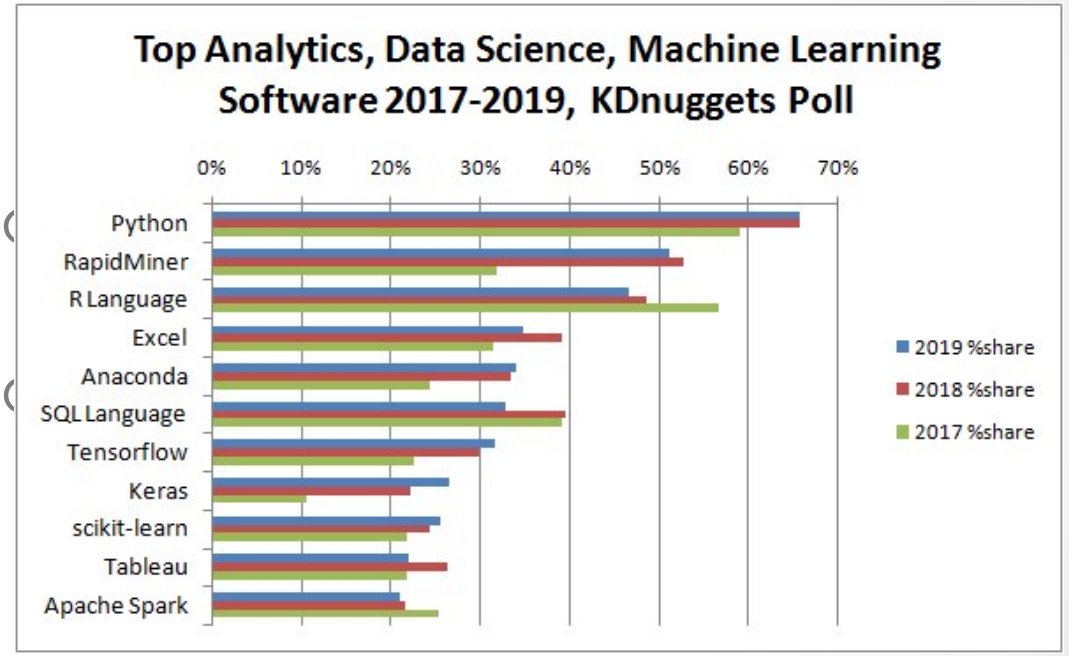
Python Basics

Installation and data types

Why Python?

- Free and open source
- Easier language to code
- Easier language to learn
- Platform independent



How to install Python (Windows/Mac)

- https://www.python.org/downloads/

How to install Python (Linux)

Run the following command in terminal for corresponding linux distribution

- *sudo apt-get install python (Debian/Ubuntu)*
- *sudo yum install python (RedHat/CentOS/Fedora)*

useful resources

(1) OnLine tutorial

http://www.learnpython.org/

(2) Book

Dive into Python

Not recommend

(3) Youtube videos

https://www.youtube.com/watch?v=_uQrJ0TkZlc

Any with a lot of views will do.

Input by definition

Let's try a few simple examples to show how simple python is.

Print something

(1) print('Hello World!')

(2) print("hello world!")

(3) Print('Hello World!')

------Case-sensitive,

-----'' and "" the same

------function name followed by ()

Input from the keyboard

```
input (‘what is your name? ‘)
name=input(‘what is your name? ‘)
course = input(‘which course are you taking? ‘)
print(name + ‘ is taking ‘ + course)

name = input('what is your name? ')
birth_year =input('what is your birth year? ')
age= 2019-birth_year
age= 2019 -int(birth_year)
print(age)

#wrong: print(‘age is ‘+age), have to be print(‘age is ‘+ str(age))
# to avoid the above error, you can use formatted string below, which format everything in the #large bracket into strings.
print(f'{name} is {age} old’)

print(name[1:-1])

name.upper()
```

String

Some common string functions for a toy string A:

- A[x], get the element at position x of A
- len(A), get the length of string A
- A.upper(), get all uppercase of string A
- A.lower(), get all lowercase of string A
- A.split(), split string A with specific separator.
- .join(Asplit), join the list of strings with the separator -> string
- A[x:y] the substring of A from x to y
- A.index('o'), A.count('.'), A.startswith(), A.endswith()

String examples

```
str1="I am in Florida, Central Florida."
x=5
print("the %s element in str1 is: " %(x))
print(str1[x])
print(astring[3:7])
print(astring[3:7:2])
```

```
str1.upper()
print(str1.upper())
print(str1)
str2 = str1.lower()
print(str2)
print(str1)
```

```
lenStr1=len(str1)
print("the length of the str1 is: ")
print(lenStr1)
print('the length of the str1 is: %s' %(len(str1))
print(str1.startswith("I am"))
print(str1.endswith("Florida"))
```

```
str3 = str1.split(',')
print('Split str1 by , ')
print(str3)
str3[0]
str3[1]
```

```
str4 =';'.join(str3)
print('Join the splited strings together with ; ')
print(str4)
str4[0]
str4[1]
```

Data types

Data type:

int e.g. 7

float e.g. 7.1

string “I am a string”

Boolean True and False

```
name = input('what is your name? ')
birth_year =input('what is your birth year? ')
age= 2023-birth_year
age= 2023 -int(birth_year)
print(age)
print(name[1:-1])
msg = f'{name} is {age} old'
print(msg)
```

operators

1. Arithmetic operators
   +, -, *, /, %, **, //
2. Assignment operators
   =, +=, -=, *=, /=, %=, **=, //=, etc
3. Comparison operators
   \>, <, >=, <=, ==, !=
4. Logical operators
   and, or, not
5. Membership operators
   in, not in
6. Bitwise operators
   &, |, ~, ^, <<, >>
7. Identity operators
   is, is not

Another way to print non-string variables

```
a = 4
b = 6

print(f"a = {a}, b = {b}")

a, b = 4, 6
c = d = 9

print(f"a + b = {a + b}")
print(f"a - b = {a - b}")
print(f"a * b = {a * b}")
print(f"a / b = {a / b}")
print(f"a % b = {a % b}")
```

Comments

- Comments are used to deactivate unnecessary code block or write short description to understand a complicated code block

```
print("hello python !!")

'''
name = "Jason"
age = 23
height = 173.54
'''
# printing variables, typecasting....
#print("Name : " + name + ", age : " + str(age) + ", height : " + str(height))

# printing variables with "format", without typecasting
#print("Name : {}, age : {}, height : {}".format(name, age, height))
```

Single line comment

Multi-line comment