Introduction

October 14, 2023

1 Introduction to python

Python syntax is more easy with the help of python career opportunity is more . python has more libraray 1 lakh 37 thoushand library in python

[1]:
```
## this is a intro section
```

[2]:
```
a = 10
```

[3]:
```
print (a)
```

```
10
```

[4]:
```
1+1
```

[4]: 2

[5]:
```
# this is a function for single line comment
a = 40
a
```

[5]: 40

[6]:
```
## this is a single line comment
```

[7]:
```
"""
Example of multiline comment
desgjakfjnjn

"""

a = 20
a
```

[7]: 20

[8]:
```
## numbers
1+3 # it showing the execution
```

[8]: 4

[9]:
```
print(1+3) # whenever we write a print it act as a output console
```

```
4
```

[10]:
```
a = 5
a
```

[10]: 5

[11]:
```
print ("hello world") # print is a inbuild function
```

```
hello world
```

[12]:
```
# variables assignment

name = "Priyanshu"
company = "wipro"
```

[13]:
```
number = 20
```

[14]:
```
number
```

[14]: 20

[15]:
```
decimal_num = 2.5
```

[16]:
```
decimal_num, type(decimal_num)
```

[16]: (2.5, float)

[17]:
```
1+2j
```

[17]: (1+2j)

[18]:
```
type(1+2j)
```

[18]: complex

[19]:
```
## variable assignment are case sensitive
company = 'Microsoft'
Company = 'Google'
print(company)
print(Company)
```

```
Microsoft
Google
```

[20]:
```
# Reserved keywords
'''
int, float, complex, bool, str, return, yield,
'''
```

[20]:
```
'\nint, float, complex, bool, str, return, yield,\n'
```

[21]:
```
## Boolean
bool(0)
```

[21]:
```
False
```

[22]:
```
bool(1)
```

[22]:
```
True
```

[23]:
```
str(23)
```

[23]:
```
'23'
```

[24]:
```
int('23') # converted into strings
```

[24]:
```
23
```

[25]:
```
type(int('23')) # check the type
```

[25]:
```
int
```

[26]:
```
a = 1
if bool (a) == True:
    print("True")
```

```
True
```

[27]:
```
# Dynamic typing language
# in other programming language we define the type of variables int, float,␣
 ↪bool, etc
# in python compiler on the run time will understand the type of variable

str1 = "Priyanshu"
a = "var"
```

[28]:
```
print(type(a))
```

```
<class 'str'>
```

[29]:
```
int(1.54)
```

[29]: 1

[30]:
```
## concatenating between different types
"1"*100
```

[30]: '1111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111'

[31]:
```
int("1")+int("1")
```

[31]: 2