

# Thoughts on the Future of Python

EuroPython 2021 – 29.07.2021

Joining from Düsseldorf, Germany

Marc-André Lemburg :: eGenix.com GmbH

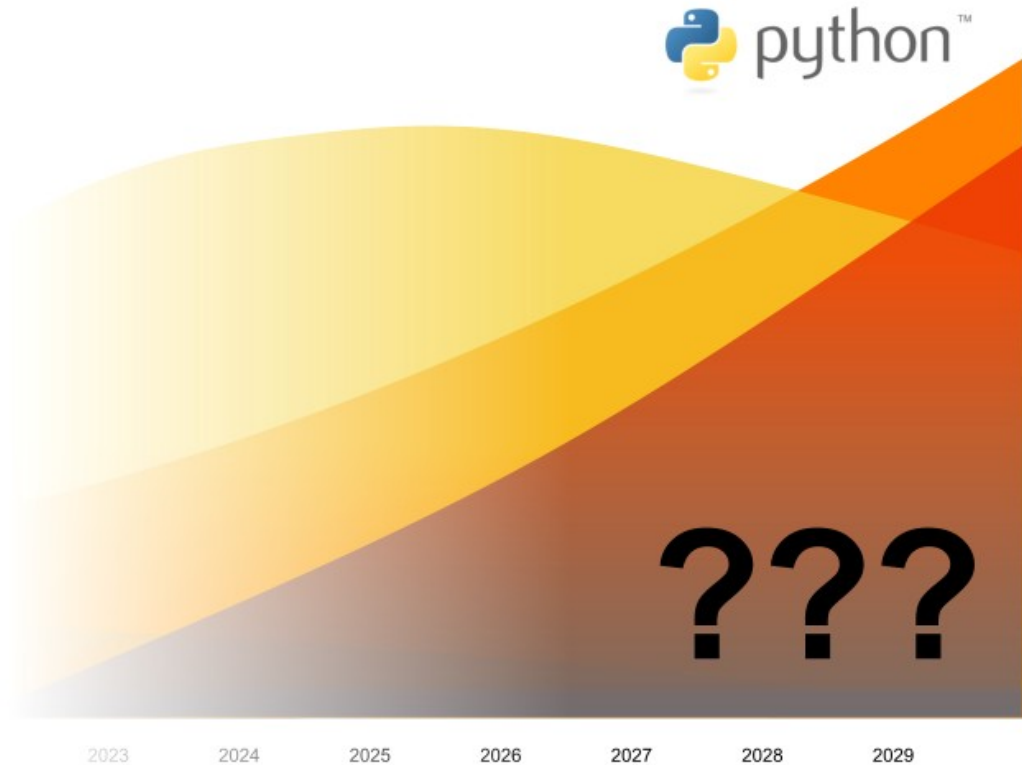
# Speaker Introduction

- Marc-André Lemburg
  - Python since 1994
  - Studied Mathematics
  - CEO eGenix.com GmbH
  - Consulting CTO and Senior Software Architect
  - EuroPython Society Chair
  - Python Software Foundation Fellow
  - Python Core Developer
  - Based in Düsseldorf, Germany
  - More: <http://malemburg.com>



# Agenda

- Python development
- Growth in the market
- Perspectives for the future
- My wish list for Python
- Discussion



## Setting the stage ...

Python has been around for long time.  
Is it still evolving ?

# A bit of history

- 1990s
  - Started in Dec 1989 by [Guido van Rossum](#) (BDFL)
  - Several copyright owners in the first 12 years
  - Development mostly via the Python Labs Team using CVS
- 2000s
  - Open development, Python Core Developers
  - PEP process for managing change
  - [Python Software Foundation](#) initiated in 2001, managing licensing
  - Moving from CVS to Subversion
- 2010s
  - Moving from Subversion to Mercurial to Git
  - [Python 2/3 transition](#) (2008 - 2020, and still on-going)

# Recent developments in Python (2015-2021)

- **Type annotations**
  - Originally only for static analysis
    - mypy, pyre, pytype, IDEs, etc.
  - Now, also used at run-time
    - dataclasses, struct2, pydantic, etc.
- **Dataclasses**
  - Efficient typed records
- **AsyncIO**
  - Native co-routines in Python
- **Contextvars**
  - Similar to thread-local variables, but for async code
- **Builtin ordered dict**
  - Insert order preserved
  - Keyword argument order preserved

# Recent developments in Python (2015-2021)

- New operators
  - Dict merge (d1 | d2)
  - Walrus operator (`:=` assignment expressions)
- Positional / keyword only parameters
  - “Backported” from C API
- New parser PEG
  - Allows for context local “keywords”
- f-Strings
  - String formatting with direct access to locals()
  - Vastly successful
- Switch/match statement
  - Structural pattern matching
  - Type / Object matching

# Recent developments in Python (2015-2021)

## Organizational Changes

- New release cycle
  - Yearly releases
- Steering Council (SC)
  - Guido stepped down as BDFL
  - PEP process now guided by SC
- PSF Developer in Residence
  - Full time dev dedicated to Cpython
  - Helps lower # of open PRs
- Guido's post-retirement
  - Now working at Microsoft with team dedicated to improving performance of CPython



# Developments currently under discussion

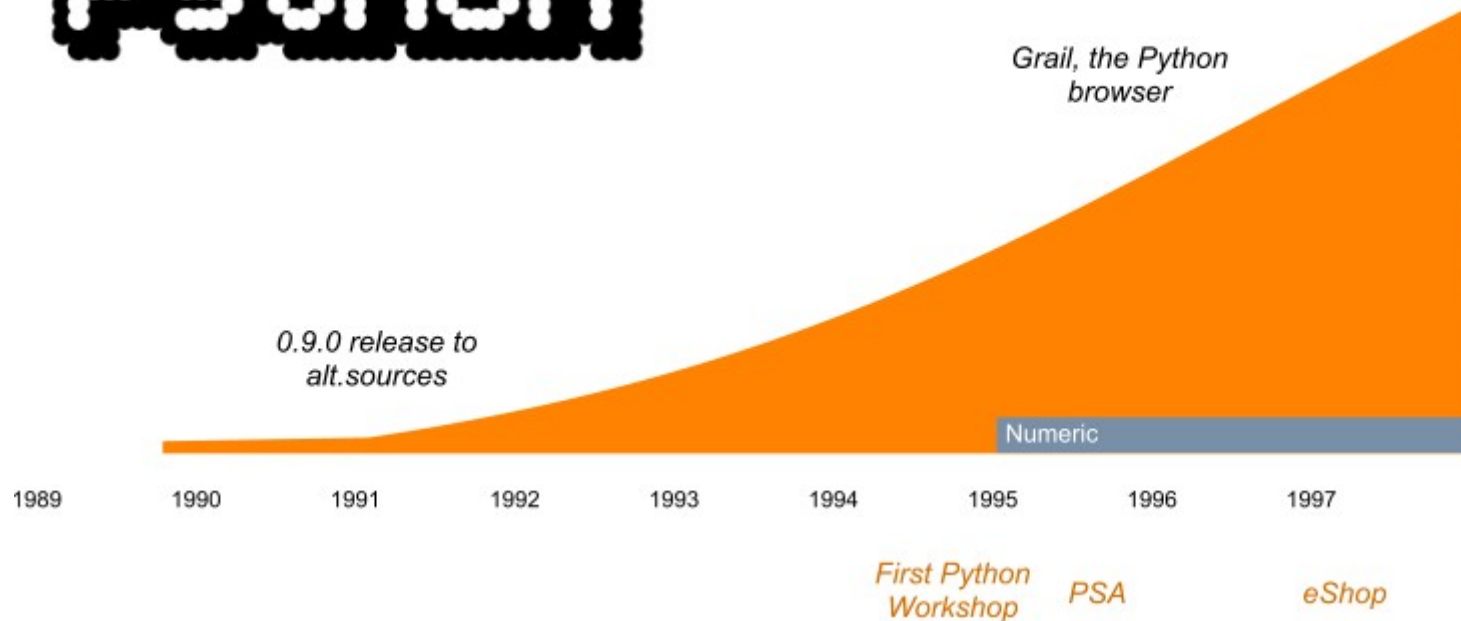
- Improving CPython performance
  - Specializing adaptive interpreter
  - Many smaller improvements
- Enhanced error reporting
  - Tracebacks include columns information
- Sub-Interpreters
  - More than one interpreter per process
  - Work around GIL limitations
- C API reworking
  - Hide more internals
  - Reduce complexity
  - Python handle project PyH
  - Prepare for removal of GIL
  - Prepare for change to garbage collection for memory management
- No end in sight ...

# Python development looks healthy ...

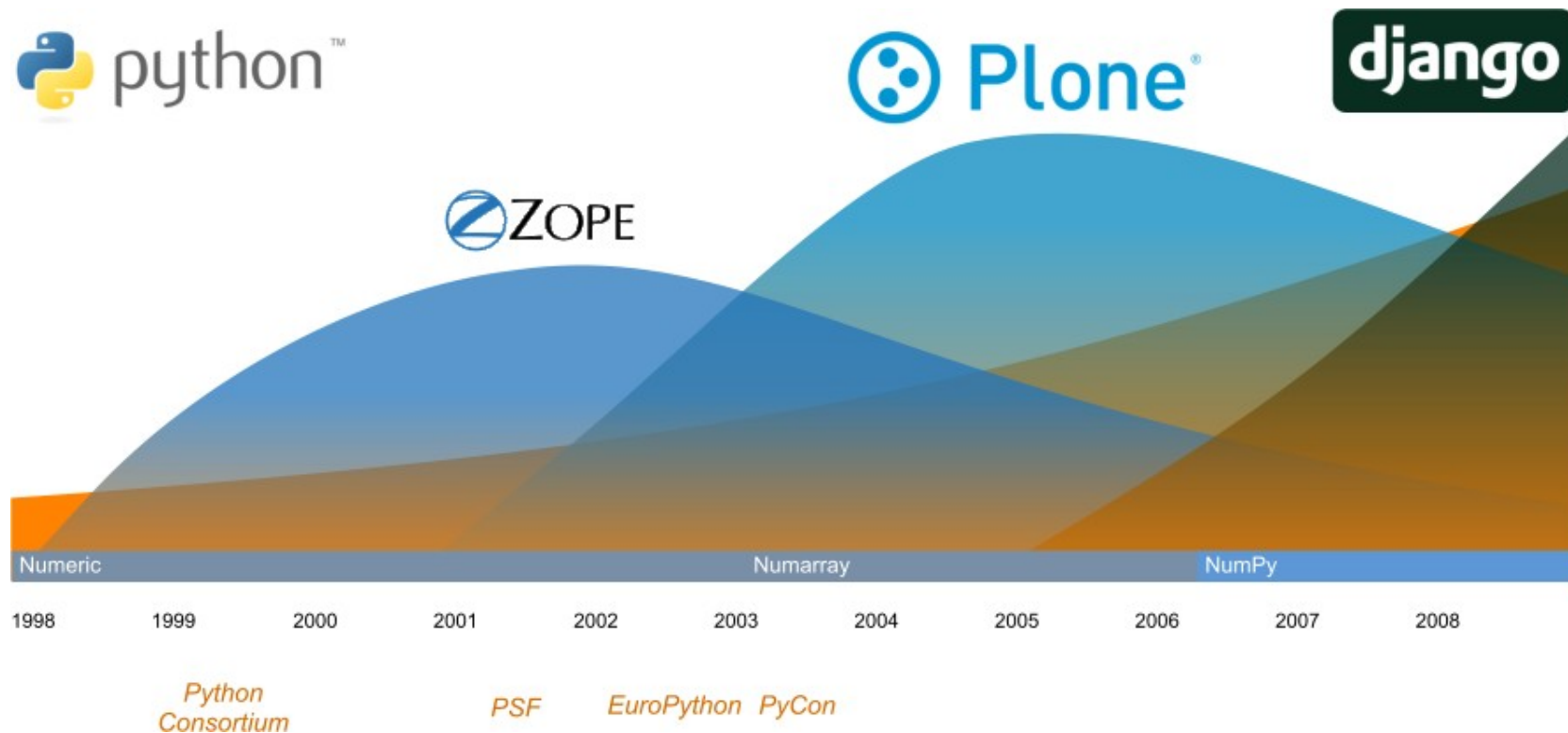
Let's have a look at  
Python's growth in the market

# Python's Growth: Initially, just Academia

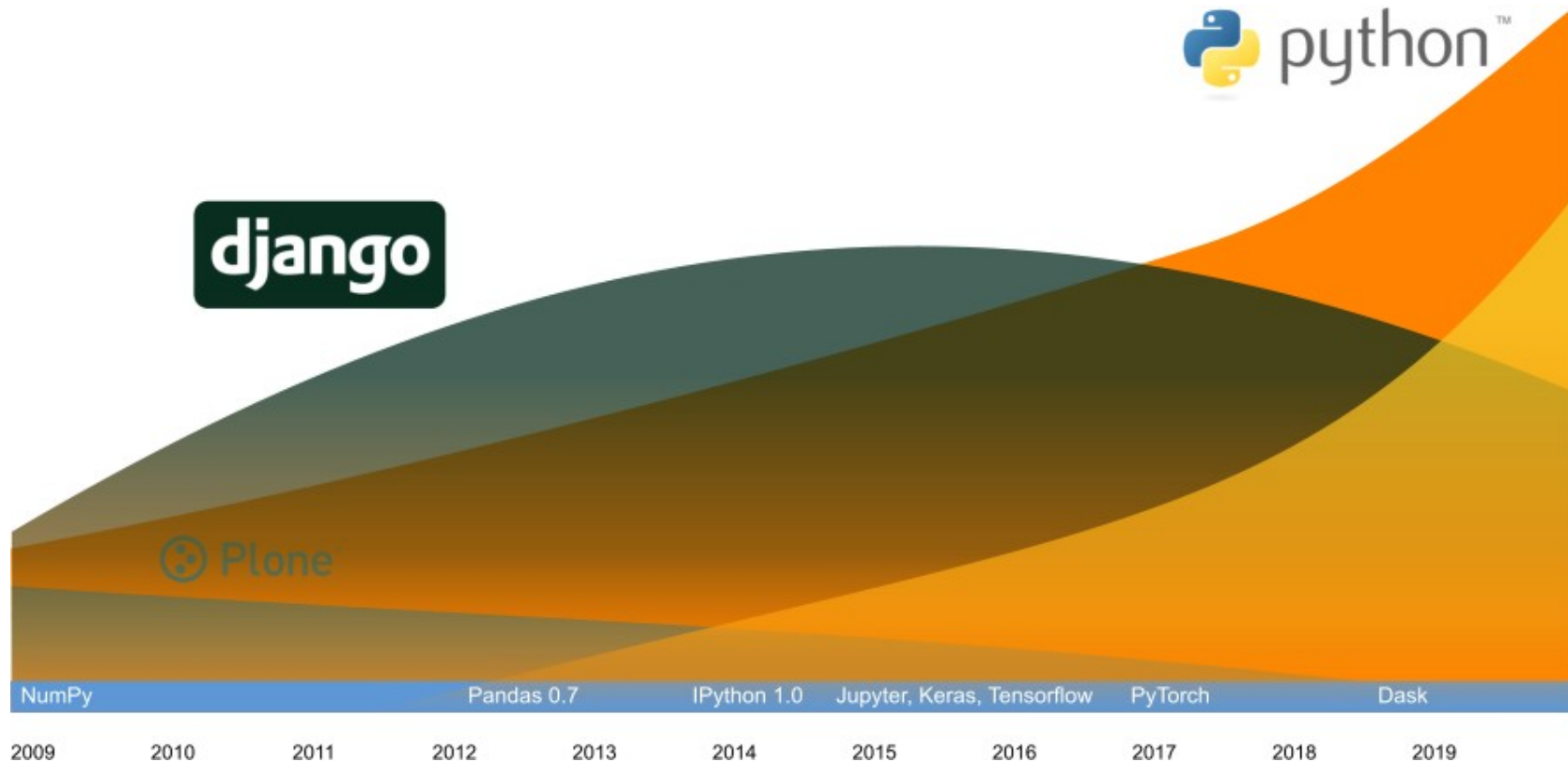
python



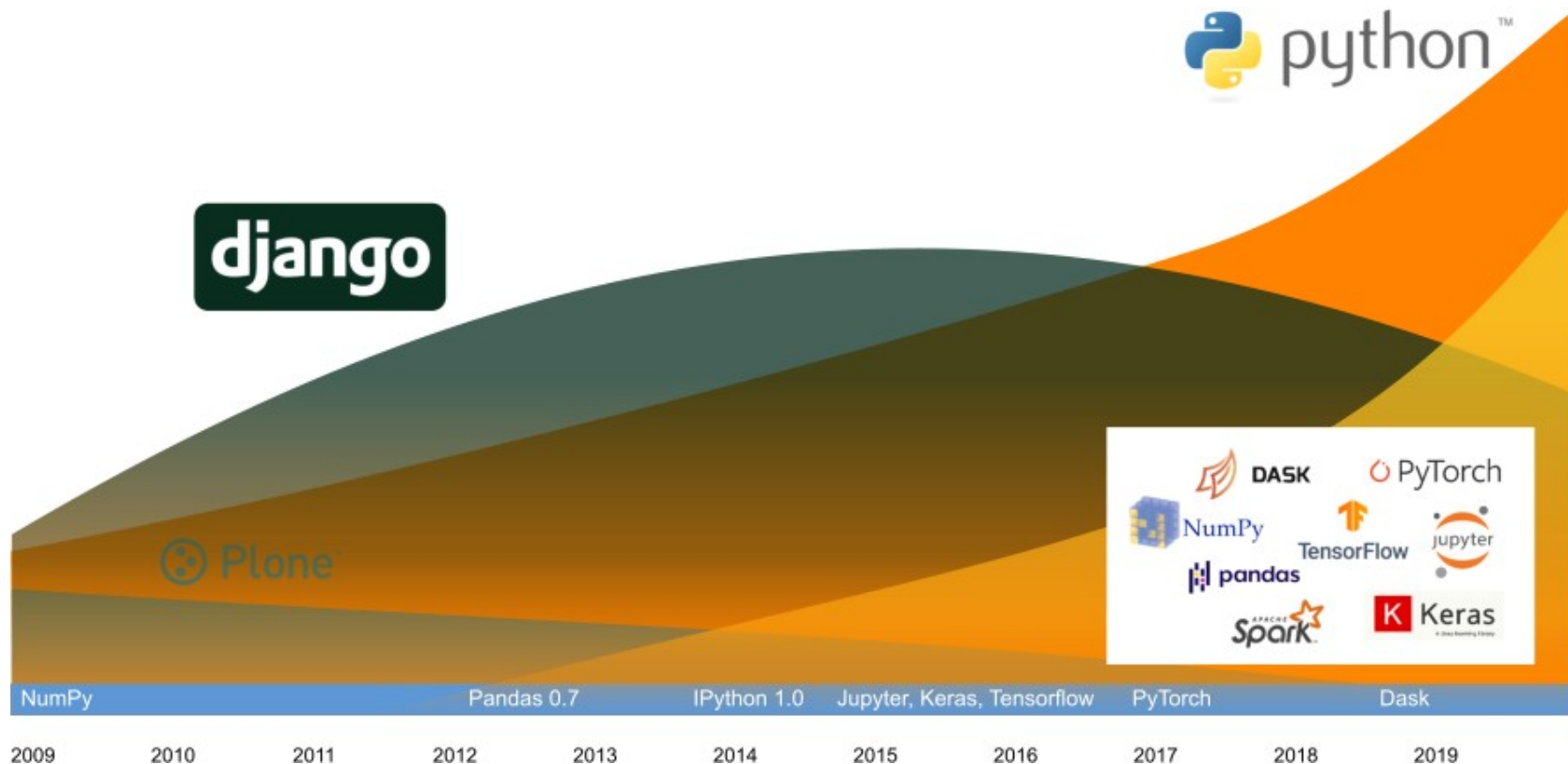
# Embracing the Web



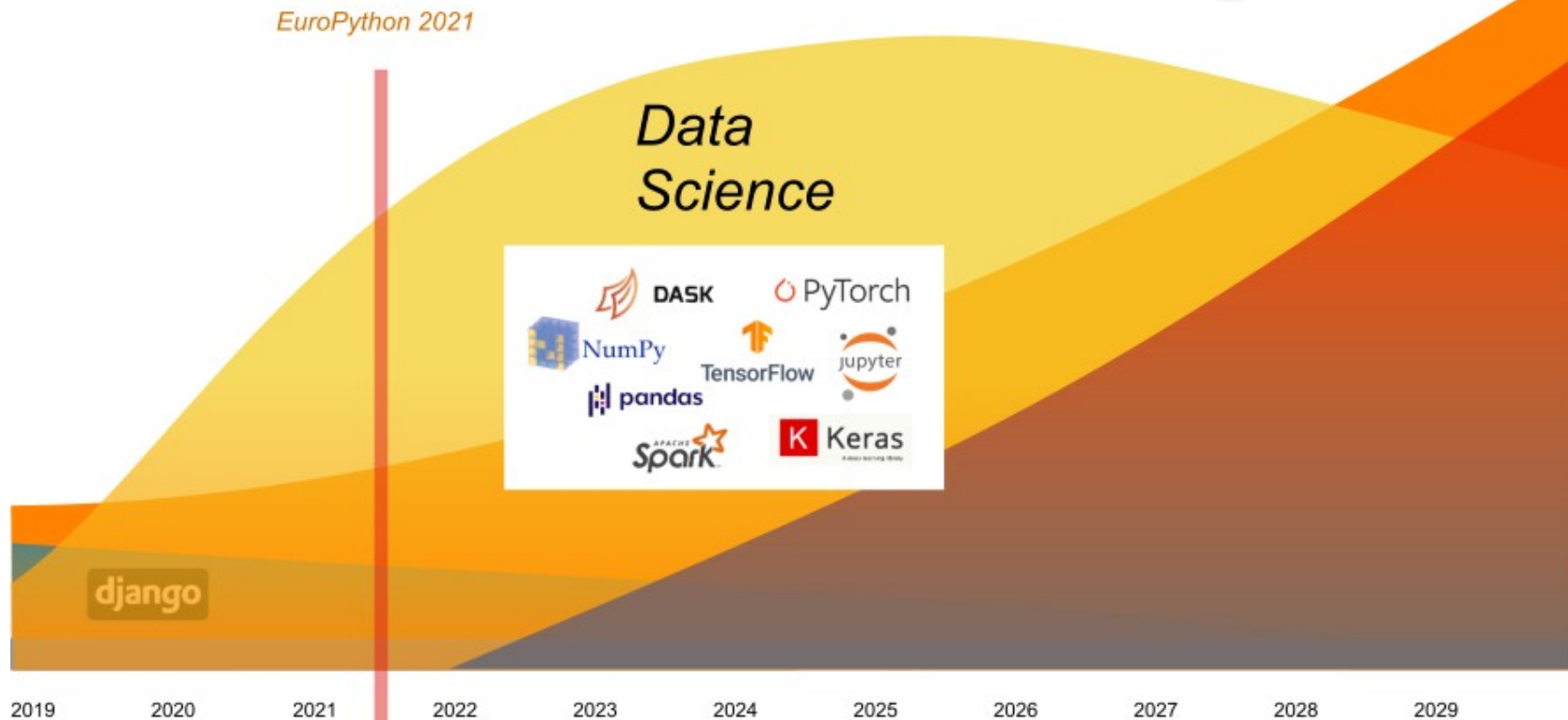
# Entering the mainstream



# Rise of Data Science

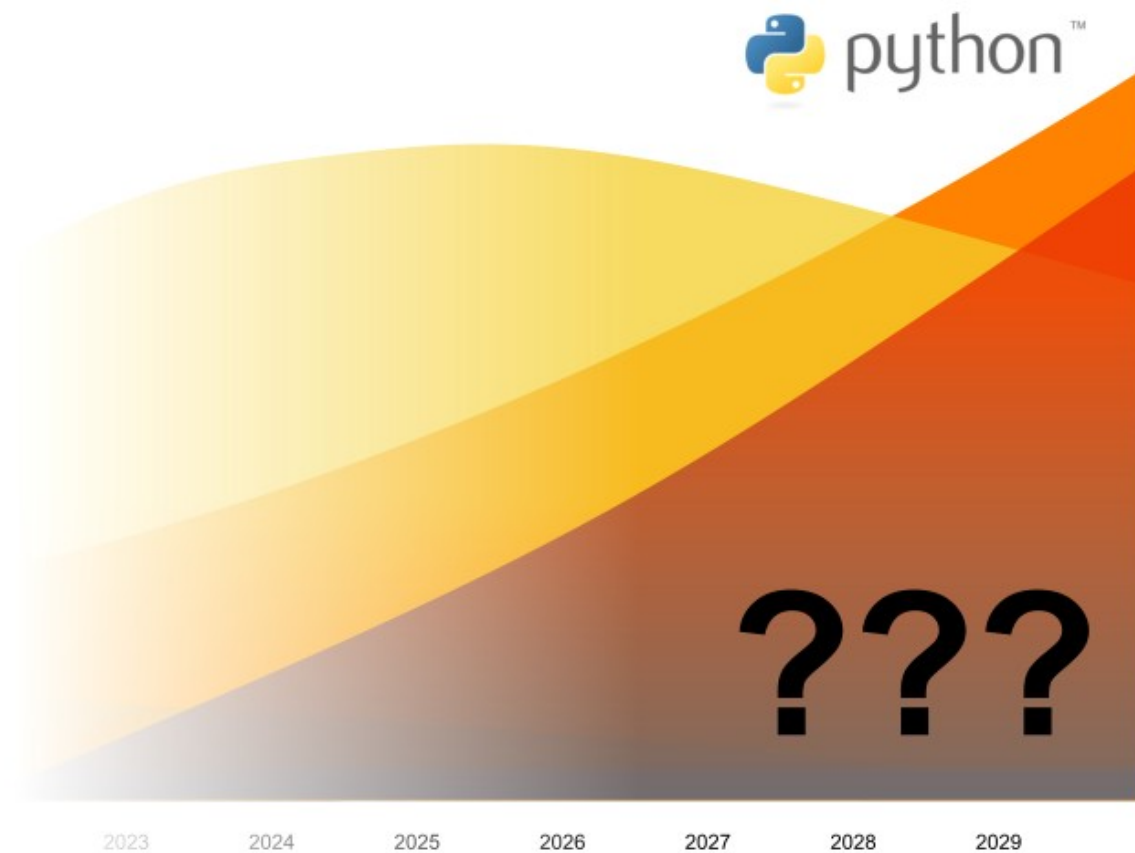


# Today: Python is all over Data Science





# The Future ?





# Analysis of where we are ...

- Python is still big in web
  - But just one of many implementation languages (JavaScript, Go, Ruby, PHP, etc.)
  - We'll see more adoption of asyncio frameworks
  - Web is moving towards API backends using smart frontends (“client-server” as we used to call this in the 1990s)
- Python is well positioned with great support for API backends
  - FastAPI
  - Django / Flask REST Frameworks
  - Graphene / Ariadne / Strawberry / Tartiflette GraphQL libraries
  - etc.

# Analysis of where we are ...

- Python is very big in data science
  - Python and R are the two main languages used in data science
  - Defacto standard for interfacing to AI/ML models
  - Time-to-market is more important than single core run time
  - This dramatically increased the scope and market share of Python
- Distributed computing is the new norm
  - Many tools are being developed to make Python perform even better for data science workloads
  - The Python GIL discussion is becoming less important

## Two markets covered – what's next ?

- Python is entering the Enterprise via data science
  - All data platforms support Python right from the start
  - Apache AirFlow (Workflow) and Superset (BI) are written in Python
  - Apache Spark (Data Platform) runs PySpark scripts
- Decision making is moving towards being data driven
  - More and more business people will get to learn about Python
  - Machine learning will be on the rise within the decision making process
  - This market will continue to grow in the next two decades

**Sounds too good to be true ...**

Is Python ready for the Enterprise ?

**Well, not quite there yet ...**





# Putting on the CTO hat and playing devil's advocate

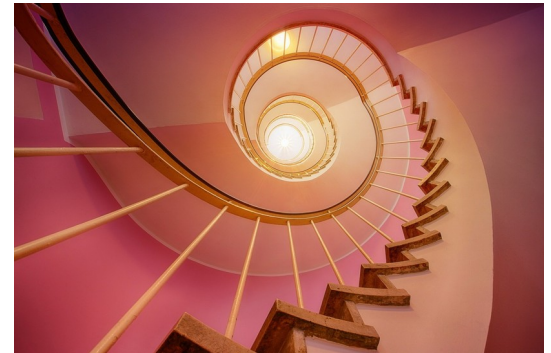
- Python's yearly release cycle poses challenges for the Enterprise
  - On the plus side, we get faster access to new features
  - Every new Python release will require testing and updating lots of software
  - We only get 1.5 years support for each version
  - Given that you typically only start working with a new version after a few patch level releases, this evaluates to just one year support
  - CI and lots of tests can help mitigate this
- Effects
  - Companies will skip releases
  - Getting commercial Python support will become more important



# Is Python Core Development helping Enterprise growth ?

*“That’s a very good question” 😊*

- Some core developments are good for this
  - Added **type safety** (type annotations and tooling around it)
  - Sub-interpreters to help using all cores
  - Keyword indexing for the data science community
  - AsyncIO for the web community
- For other core developments, it’s not so obvious
  - Adding more and more syntax,  
**raising the bar for newbies**
- But: **There’s more ...**



# We have many communities working for Python's success

- **Data science** community
  - Making Python the standard for AI/ML, data modeling and analysis
- **Web** community
  - Moving complexity out of the way
- **Devops** community
  - Using Python's platform independence for everyone's benefit
- **Other communities**
  - Building on Python's ease of use and deployment to improve efficiency and accessibility
- **Industry leaders demonstrating how they use Python**
  - Companies such as AirBnB, Netflix, Google, etc. open sourcing solutions they have written in Python





# The Future: In the cloud ?

- Python as the **cloud language** ?
  - Already in use in Ansible for devops
  - Good APIs and support for all cloud providers (individual and via Apache Libcloud)
  - OpenStack largely written in Python
  - Python runs in the browser
  - AWS Lambda, Google Cloud Functions, Azure Functions all run Python
  - Strong competitor: Node.js (JavaScript)
  - In pretty good shape already, but needs more market awareness



# The Future: As the new enterprise language ?

- Python as **business application language** ?
  - Strong competition from Java / C++ / C#
  - **Most business apps written in Java, even new ones**
  - Consulting companies love long running complex projects...
  - **“Nobody ever got fired for choosing ~~IBM~~ Java”**
- Perhaps better approaches
  - Integrate applications and lower level libraries
  - Orchestrate compute, tests and tooling
  - Provide accessibility to complex systems
  - **Play the Trojan Horse card**  
(bottom up adoption, just like in data science)



# Lots of potential in the Enterprise Application world

- ERP (Enterprise Resource Management)
  - **Odoo** (SMEs only)
- MDM (Master Data Management)
  - ???
- BI (Business Intelligence)
  - **Superset** (focus on data science)
- ETL (Extract Transform Load ... convert data)
  - **Airflow, PySpark** (focus on data science)
- ECM (Enterprise Content Management)
  - **Plone** (little use for this purpose)
- Web front end for EAs (Enterprise Applications)
  - ???
- **All predominantly still Java**
- **Billions in market size**
- Talk at EP2018:  
*Making money using Python*





**We'll get there, just takes a bit of effort & time**

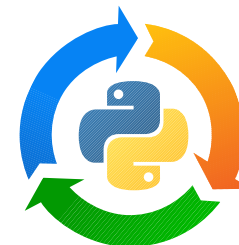


# Pushing Python to play with the big guys

My wish list for Python.

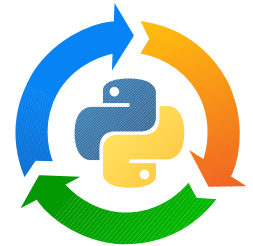
# Wish list for Python: Language Level

- Not another 2-to-3 transition
  - Will be difficult with the (needed) C API changes
- A **Python LTS version**
  - 5 years support
  - Another 5 years security fixes
- Python without a GIL
  - Or better ways to share data between processes
- Simplified C API
  - Use Cython for wrapping C code
- **Mobile app support**
  - Android and iOS
- Faster CPython
  - Native
  - Using tools such as Cython, Numba, etc.
  - Addition of fast native data structures (e.g. for graphs)
- **Syntax for parallel execution**
  - Based on Multiprocessing, MPI, Dask, etc.



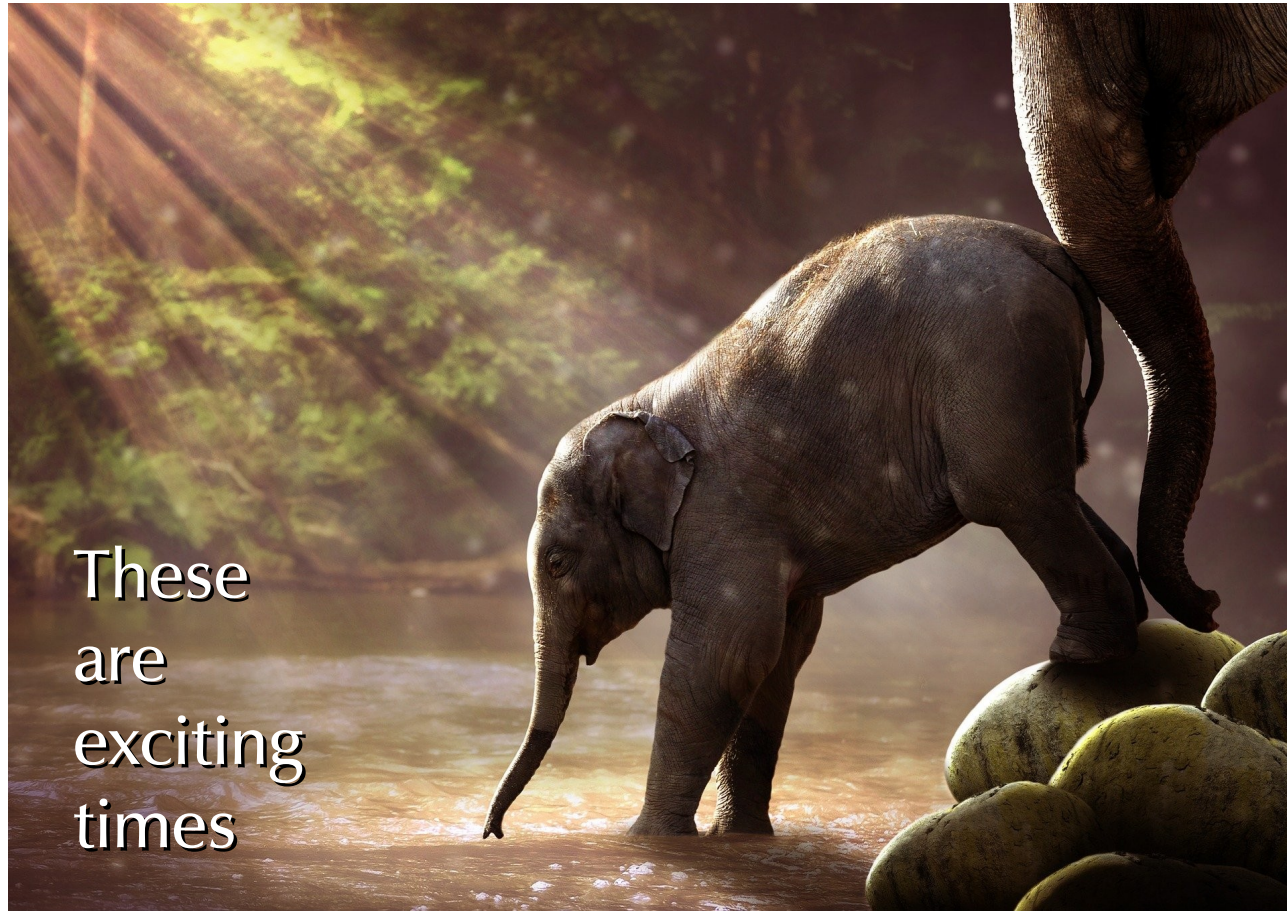
# Wish list for Python: Application Level

- An “Anaconda” style environment for business apps
  - Well integrated tooling, IDE, etc.
  - Builtin deployment tools
  - App store integration
- **Rapid development solutions**
  - Readily usable front-end / back-end components
  - Database, workflow, user and session components
  - Auditing, sign-on, admin components
- **More polished libraries**
  - Business graphics
  - Business UIs and dashboards
  - Well maintained integration libraries
  - Backed by companies
- **More paid offerings**
  - Support, SLAs
  - Easy access to consultants





# Python has a bright future ahead



These  
are  
exciting  
times



**Thank you for your attention !**



Time for discussion

# Contact

**eGenix.com Software, Skills and Services GmbH**

Marc-André Lemburg

Pastor-Löh-Str. 48

D-40764 Langenfeld

Germany

eMail: [mal@egenix.com](mailto:mal@egenix.com)

Phone: +49 211 9304112

Fax: +49 211 3005250

Web: <http://www.egenix.com/>

# References

- Some photos taken from Pixabay, details available on request
- All other graphics and photos are (c) eGenix.com or used with permission
- Logos are trademarks of their resp. trademark holders