COMP1811

Paradigms of Programming

Object-Oriented Programming

Why Python?

Dr. Yasmine Arafa

Object-Oriented Programming - OOP

Python

- Python is covered on COMP1811 as an example of an OOP language.

- But, in reality, it is a multi-paradigm (hybrid) programming language:
  - Primarily Object-Oriented, but also
  - Supports many elements of:
    - **Procedural Programming:**
      - code is broken into functions,
      - commands within these functions are executed sequentially, and
      - can have side-effects.

        (Side-effects happen if a function modifies some state variable value(s) outside that function.)

Object-Oriented Programming - OOP
Python, *Cntd.*

- Supports many elements of:
  - **Functional Programming (FP):**
    - Python functions can return only one value,
    - this functions can deigned to disallows side effects, and
    - other functional programming features also supported.
- Term1 will cover the OOP features of Python and part of term2 will cover FP in Python.
- An Objects-last approach will be used to cover Python, which means we’ll:
  - cover the Python basics and language constructs first,
  - then cover the OOP features of Python.

Python - OO Programming

about...

- Developed by Guido van Rossum in the 1980's
  - only became popular in the last decade or so
  - Name inspired by the Monty Python British comedy group
- Two common versions Python 2.x and Python 3.x
  - Python 2.x dominates, but was deprecated in January 2020.
  - Python 3.x is the future of Python and this module will only cover 3.x
- Supports a multitude of programming paradigms
  - OOP, procedural, functional, etc.

About Python
cntd.

- General purpose
  - Comprehensive standard libraries includes numeric modules, GUI support, development tools, data sciences, machine learning, crypto services, OS interfaces, networking modules, etc.
- Is generally an *interpreted* language,
  - **Interpreted** means that Python code is converted to machine code and executed by another program (the Python interpreter) one statement at a time at run-time.
  - For **Compiled** code, the entire source code is compiled into machine code on one machine and then executed altogether on a target machine.
  - Whether the code is interpreted or compiled is not a property of the language but a property of the implementation.

Why Python?

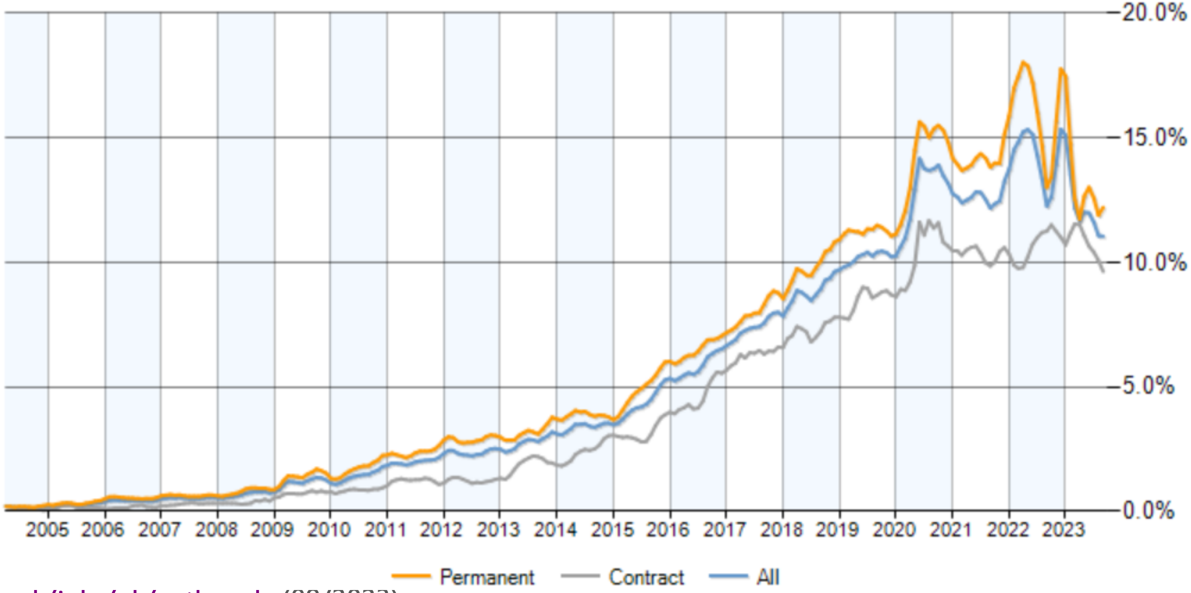
because of its notable features…

- Easy to learn.
- Supports quick development.
- Cross-platform.
- Open Source.
- Extensible.
- Embeddable.
- Large standard library and active community.
- Useful for a wide variety of applications.

Why Python?

because of the Jobs Demand Trend - UK…

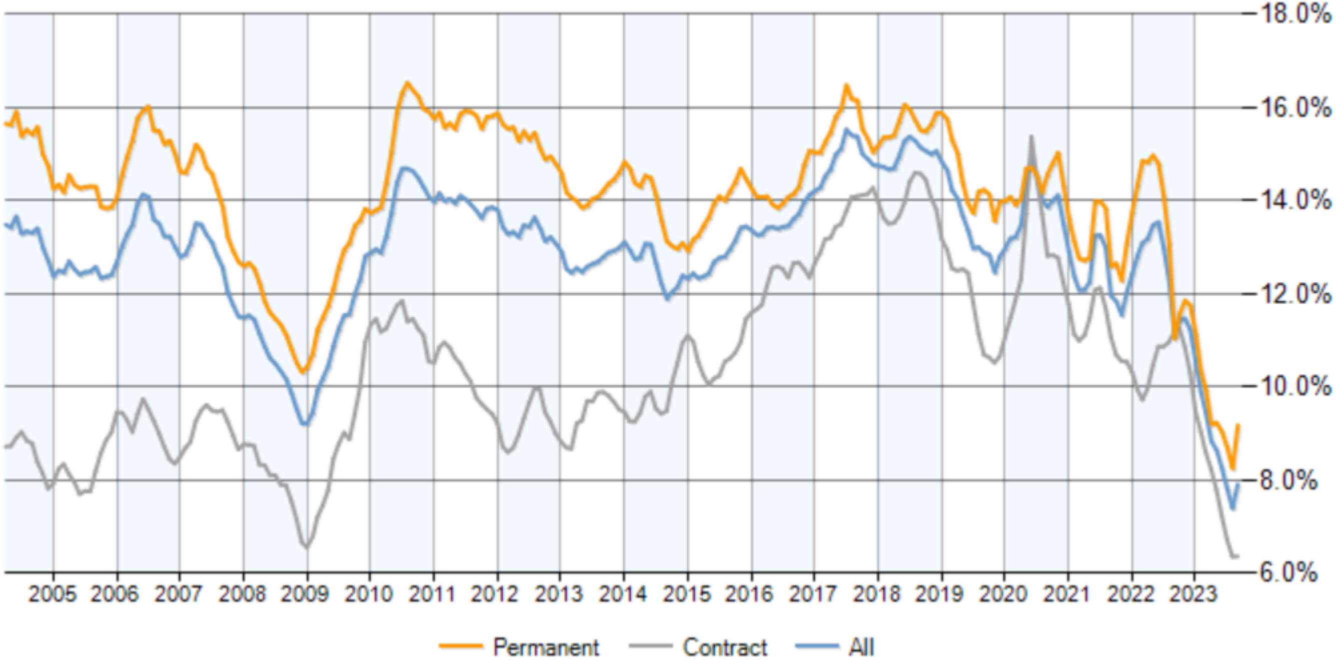
- The trend of job postings citing Python as a proportion of all IT jobs with a match in the Programming Languages category.



https://www.itjobswatch.co.uk/jobs/uk/python.do (09/2023)

Why Python?
because of the Jobs Demand Trend - UK...

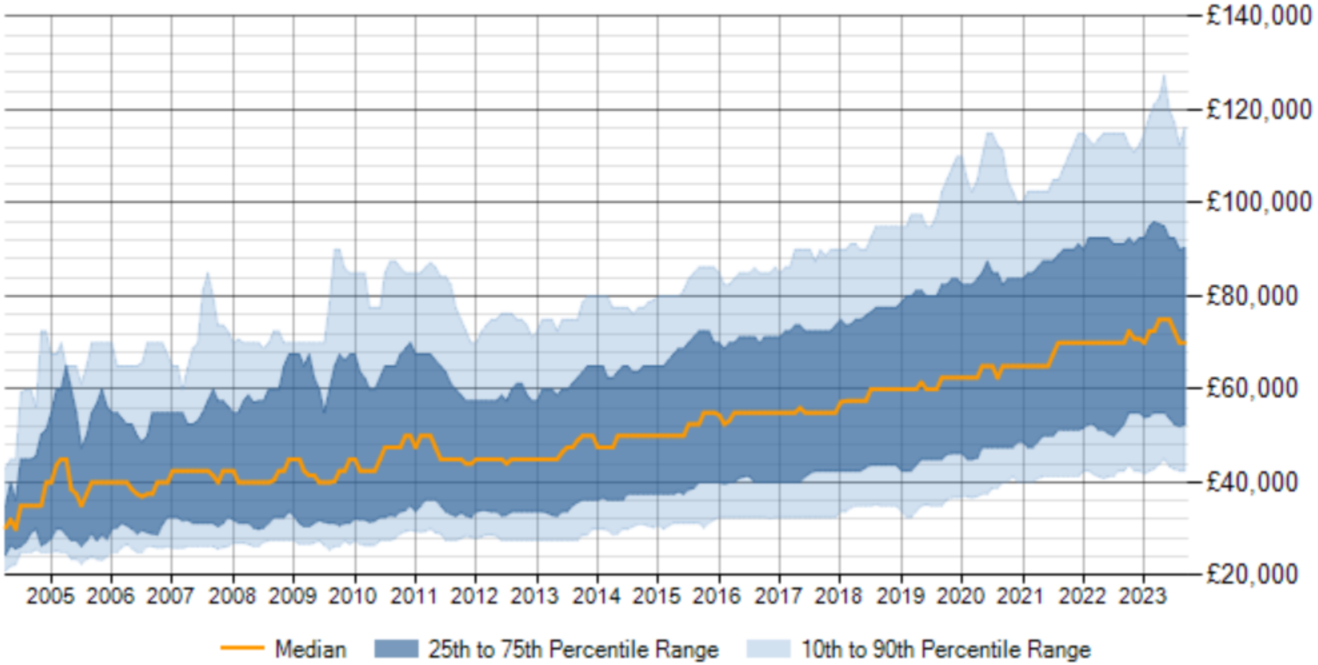
- Compared to Java...



https://www.itjobswatch.co.uk/jobs/uk/java.do (09/2023)

Why Python?
Salary Trend - UK

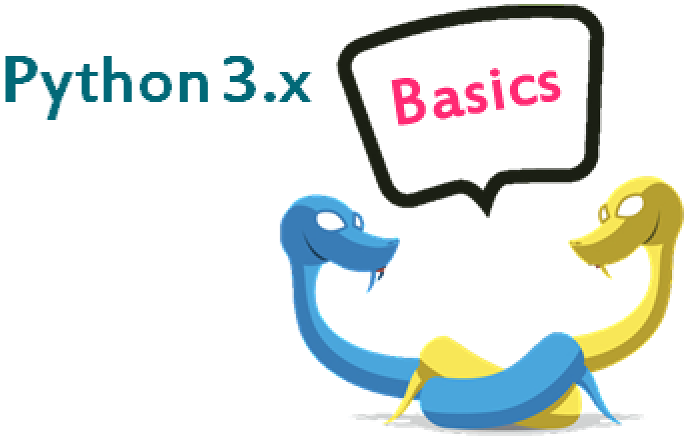
- Take a look at the 3 month moving average for salaries quoted in permanent IT jobs citing Python in the UK.



https://www.itjobswatch.co.uk/jobs/uk/python.do (09/2023)

Let's get started...

Python basics introduced in the next section...



Adapted from https://www.cleanpng.com