Make your life Easy

| *Marc Philippe de Villeres* |

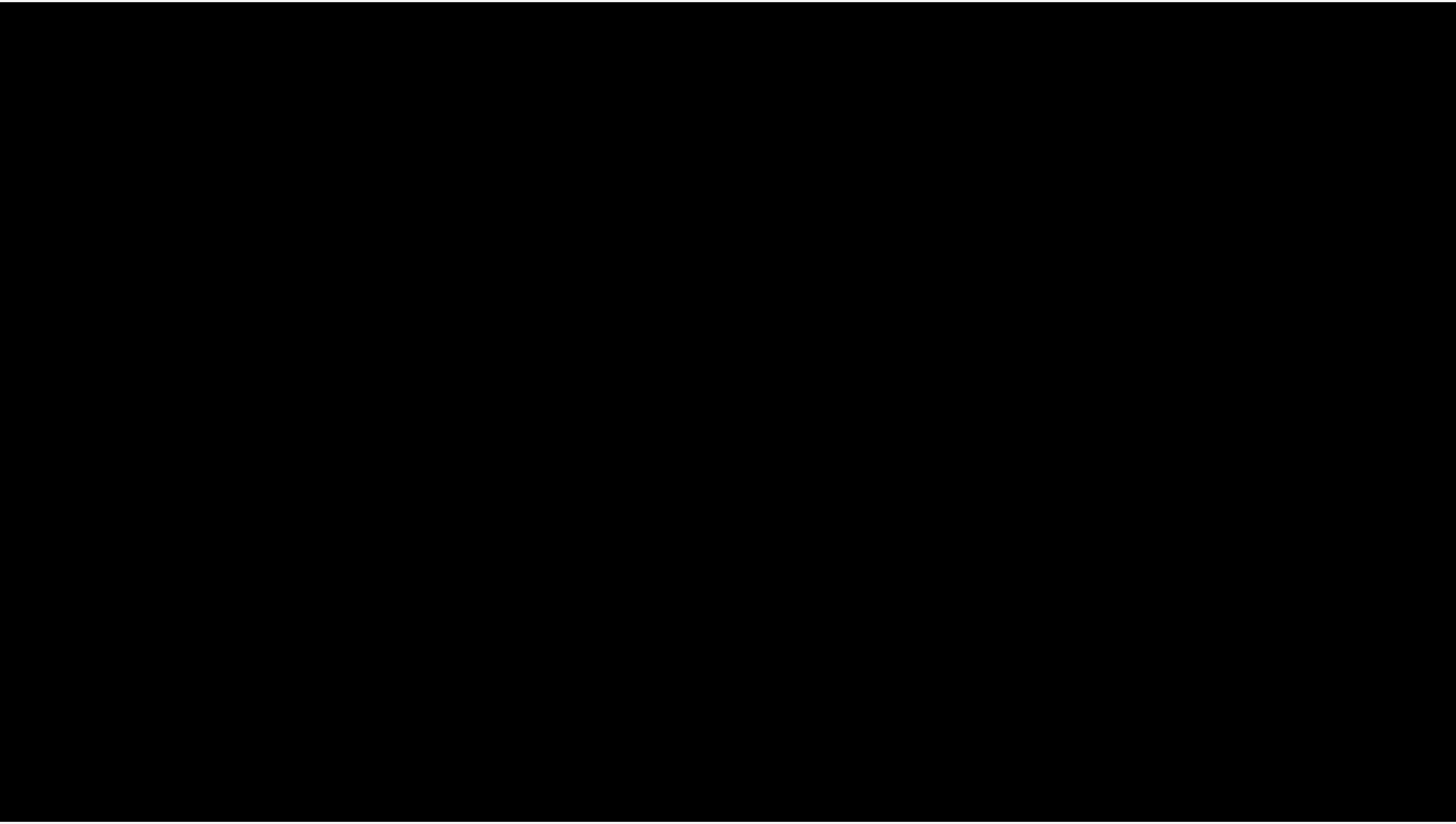
About me

Electronics

Software

Open Source

Data



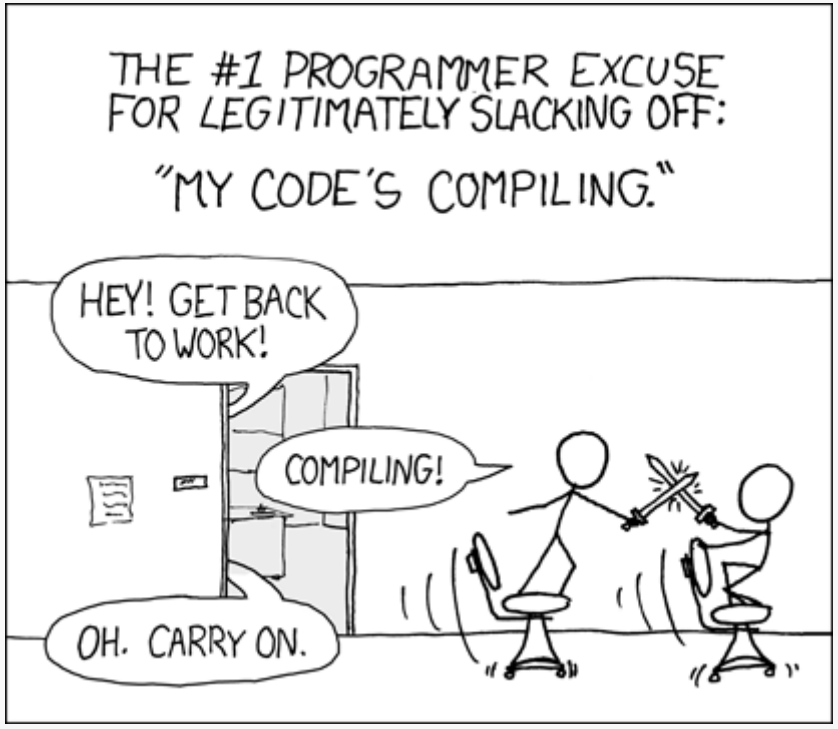
https://www.youtube.com/watch?v=nKIu9yen5nc
Credit: code.org

What is Python

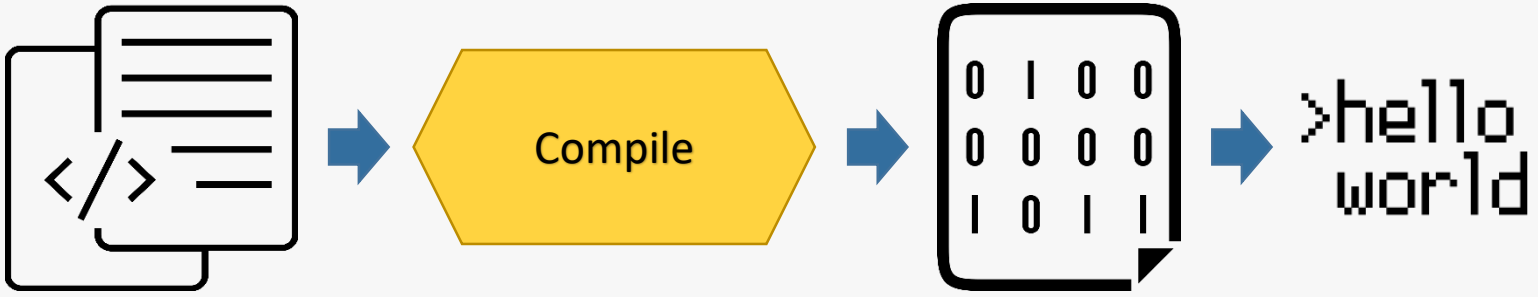
"Python is an interpreted, object-oriented, high-level programming language with dynamic semantics." – python.org

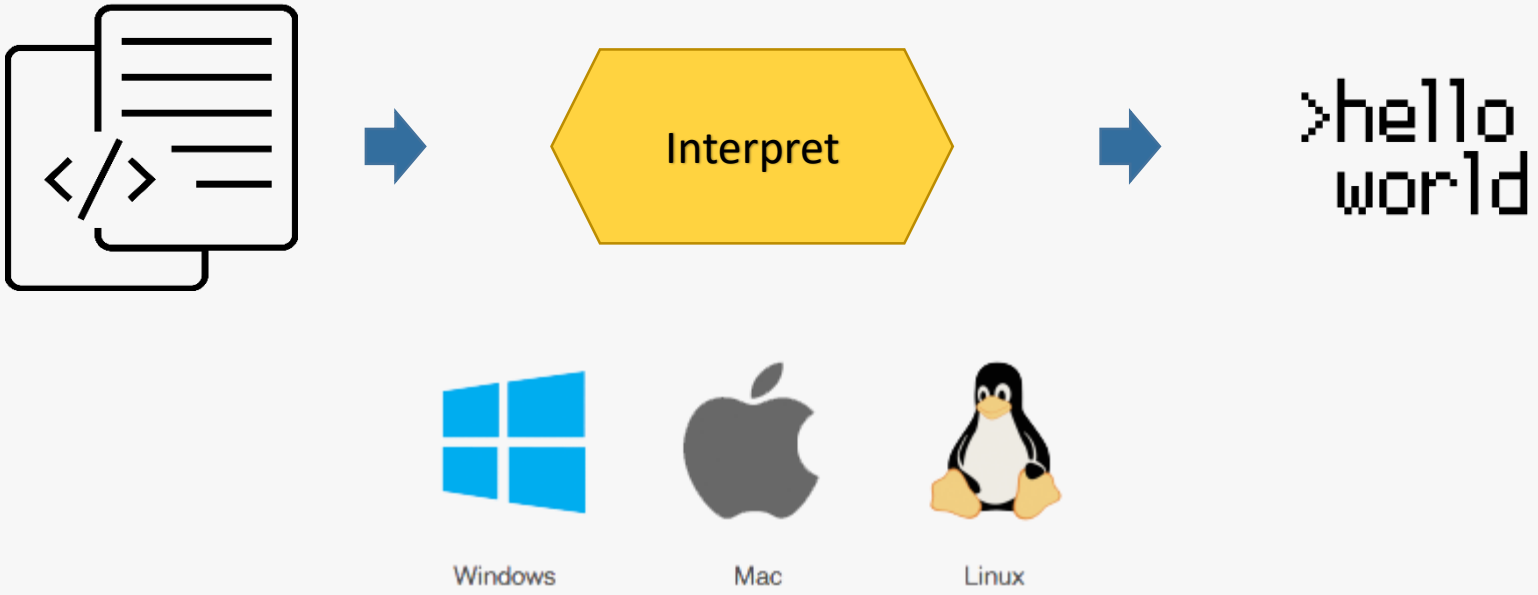
What is Python

"Python is an interpreted, object-oriented, high-level programming language with dynamic semantics." – python.org



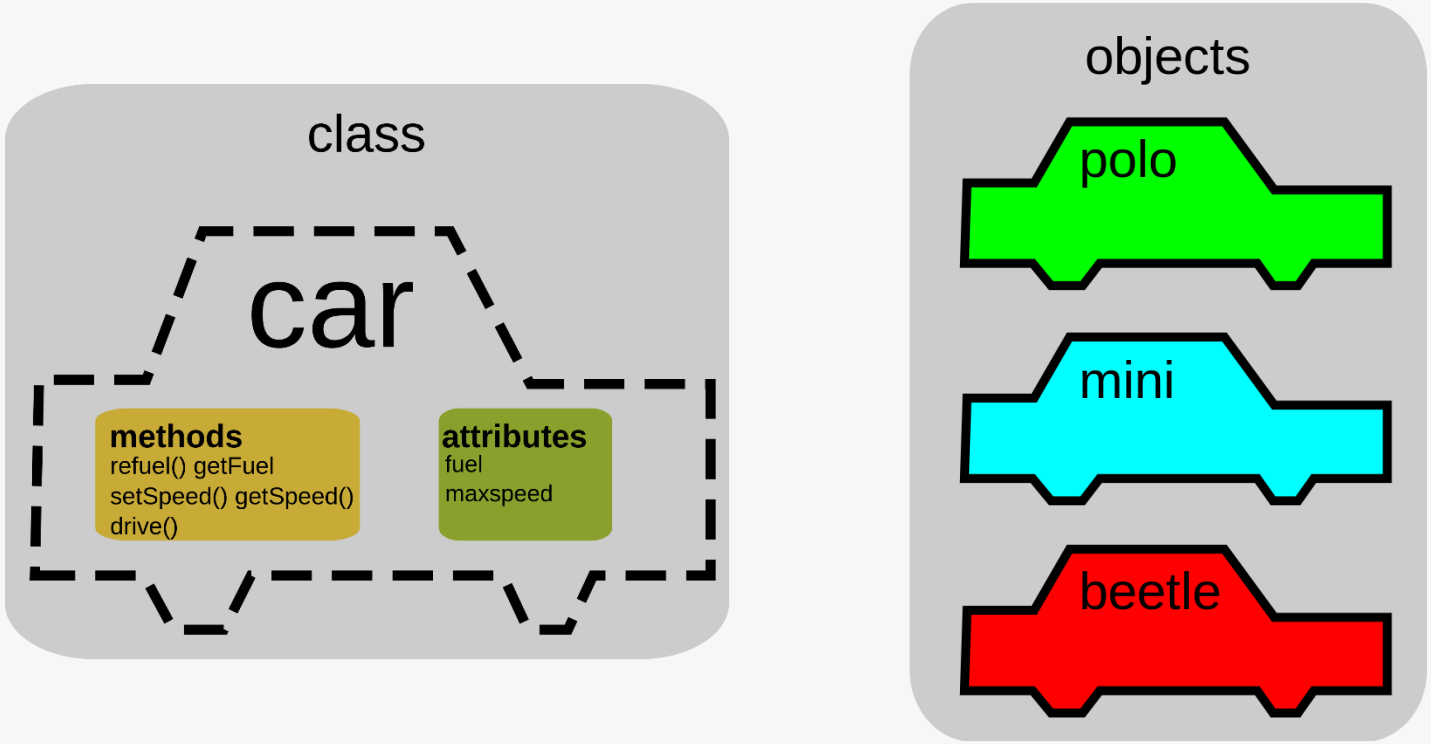
https://xkcd.com/303/


Compile
0 1 0 0
0 0 0 0
1 0 1 1
>hello
world


Interpret
>hello
world
Windows
Mac
Linux

What is Python

"Python is an interpreted, object-oriented, high-level programming language with dynamic semantics." – python.org

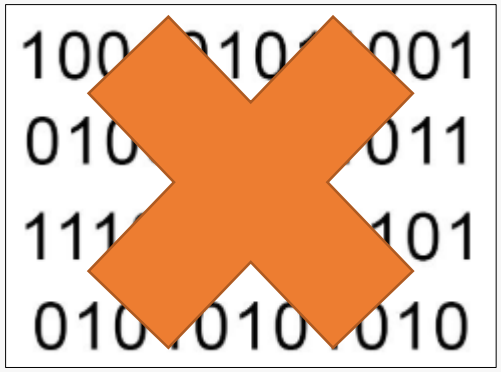

class
car
methods
refuel() getFuel
setSpeed() getSpeed()
drive()
attributes
fuel
maxspeed
objects
polo
mini
beetle

What is Python

"Python is an interpreted, object-oriented, high-level programming language with dynamic semantics." – python.org

Assembly Language

Machine Language



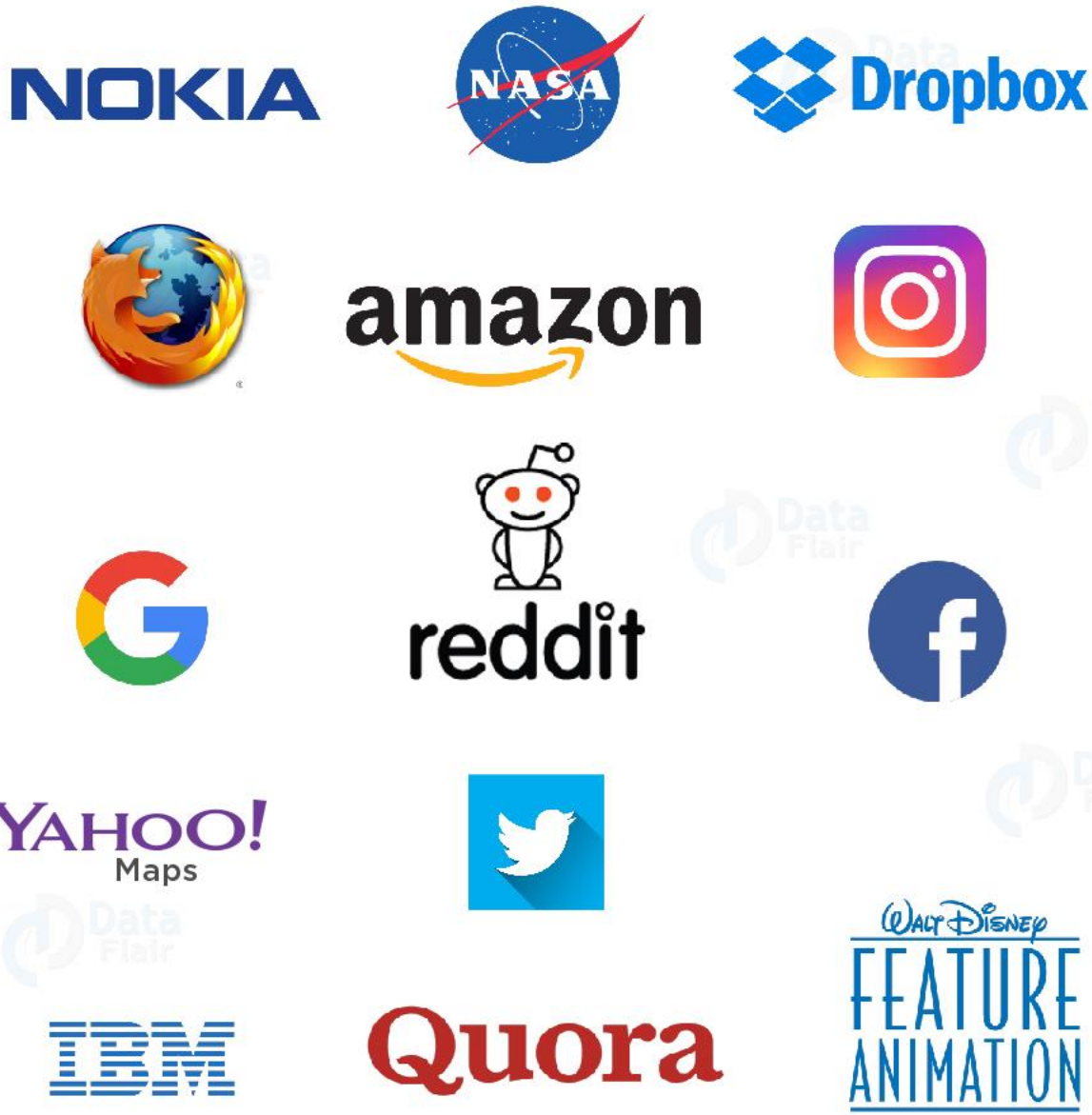
```
prep()
create_output_file()
text = get_text_from_file(filename='retired_asset.txt')

# SPLITS TEXT IN TO INDIVIDUAL FILES
split_word = 'SAMPLE COMPANY'
splitted_text = text.split(split_word)
```

```
import this
"""The Zen of Python, by Tim Peters. (poster by Joachim Jablon)"""

Beautiful is better than ugly.
Explicit is better than impl..
Simple is better than cомplεx.
Cомplεx is better than c0mp1|c@ted.
Flat is better than nested.
S p a r s e  is better than dense.
Readability counts.
Special cases aren't special enough to break the rules.
Although practicality beats purity.
raise PythonicError("Errors should never pass silently.")
# Unless explicitly silenced.
In the face of ambiguity, refuse the temptation to guess.
There should be one-- and preferably only one --obvious way to do it.
# Although that way may not be obvious at first unless you're Dutch.
Now is better than ...                    never.
Although never is often better thanrightnow.
If the implementation is hard to explain, it's a bad idea.
If the implementation is easy to explain, it may be a good idea.
Namespaces are one honking great idea -- let's do more of those!
```

Top Companies using Python



USE CASE I

LINK TO THE CODE

USE CASE II

LINK TO THE CODE

Q & A