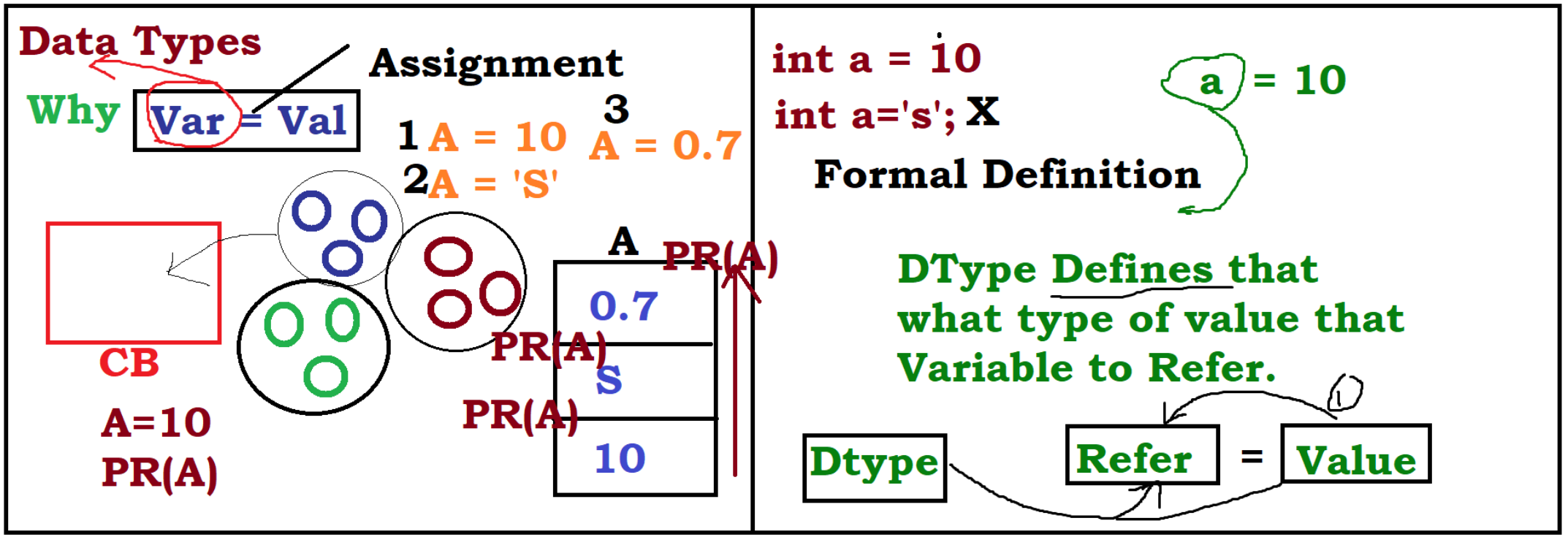
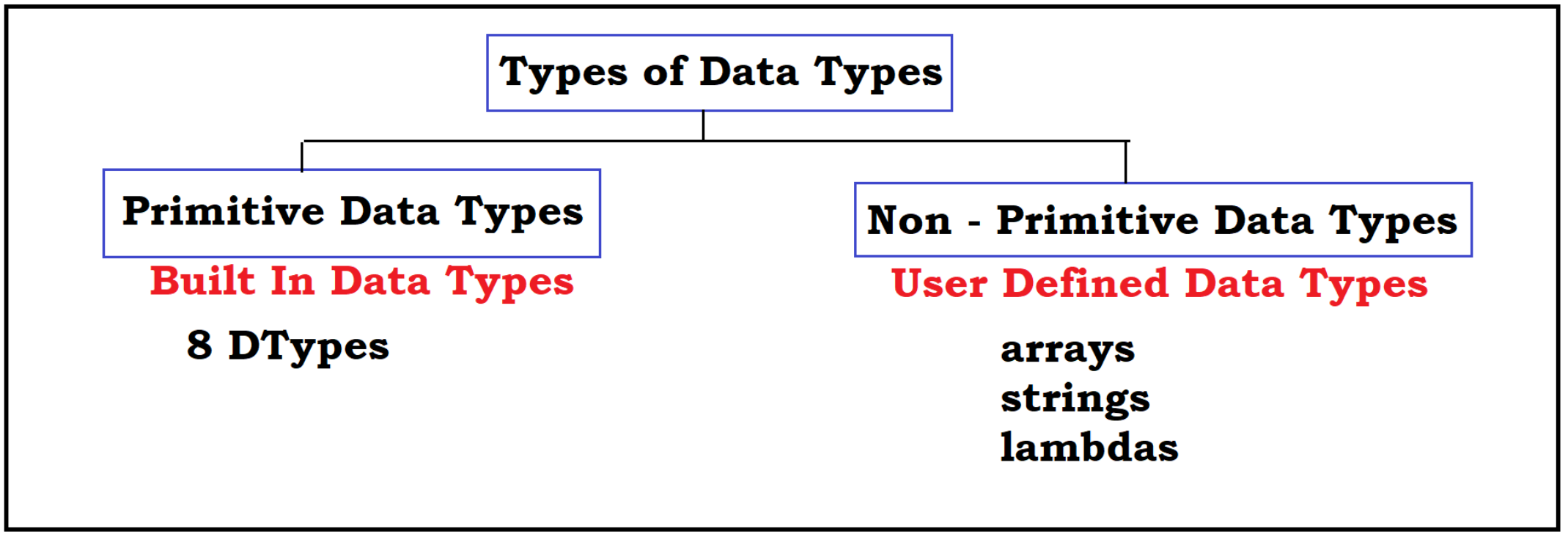

Data Types
Why
Var = Val
Assignment
1 A = 10
2 A = 'S'
3 A = 0.7
CB
A=10
PR(A)
A
PR(A)
0.7
PR(A)
S
PR(A)
10
int a = 10
int a='s'; X
Formal Definition
a = 10
DType Defines that what type of value that Variable to Refer.
Dtype
Refer
=
Value


Types of Data Types
Primitive Data Types
Built In Data Types
8 DTypes
Non - Primitive Data Types
User Defined Data Types
arrays
strings
lambdas

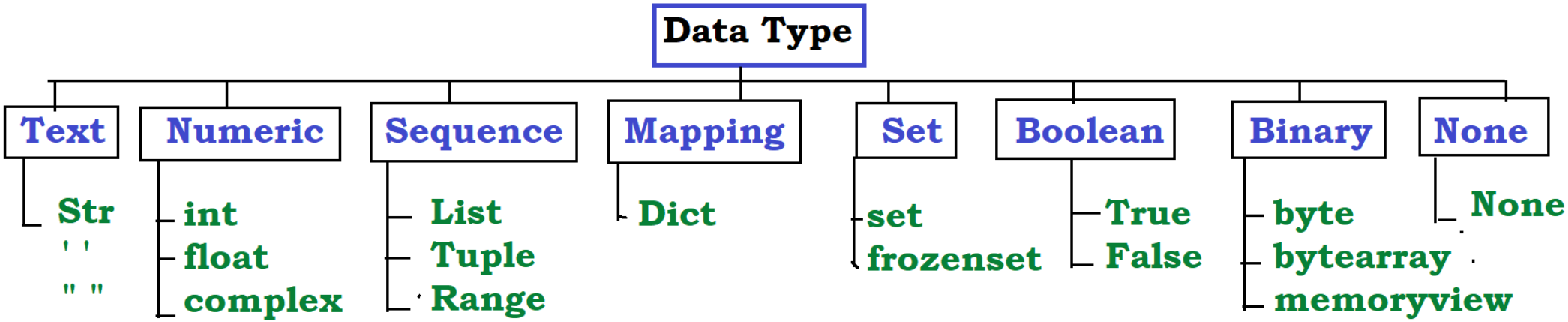
int a=10 x

How DType Works

```
a=10
print(a)
print(type(a))
#<class 'int'>
```

Dynamic Typing

Built-In Data Type 8 Data Types



Text Type

```
a="Ram"
b='C'
print(type(b))
#<class 'str'>
```

Numeric Type

```
a=10.78
print(type(a))
#<class 'float'>
print(int(a))#10
b=int(a)
print(type(a))
#<class 'float'>
print(type(b))
#<class 'int'>
```

Complex

Real + i Imaginary

a + ib **3+i4**

```
a=10+9j
print(a)
print(type(a))
#(10+9j)
#<class
'complex'>
```

int + Float + Complex

PYTHON NUMBERS

According to CSE Stream

26

$a + bj$

$a + jb$

$a + bi$

a	b	c	d	e
f	g	h	i	j
l	m	n	o	p
q	r	s	t	u
v	w	x	y	z

Iterable

Supporter

Complex

$x = 3 + 9j$

CN X

j 9

Sequence Type

Values are Arranged Side-By-Side in Continous Manner

```
a={'a':10,
   'b':20}
print(a)
print(type(a))
#{'a': 10, 'b': 20}
#<class 'dict'>
```

```
#[10, 20, 30]
#<class 'list'>
b=(10,20,30)
print(b)
print(type(b))
#(10, 20, 30)
#<class 'tuple'>
for i in range(0,5):
    print(i)
0
1
2
3
4
```

```
a=None
print(a)
print(type(a))
#None
#<class
'NoneType'>
```

```
a={10,20,30,40,50,60,70}
print(type(a))
#{50, 20, 70, 40, 10, 60, 30}
#<class 'set'>
```

```
a=True
print(a)
print(type(a))
#True
#<class 'bool'>
```

```
a=B'CS'#Byte B=Byte , b=bit
print(a)
print(type(a))
#b'CS'#b'CS'
#<class 'bytes'>
b=bytearray(5)#Allot Continous
Memory Allocation
print(b)
print(type(b))
#bytearray(b'\x00\x00\x00
\x00\x00')
#<class 'bytearray'>
c=memoryview(bytes(a))
print(c)
print(type(c))
#<memory at
0x0000020A8330F940>
#<class 'memoryview'>
```