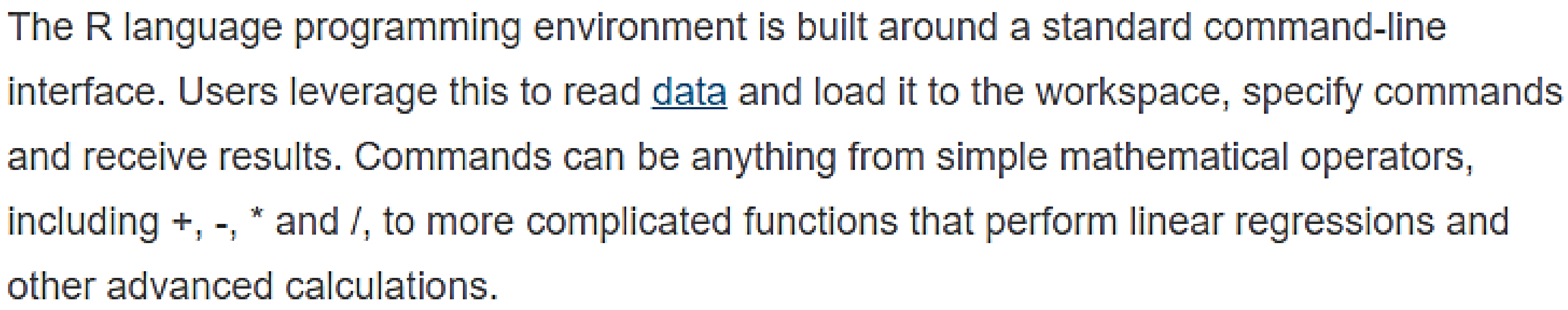
By Ed Burns

The R programming language is an open source scripting language for predictive analytics and data visualization.

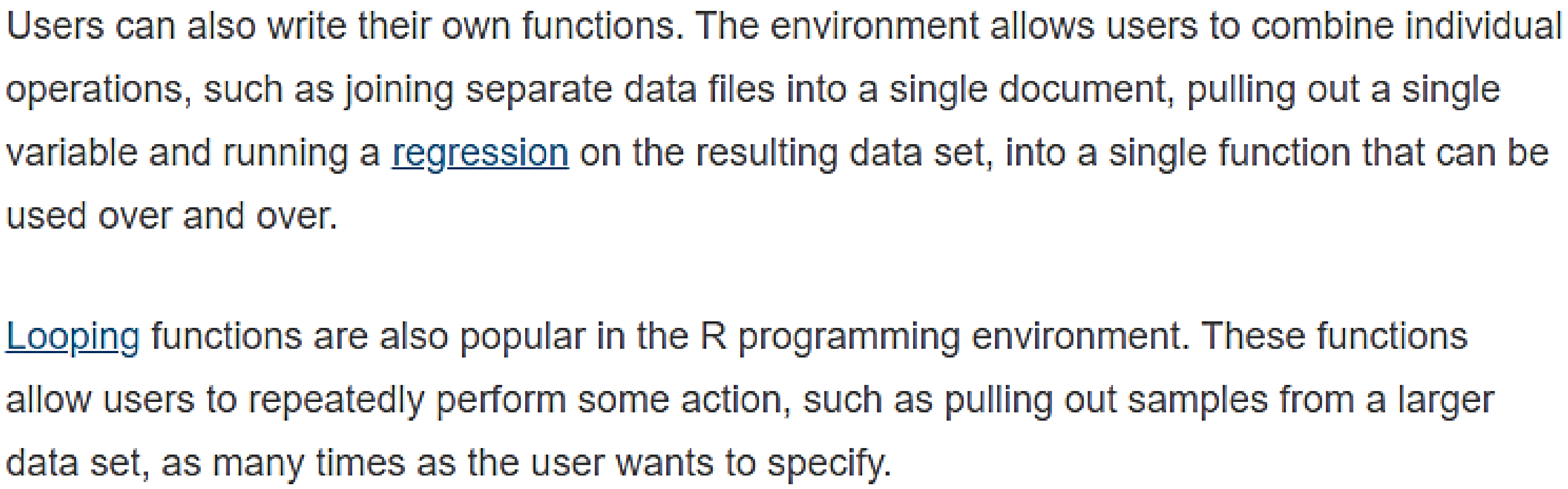
The initial version of R was released in 1995 to allow academic statisticians and others with sophisticated programming skills to perform complex data statistical analysis and display the results in any of a multitude of visual graphics. The "R" name is derived from the first letter of the names of its two developers, Ross Ihaka and Robert Gentleman, who were associated with the University of Auckland at the time.



The R programming language includes functions that support linear modeling, non-linear modeling, classical statistics, classifications, clustering and more. It has remained popular in academic settings due to its robust features and the fact that it is free to download in source code form under the terms of the Free Software Foundation's GNU general public license. It compiles and runs on UNIX platforms and other systems including Linux, Windows and MacOS.



The R language programming environment is built around a standard command-line interface. Users leverage this to read data and load it to the workspace, specify commands and receive results. Commands can be anything from simple mathematical operators, including +, -, * and /, to more complicated functions that perform linear regressions and other advanced calculations.

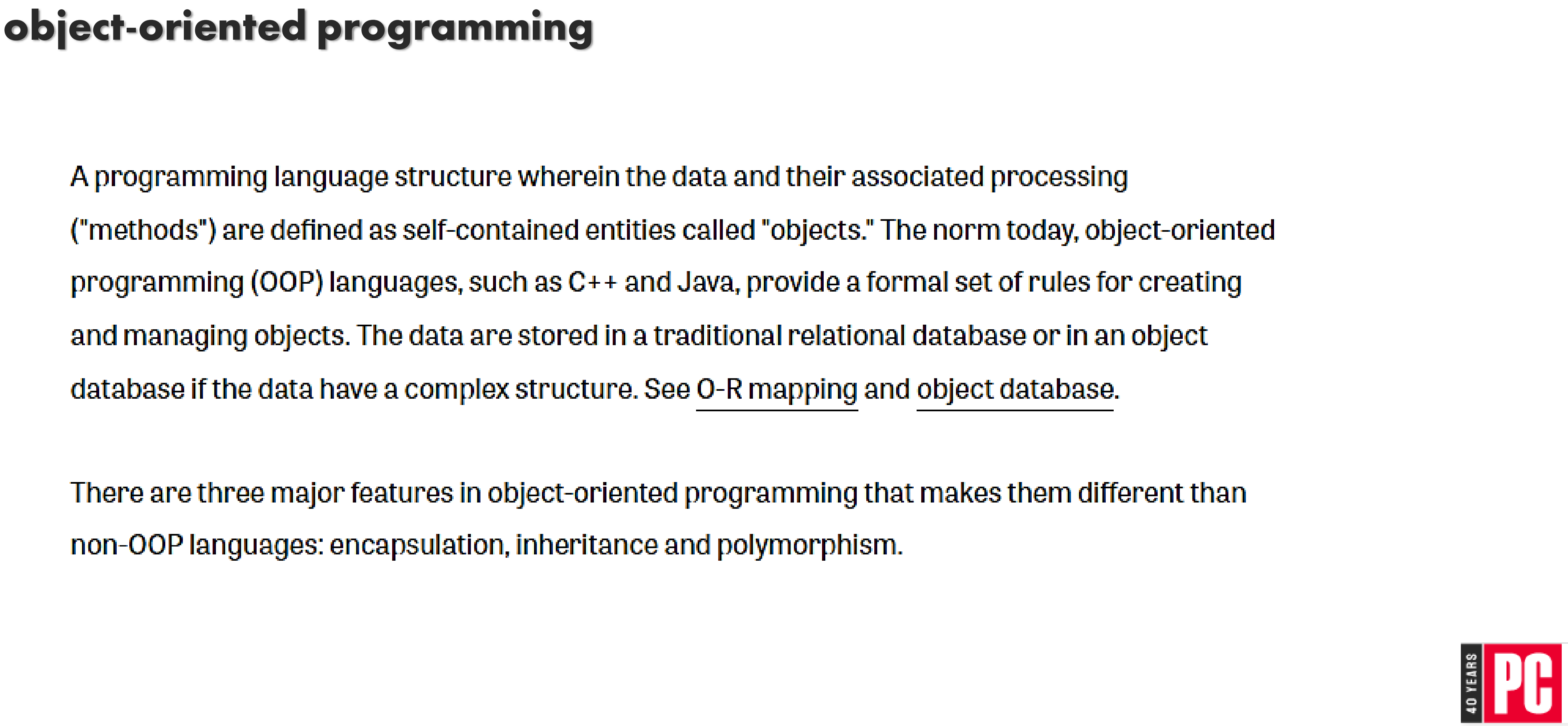


Users can also write their own functions. The environment allows users to combine individual operations, such as joining separate data files into a single document, pulling out a single variable and running a regression on the resulting data set, into a single function that can be used over and over.

Looping functions are also popular in the R programming environment. These functions allow users to repeatedly perform some action, such as pulling out samples from a larger data set, as many times as the user wants to specify.

Task 6

Not object-oriented programming Languages

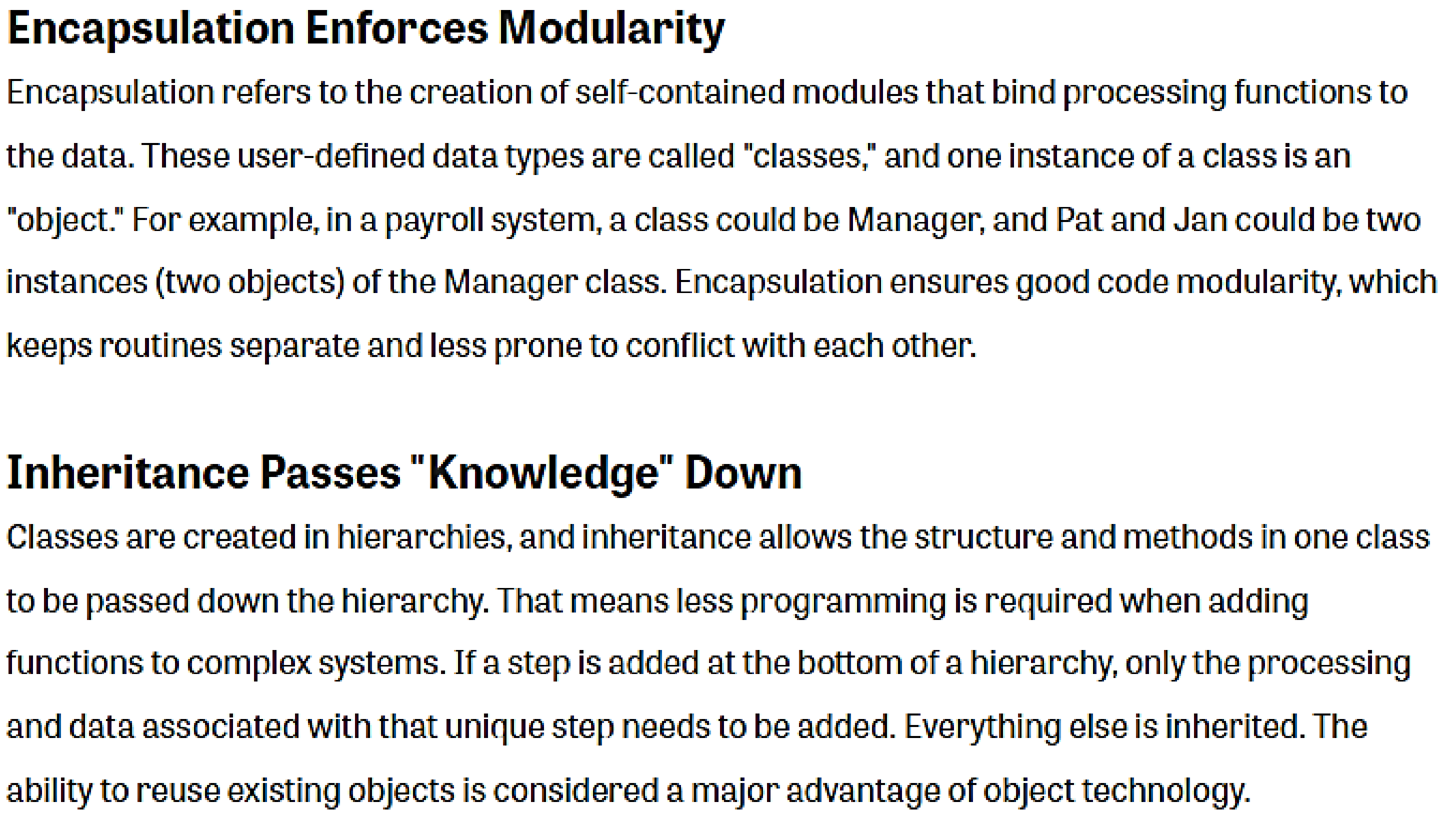


object-oriented programming

A programming language structure wherein the data and their associated processing ("methods") are defined as self-contained entities called "objects." The norm today, object-oriented programming (OOP) languages, such as C++ and Java, provide a formal set of rules for creating and managing objects. The data are stored in a traditional relational database or in an object database if the data have a complex structure. See O-R mapping and object database.

There are three major features in object-oriented programming that makes them different than non-OOP languages: encapsulation, inheritance and polymorphism.

1 object-oriented programming language

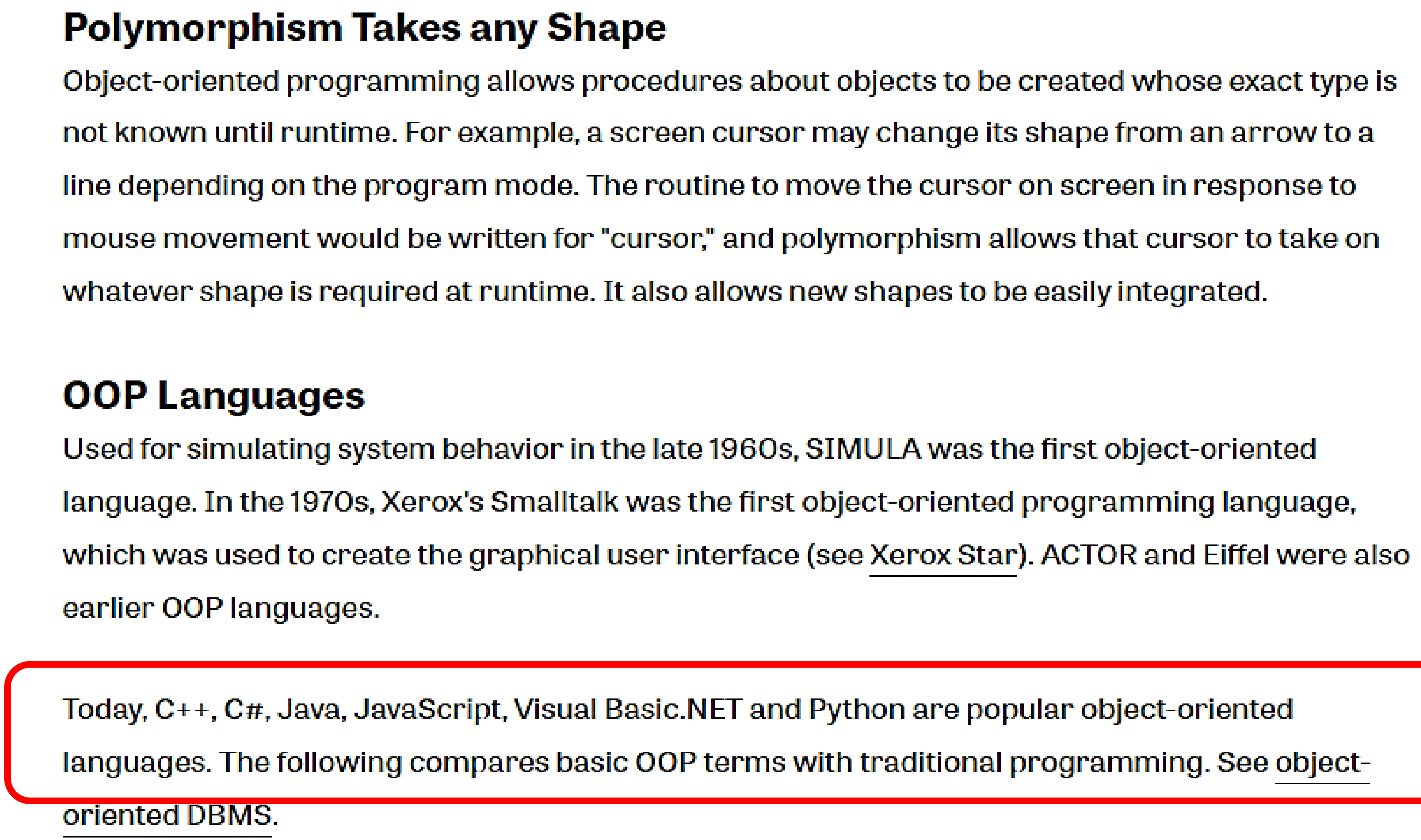


Encapsulation Enforces Modularity

Encapsulation refers to the creation of self-contained modules that bind processing functions to the data. These user-defined data types are called "classes," and one instance of a class is an "object." For example, in a payroll system, a class could be Manager, and Pat and Jan could be two instances (two objects) of the Manager class. Encapsulation ensures good code modularity, which keeps routines separate and less prone to conflict with each other.

Inheritance Passes "Knowledge" Down

Classes are created in hierarchies, and inheritance allows the structure and methods in one class to be passed down the hierarchy. That means less programming is required when adding functions to complex systems. If a step is added at the bottom of a hierarchy, only the processing and data associated with that unique step needs to be added. Everything else is inherited. The ability to reuse existing objects is considered a major advantage of object technology.



Polymorphism Takes any Shape

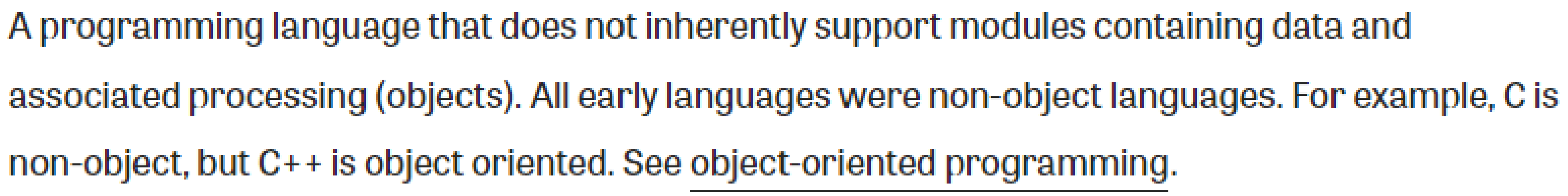
Object-oriented programming allows procedures about objects to be created whose exact type is not known until runtime. For example, a screen cursor may change its shape from an arrow to a line depending on the program mode. The routine to move the cursor on screen in response to mouse movement would be written for "cursor," and polymorphism allows that cursor to take on whatever shape is required at runtime. It also allows new shapes to be easily integrated.

OOP Languages

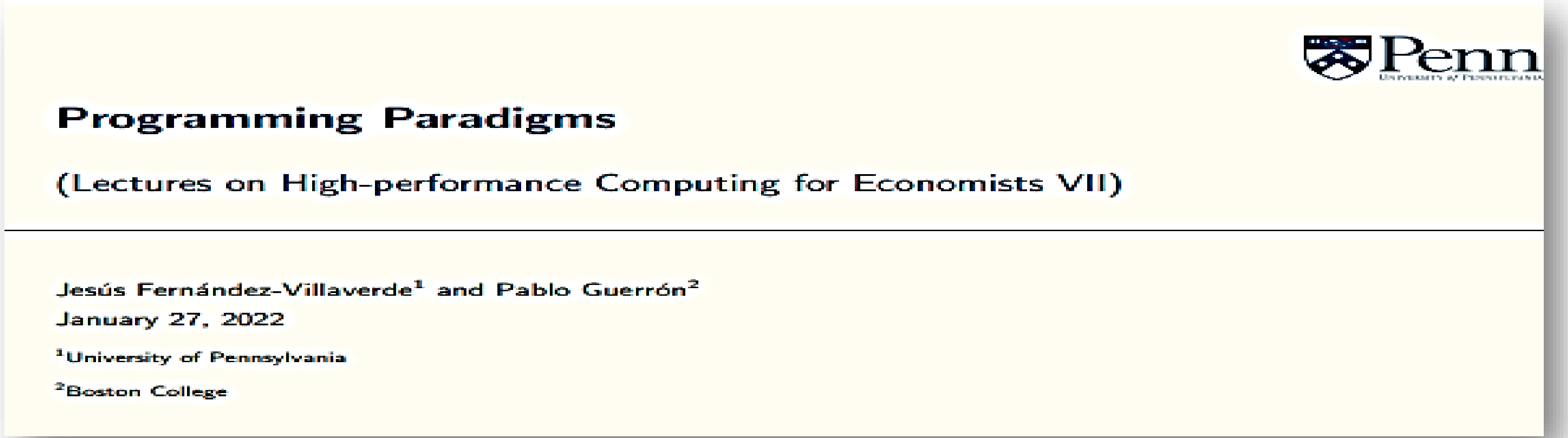
Used for simulating system behavior in the late 1960s, SIMULA was the first object-oriented language. In the 1970s, Xerox's Smalltalk was the first object-oriented programming language, which was used to create the graphical user interface (see Xerox Star). ACTOR and Eiffel were also earlier OOP languages.

Today, C++, C#, Java, JavaScript, Visual Basic.NET and Python are popular object-oriented languages. The following compares basic OOP terms with traditional programming. See object-oriented DBMS.

2 Not object programming



A programming language that does not inherently support modules containing data and associated processing (objects). All early languages were non-object languages. For example, C is non-object, but C++ is object oriented. See object-oriented programming.



Penn

Programming Paradigms

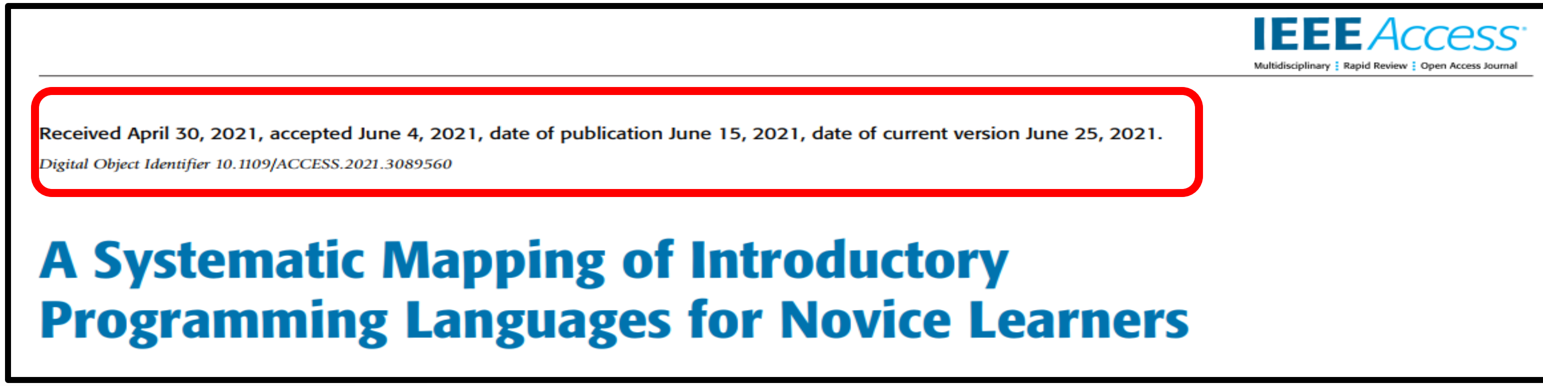
(Lectures on High-performance Computing for Economists VII)

Jesús Fernández-Villaverde[1] and Pablo Guerrón[2]
January 27, 2022

[1]University of Pennsylvania

[2]Boston College

OOP	Multi paradigm languages
C	C^{++}
Older versions of MatLab	MatLab
Fortran	



IEEE Access
Multidisciplinary : Rapid Review : Open Access Journal

Received April 30, 2021, accepted June 4, 2021, date of publication June 15, 2021, date of current version June 25, 2021.
Digital Object Identifier 10.1109/ACCESS.2021.3089560

A Systematic Mapping of Introductory Programming Languages for Novice Learners

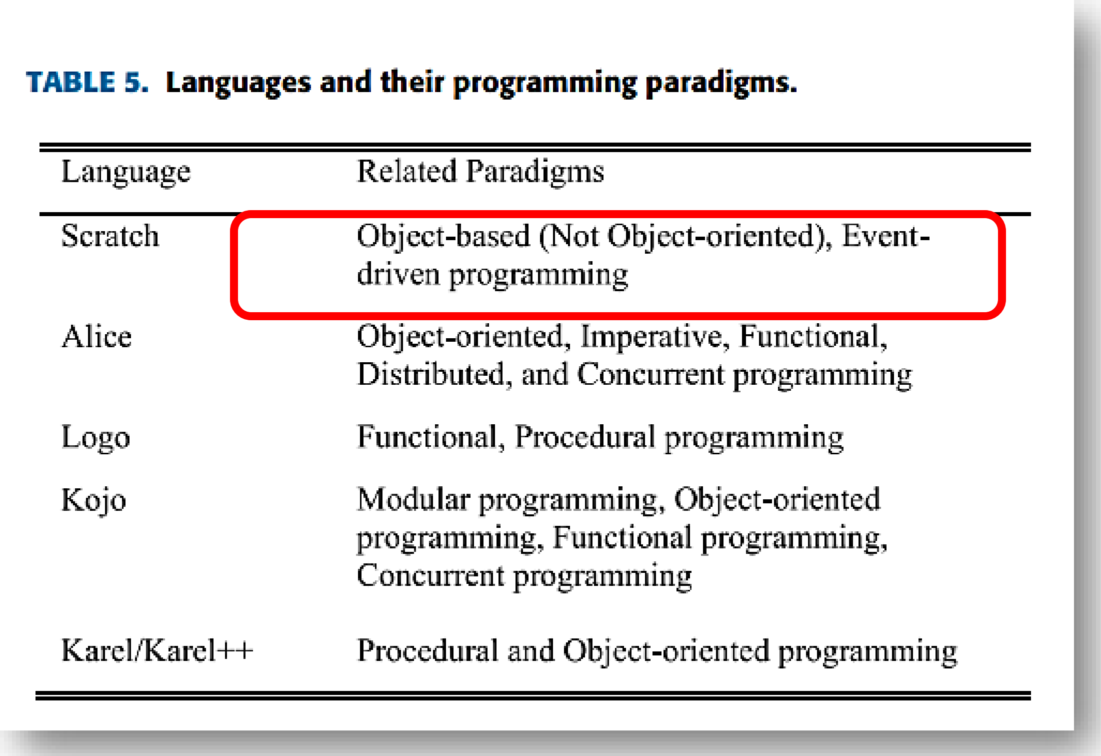


TABLE 5. Languages and their programming paradigms.

Language	Related Paradigms
Scratch	Object-based (Not Object-oriented), Event-driven programming
Alice	Object-oriented, Imperative, Functional, Distributed, and Concurrent programming
Logo	Functional, Procedural programming
Kojo	Modular programming, Object-oriented programming, Functional programming, Concurrent programming
Karel/Karel++	Procedural and Object-oriented programming

Links and references

https://www.technologyreview.com/-ynti-htiw-gninrael-enihcam-ia/٢٠٢٠/١٠/١٦/١٠١٠٥٦٦
laicos_rt=muidem_mtu&tnemegagne.diapnu.rotisiv_etis=ngiapmac_mtu&rettiwT=ecruos_mtu؟/atad

https://futurology.life/21-most-innovative-egypt-based-artificial-intelligence-companies/

https://github.com/harryadel/AI-ML-Driven-Companies-In-Egypt/blob/master/README.md#aiml-driven-companies-in-egypt

https://datamagazine.co.uk/-enihcam-evtiavonni-tsom-١٨
/tpyge-seinapmoc-sputrats-gninrael

https://www.f6s.com/companies/artificial-intelligence/egypt/co

https://www.sas.upenn.edu/~jesusfv/Lecture_HPC_fdp.smgidaraP_gnimmargorP_٧ https://doc.lagout.org/science/%20retupmoC ٠Science/%20selpicnirP ١of%20Programming %20Languages/Programming%20Languages%20-%20Principles%20and%20Paradigms.pdf

https://www.researchgate.net/publication/2336195