why there are many IDEs in market?

Microsoft -> Visual Studio Code Jetbrains -> Pycharm

Data science -> Anaconda (software) -> Jupyter-notebook

Python -> IDLE

Front-end development - HTML, CSS, JAVASCRIPT, ReactJS, AngularJS, PHP

back-end development - Python, NodeJS, Golang, .net, etc...

Features of Python

- Easy to code
- Easy to read

  (psuedo code and python code can be easily matched)

- Free and open source (free to download, modify it freely and re-distribute freely)
- Robust library support (community) (Python has strong community of developers and there are multiple libraries for veriety of use cases)

  https://pypi.org/ (https://pypi.org/) - you will get count of libraries available in Python

- Interpreted (line by line code execution) (your code is executed line by line)

- Portable (same code can run on multiple platforms)
- Both Object-oriented and procedure-oriented programming are supported
- Exensible
- Supports GUI
- Dynamically typed (duck typing)

To run code in JAVA

- Step 1: write code
- Step 2: compile code
- Step 3: Run code

To run code in Python (Python Virtual Machine)

- Step 1: write code
- Step 2: run code

In [ ]:
```
file_1.java >>>>

    ```
    string name;
    name = "prashant"
    ```

javac file_1.java => compiled version [ERROR]
```

- Languages categorized into 2 types based on how you defined variables

1. Statically typed language lanugages where we declare our variable types before-hand
2. Dynamically typed language you can directly assign values to variables and type of that variable will be determined on run-time

In [ ]:
```
Basic data types

    - int     (1, 100, 0, 10000, 3000)
    - float   (20.4, 100.00, 0.5)
    - string  ("prashant", "velocity")
    - boolean (True, False)
```

Variable

- Variable is kind of a label
- Variable is something can be changed
- Variable is way of referring to a memory location. This particular memory location contains the value
- Variable is nothing but just a reference.
- You can think of variables as a "container" to contain some kind of a value
- value of varible may change during program execution

In [ ]:
```
English :

    words
        - verb
        - pronoun
        - adjectives
```

Programming language - keywords - built-in functions - variable - identifiers

keywords :: words reserved by the programming language

built-in functions :: - functions that are already defined in the language. - functions that are readily available inside programming language

Keywords

In [4]:
```
help("keywords")
```

```
Here is a list of the Python keywords.  Enter any keyword to get m
ore help.

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

built-in functions

- print
- type
- help

What is comment?

- comment is something written inside the code to describe itself.
- we write comments in the code for developers own understanding.

In [1]:
```
# checking feature `duck typing`

example = 100
type(example)  # `example` is a variable referring to `int` value
```

Out[1]:
```
int
```

In [2]:
```
example = "prashant"
type(example) # `example` is a variable referring to `str` value
```

Out[2]:
```
str
```

In [3]:
```
example = 0.50
type(example)  # `example` is a variable referring to `float` value
```

Out[3]:
```
float
```

Basic data types in python

- str 'prashant', "prashant", '''prashant''', """prashant"""
- int
- float
- bool

String: is nothing but set of characters.

In [5]:
```
type('100')
```

Out[5]:
```
str
```

In [ ]:
```
# c++

int c;    # first statement to declare varible name and its type
c = 100
```

In [14]:
```
name1 = "prashant"

name2 = "rahul"

name3 = "virat"

print(name1)
print(name2)
print(name3)
name3
```

```
prashant
rahul
virat
```

Out[14]:
```
'virat'
```