In [ ]: In [1]:

float-float
string-str
boolean-bool
complex-complex

```
number=10
number
```

```
# In python we have some
basic data types
```

integer-int

Out[1]: 10

```
type(numbe
r)
```

In [2]:

Out[2]: int

binary
octa
hexa

bi mean two
it required only two digits
we have 0 1 2 3 4 5 6 7 8 9
to make binary we need only 0 and 1 example:
0b111, 0B111

In [3]:

```
0b111
```

Out[3]: 7

In [ ]: In [4]:

```
 4 2 1
 0 0 0 0 0 0 1 1 0 1 0 2 0 1 1 3 1 0 0 4 1 0 1 5 1 1 0 6 1 1 1 7
```

```
 0b1111
```

Out[4]: 15

octa mean eight
it required only 8 digits
we have 0 1 2 3 4 5 6 7 8 9
to make binary we need only
0 1 2 3 4 5 6 7 example:
0o776, 0O1234

In [5]:
```
0o123
```

```
��������
```

Out[5]: 83

```
h������
```

hexa mean 16
it required only 0 to 9 and A to F
we have 0 1 2 3 4 5 6 7 8 9 A B C D E F
example: 0x7A2, 0Xabc

In [8]:
```
#
16^0*1+16^1*(10)+
16^2(6)
0x6a1
```

Out[8]: 1697

```
��
```

```
��
```

```
��
```

```
��
```

```
��
```

In [11]:
```
type(numbe
r)
number=10.
56
```

Out[11]: float

```
                    # 10 is multilpying
                    with e1=10
In [12]:
10e+1

Out[12]: 100.0
                    # 10 is multiplying
                    with e2=100
In [13]:
10e+2

Out[13]: 1000.0
                     # 10 is multiplying
                     with e3=1000
In [14]:
10e+3

Out[14]: 10000.0
In [15]:        # 123.65x1000=
123.65e3        123650

Out[15]: 123650.0
                    10e-4 # 10/10000
                    10e+5 # 10*100000
In [16]:            12345e-2 #
10e-1 # 10/10       12345/100=123.45
10e-2 # 10/100
10e-3 # 10/1000

Out[16]: 123.45
                    # check the output
                    is zero
In [17]:
12345e-10

Out[17]: 1.2345e-06
```

10e2 as same as 10e+2 multiplying

10e-2 means dividing

```
          ❖❖❖❖❖❖❖❖❖❖❖❖❖
                integer #
In [ ]: In      variables  int
                # keyword
                "integer" #
                string

[21]:
                name="python"
                name


Out[21]: 'python'
          type(name
          )
In [22]:

Out[22]: str
```

```
            bad'
            type(name1)
In [24]:
name1='hydera

Out[24]: str
         'python'

 In [26]:

Out[26]: 'python'
                'python'")hello

In [29]:
print("hello     'python'

In [32]:
```

of a specific part of code Doc string

if some body using triple quote means the user try to say some thing about that code

```
                                """
In [ ]: In [33]:                i have performed addition
                                operation
                                here i have taken a value as 10
                                and b value as 20 """
print('hello "python"')         a=10
                                b=20
hello "python"                  c=a+b
                                c
```

triple quotes are used to write a story

```
Out[33]: 30
             "hello"+"pyt
             hon"
In [34]:

Out[34]: 'hellopython'
             'hello'+'pyt
             hon'
In [36]:

Out[36]: 'hellopython'
                """hello"""+"""p
                ython"""
In [37]:

Out[37]: 'hellopython'


In [38]: In


[ ]: In [ ]:
             In [39]:
             import
```

```
random       # str

random.randi ??????
             ??????
nt()         ??

             value=True
# int        type(value)
# float
```

```
Out[39]: bool
In [40]:   type(value
value1=Fal 1)
se
```

```
Out[40]: bool
            e"
In [42]:    type(value2
value2="Tru )
```

```
Out[42]: str
```

```
In [45]:
```

```
In [49]:
true=100
value3=true1
value3
```

```
---------------------------------------
-------------------------------- -
NameError Traceback (most recent call las
t)
Cell In[45], line 2
 1 true=100
----> 2 value3=true1
 3 value3

NameError: name 'true1' is not defined
```

```
In [ ]:
```

```
In [47]: In [48]:
```

```
#NameError: name 'true1' is not defined

#in the above lines check the name true1

is there or not true1=500
```

```
name=true1
```

```
                  
```

It represent as a+bj
where a is real number

```
Out[49]: complex
In [50]:  )
dir(value
```

```
Out[50]: ['__abs__',
          '__add__',
          '__bool__',
          '__class__',
          '__complex__',
          '__delattr__',
          '__dir__',
          '__doc__',
          '__eq__',
          '__format__',
          '__ge__',
          '__getattribute__',
          '__getnewargs__',
          '__getstate__',
          '__gt__',
          '__hash__',
          '__init__',
          '__init_subclass__',
          '__le__',
          '__lt__',
          '__mul__',
          '__ne__',
          '__neg__',
          '__new__',
          '__pos__',
          '__pow__',
          '__radd__',
          '__reduce__',
          '__reduce_ex__',
          '__repr__',
          '__rmul__',
          '__rpow__',
          '__rsub__',
          '__rtruediv__',
          '__setattr__',
          '__sizeof__',
          '__str__',
          '__sub__',
          '__subclasshook__',
          '__truediv__',
          'conjugate',
          'imag',
          'real']
```

b is an imagiznary number
ex: 3+4j

```
value=3+4j
type(value)
```

```
In [ ]: In [ ]:    #<package
                   name>.<method
                   name>
                   In [55]:
                   value=3+4j
                   dir(value)
# value is kind of # conjugate imag
package # m-1:     real
conjugate          value.conjugate()
# m-2: imag
# m-3:real

Out[55]: (3-4j)
                 value.real # real
                 number
In [56]:

Out[56]: 3.0
                   value.imag #
                   imaginary numbr
In [57]:

Out[57]: 4.0
                                          call las t)
                                          Cell In[60], line 1
In [60]:                                  ----> 1 value.img
value.imag # use tab

---------------------------------------   AttributeError: 'complex' object has no
---------------------------------- -      attribute 'img'
AttributeError Traceback (most recent
In [59]:      )
dir(value

Out[59]: ['__abs__',
          '__add__',
          '__bool__',
          '__class__',
          '__complex__',
          '__delattr__',
          '__dir__',
          '__doc__',
          '__eq__',
          '__format__',
          '__ge__',
          '__getattribute__',
          '__getnewargs__',
          '__getstate__',
          '__gt__',
          '__hash__',
          '__init__',
          '__init_subclass__',
          '__le__',
          '__lt__',
          '__mul__',
          '__ne__',
          '__neg__',
          '__new__',
          '__pos__',
```

```
'__pow__',
'__radd__',
'__reduce__',
'__reduce_ex__',
'__repr__',
'__rmul__',
'__rpow__',
'__rsub__',
'__rtruediv__',
'__setattr__',
'__sizeof__',
'__str__',
'__sub__',
'__subclasshook__',
'__truediv__',
'conjugate',
'imag',
'real']
```

syntax error ===== avoid
Name error: variable is not defined
Attribute error : module is not avialble

integer ==== int
float ===== float
string ==== str
boolean === bool
complex === complex

In [ ]:

In [ ]:

In [ ]:

In [ ]: