PYTHON
Types

Types

type(11)	int
type(21.213)	float
type(“Hello Python 101”)	str

Types

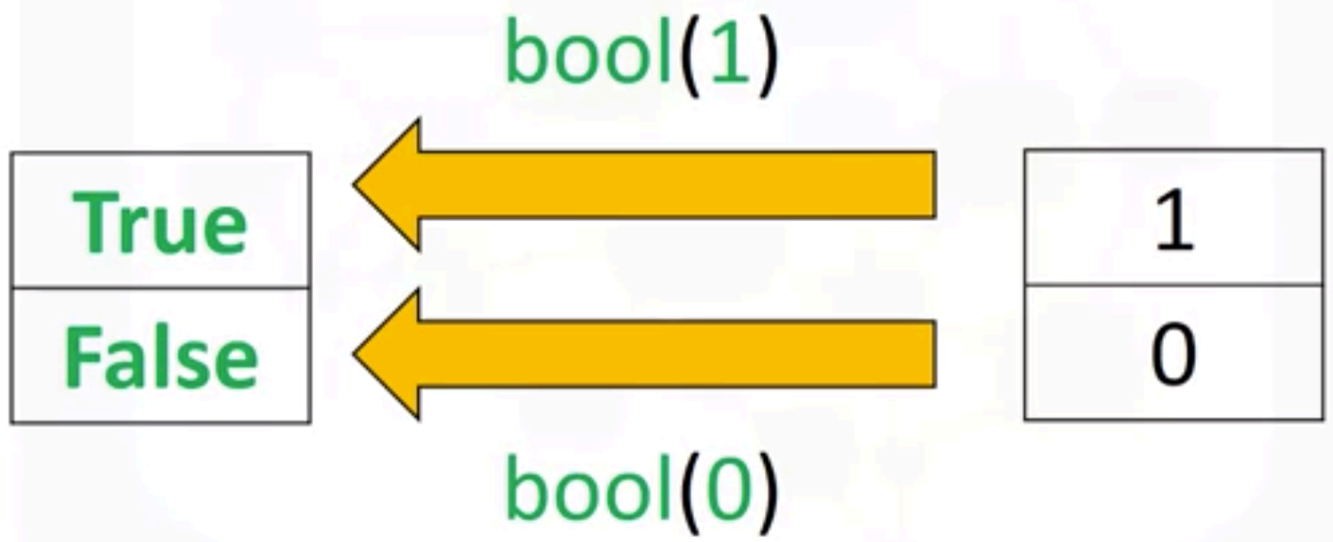
float(2):2.0

int(1.1):1

int('1'):1

int('A')

Types



PYTHON
Expressions and Variables

Expressions

Expressions: Mathematical Operations

```
25 // 5
```

5

```
25 // 6
```

4

```
25 / 6
```

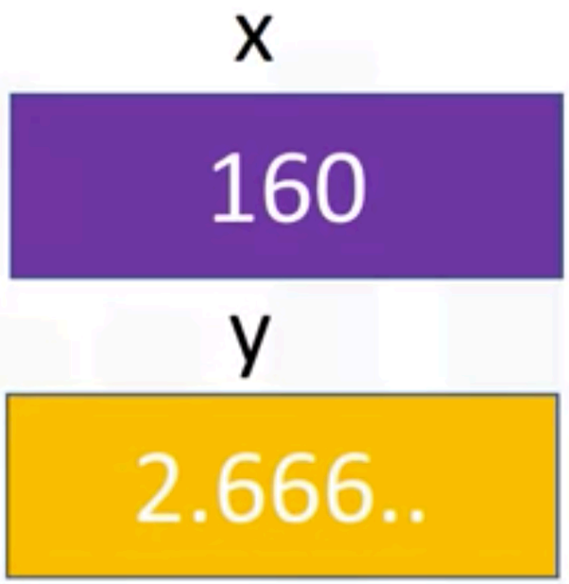
4.166..

Variables

x=160

y=x/60

y:2.6666..



Variables

total_min = 43 + 42 + 57

total_hr = total_min / 60

total_hr:2.367

PYTHON
Strings

STRINGS

Name= "Michael Jackson"

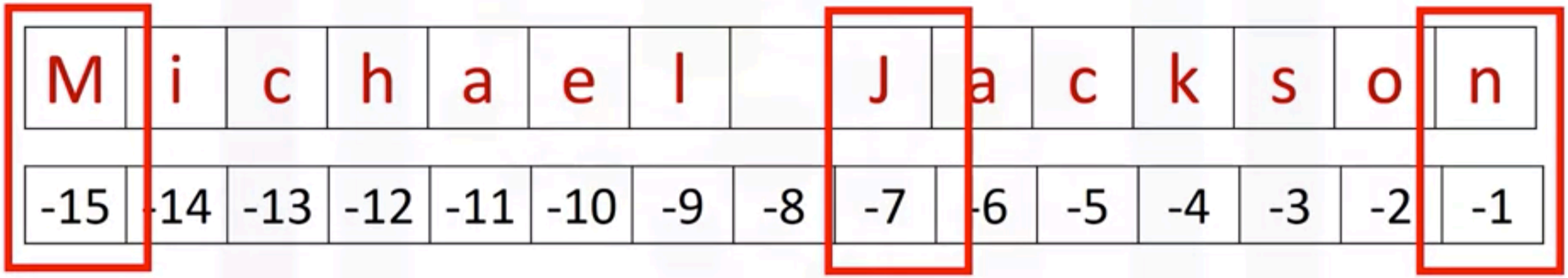
M	i	c	h	a	e	l		J	a	c	k	s	o	n
0	1	2	3	4	5	6	7	8	9	10	11	12	13	14

Name[0]:M

Name[6] :l

STRINGS

Name= "Michael Jackson"



Name[-15] :M Name[-7] :J Name[-1]: n

COGNITIVE CLASS.ai

STRINGS: Slicing

Name= "Michael Jackson"

M	i	c	h	a	e	l		J	a	c	k	s	o	n
0	1	2	3	4	5	6	7	8	9	10	11	12	13	14

Name[0:4] =Mich

Name[8:12] =Jack

STRINGS: Stride

Name= "Michael Jackson"

M	i	c	h	a	e	l		J	a	c	k	s	o	n
0	1	2	3	4	5	6	7	8	9	10	11	12	13	14

Name[::2]:"McalJcsn"

Name[0:5:2]:"Mca"

TUPLES: Slicing

len("Michael Jackson") =15

M	i	c	h	a	e	l		J	a	c	k	s	o	n
1	2	3	4	5	6	7	8	9	10	11	12	13	14	15

String

Name= “Michael Jackson”

Statement = Name +“ is the best”

Statement = “Michael Jackson is the best”

Tuples

3* "Michael Jackson "

"Michael Jackson Michael Jackson Michael Jackson "

COGNITIVE CLASS.ai

Strings: Immutable

Name= “Michael Jackson”

Name[0]=“J”

Name= Name+“ is the best”

Name: “Michael Jackson is the best”

Strings: escape sequences

- \ are meant to proceed escape sequences
- escape sequences are strings that are difficult to input

```
print(" Michael Jackson \n is the best" )
```

```
Michael Jackson
 is the best
```

Strings: escape sequences

```
print(" Michael Jackson \t is the best" )
```

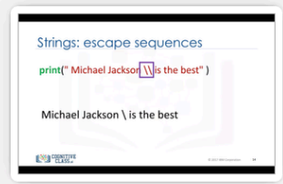
Michael Jackson		is the best

Strings: escape sequences

```
print(" Michael Jackson \\is the best" )
```

Michael Jackson \ is the best

String Methods



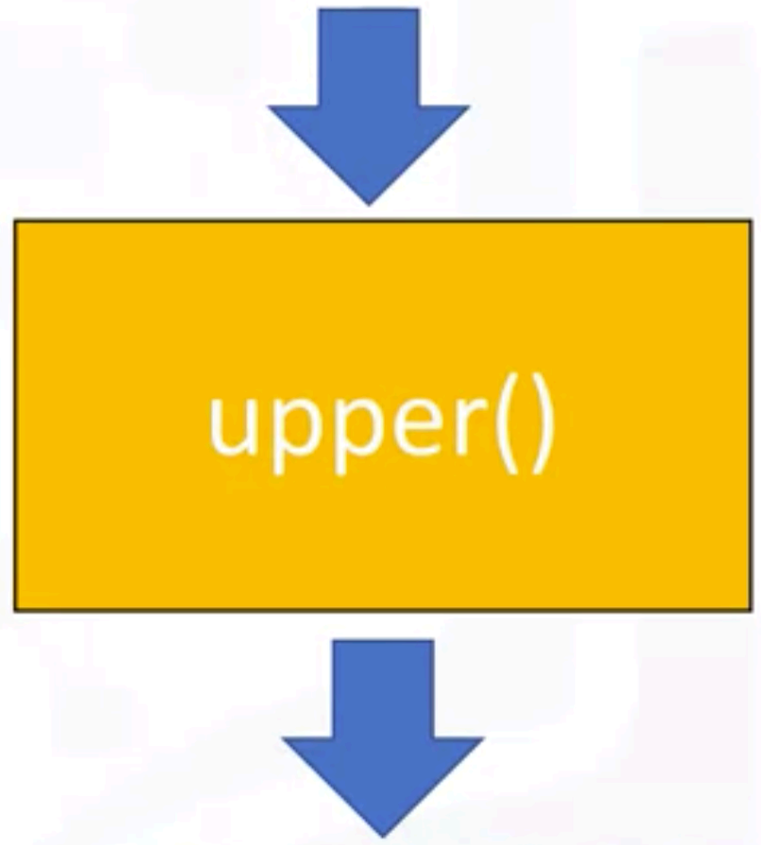
COGNITIVE CLASS.ai

Method

A="Thriller is the sixth studio album"

B=A.upper()

B:“THRILLER IS THE SIXTH STUDIO ALBUM”



Method

A='Michael Jackson is the best'

B=A.replace('Michael', 'Janet')

B: 'Janet Jackson is the best'

'Janet Jackson is the best'

COGNITIVE CLASS.ai

STRINGS: Stride

Name= "Michael Jackson"

M	i	c	h	a	e	l		J	a	c	k	s	o	n
0	1	2	3	4	5	6	7	8	9	10	11	12	13	14

Name.find('el'):5

Name.find('Jack'):8

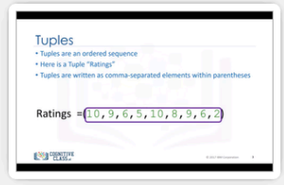
PYTHON
Tuples and Lists

Tuples

Tuples

- Tuples are an ordered sequence
- Here is a Tuple “Ratings”
- Tuples are written as comma-separated elements within parentheses

```
Ratings =(10,9,6,5,10,8,9,6,2)
```



Tuples

```
Tuple1 =("disco",10,1.2)
```

0	"disco"
1	10
2	1.2

Tuple1[0]: "disco"

Tuple1[1]: 10

Tuple1[2]: 1.2

Tuples

```
("disco",10,1.2)
```

```
tuple2 = tuple1 + ("hard rock",10)

("disco",10,1.2, "hard rock",10)
```

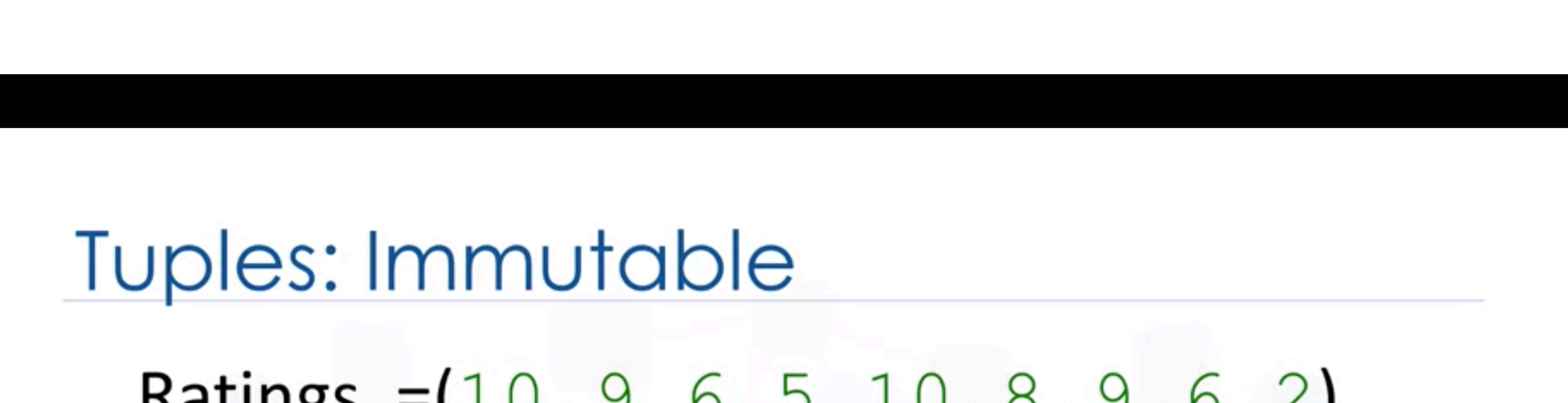
Tuples: Slicing

("disco", 10, 1.2, "hard rock", 10)

tuple2[0:3] : ('disco', 10, 1.2)

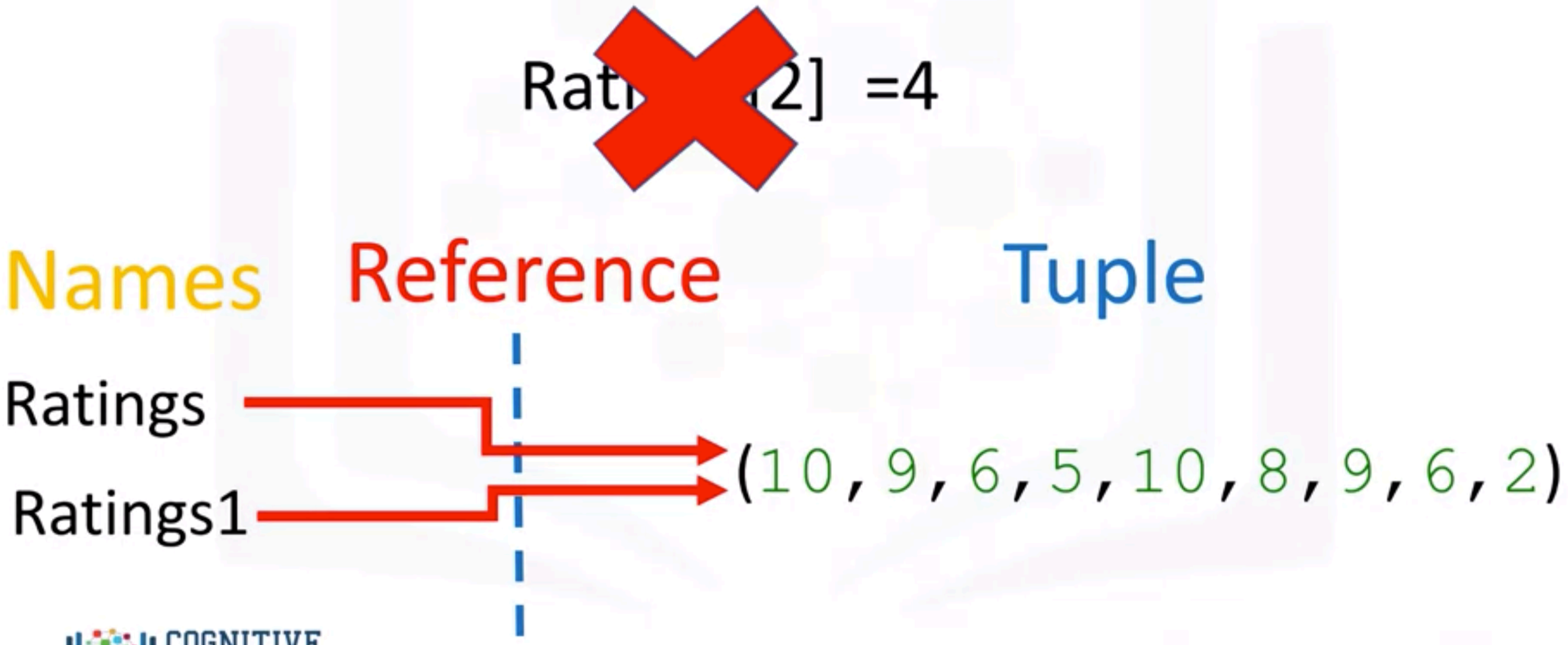
Tuples: Immutable

```
Ratings =(10,9,6,5,10,8,9,6,2)
Ratings1=Ratings
```



COGNITIVE CLASS.ai

Tuples: Immutable



COGNITIVE CLASS.ai

Tuples: Immutable

```
Ratings =(10,9,6,5,10,8,9,6,2)
```

```
RatingsSorted=sorted(Ratings)
```

```
(2, 5, 6, 6, 8, 9, 9, 10, 10)
```

COGNITIVE CLASS.ai

Tuples: Nesting

NT =(1, 2, ("pop", "rock") ,(3,4),("disco",(1,2)))

0	1	2	3	4

NT[2]: ("pop", "rock") [1] =“rock”

0	1

NT[2] [1] =“rock”

Lists

- Lists are also ordered sequences
- Here is a List “L”
- A List is represented with square brackets
- List mutable

```
L =["Michael Jackson", 10.1,1982]
```

Lists

```
["Michael Jackson", 10.1,1982,[1,2],('A',1)]
```

Lists

```
L =["Michael Jackson", 10.1,1982]
```

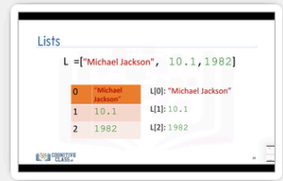
-3	0	“Michael Jackson”
-2	1	10.1
-1	2	1982

L[-3]: “Michael Jackson”

L[-2]: 10.1

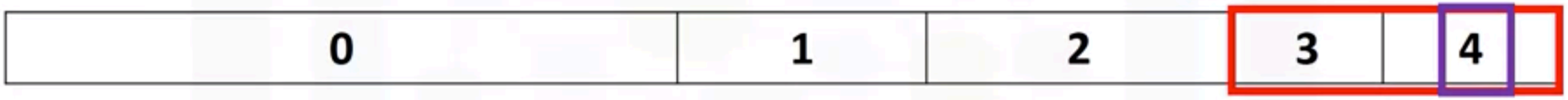
L[-1]: 1982

COGNITIVE CLASS.ai



Lists: Slicing

L =["Michael Jackson", 10.1,1982, “MJ”, 1]



L[3:5]:[“MJ”, 1]

Lists

```
L=["Michael Jackson", 10.1,1982]
```

```
L.extend(["pop",10])
```

```
L=["Michael Jackson", 10.1,1982, "pop",10]
```

0	1	2	3	4

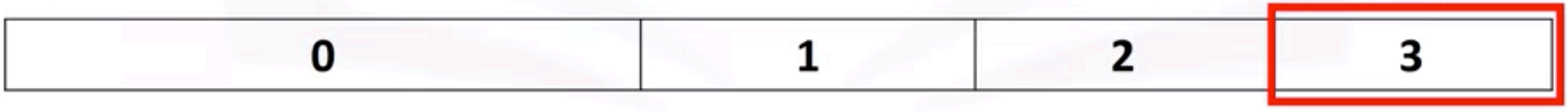
COGNITIVE CLASS.ai

Lists

```
L=["Michael Jackson", 10.1,1982]
```

```
L.append (["pop",10])
```

```
L=["Michael Jackson", 10.1,1982, ["pop",10]]
```

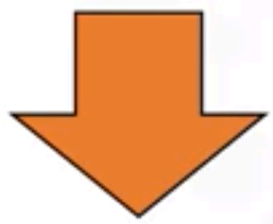


COGNITIVE CLASS.ai

Lists

```
A=["hard rock",10,1.2]
```

```
del(A[0])
```



```
A:[10,1.2]
```

Convert String to List

"hard rock".split()

["hard", "rock"]

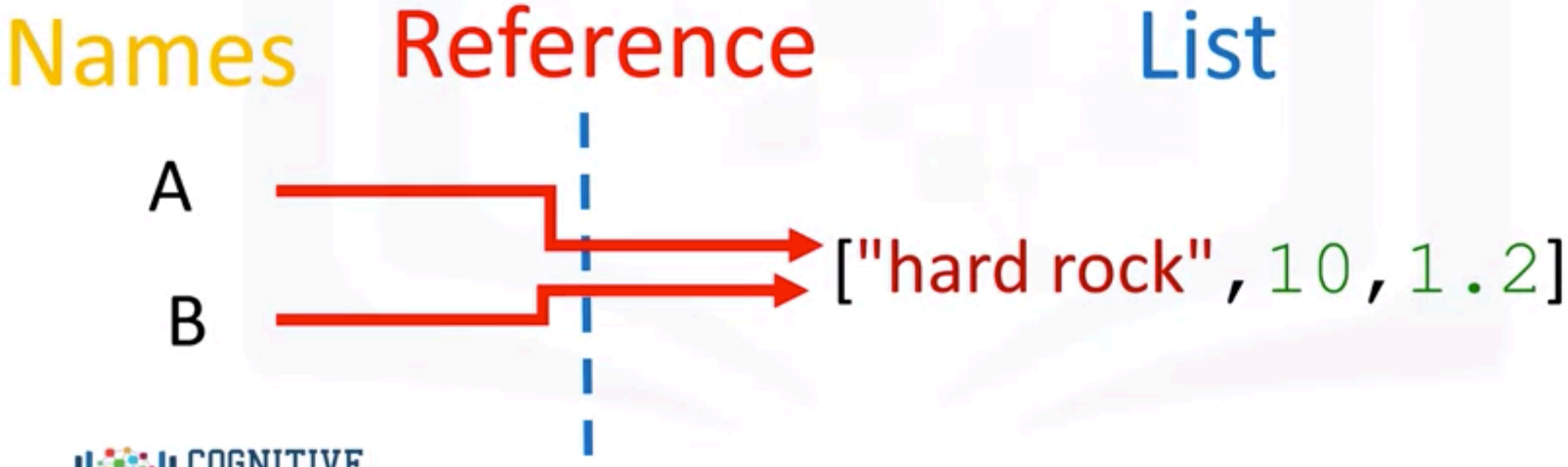
Lists

"A,B,C,D".split(",")

["A", "B", "C", "D"]

Lists: Aliasing

```
A=["hard rock",10,1.2]
B=A
```

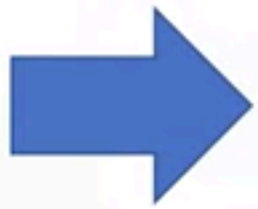


COGNITIVE CLASS.ai

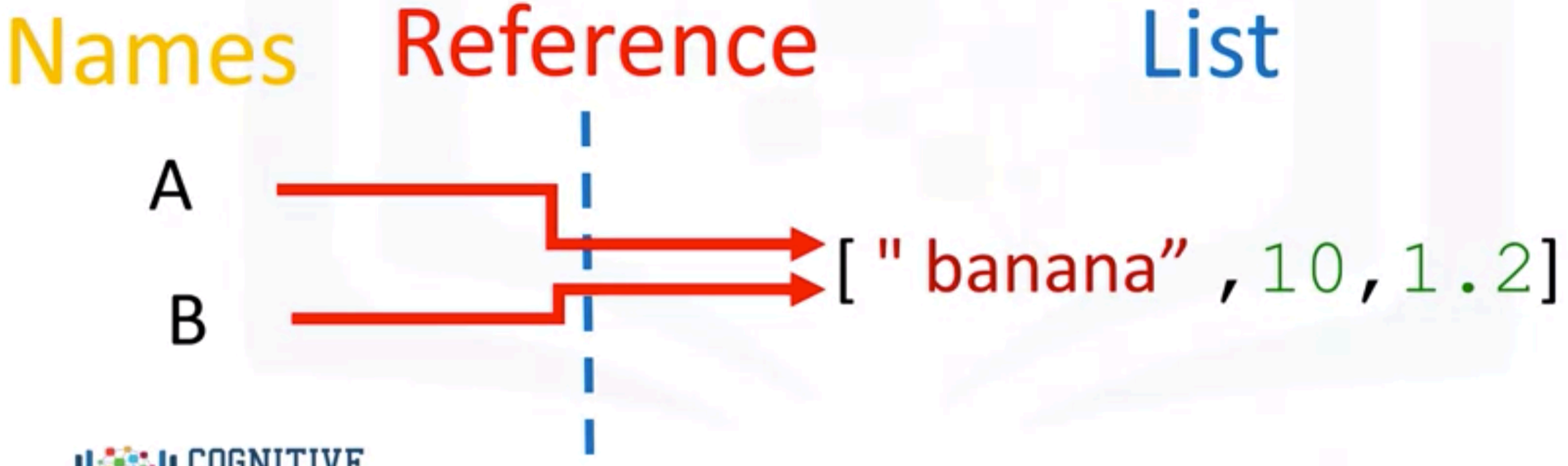
Lists: Aliasing

B[0]="hard rock”

A[0]=" banana ”



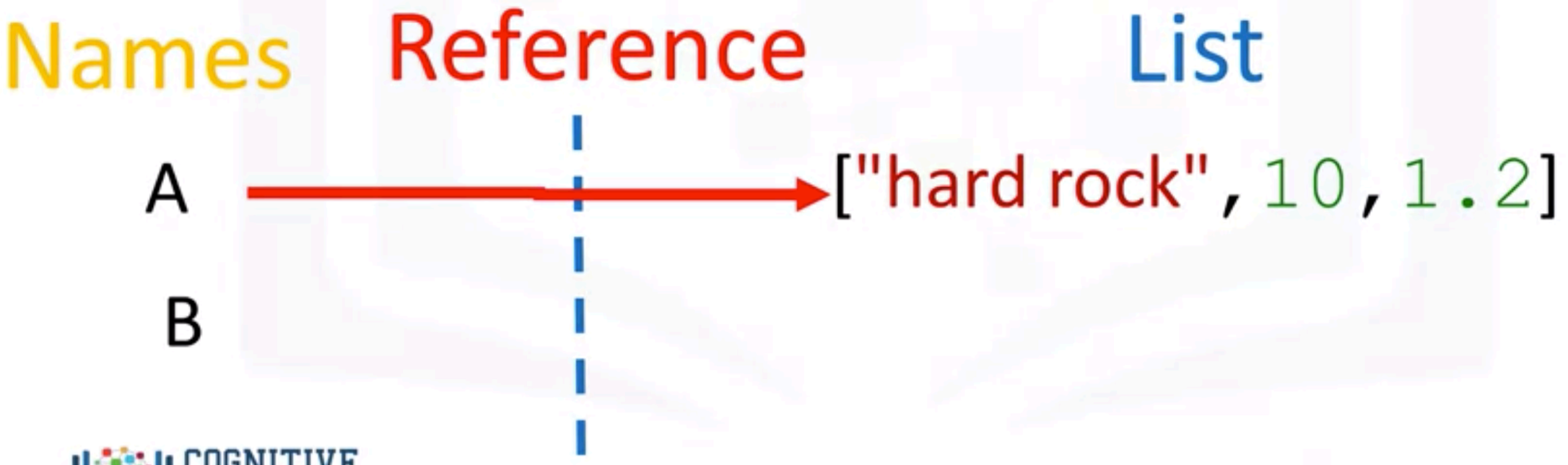
B[0]: " banana ”



COGNITIVE CLASS.ai

Lists: Clone

```
A=["hard rock",10,1.2]
B=A[:]
```

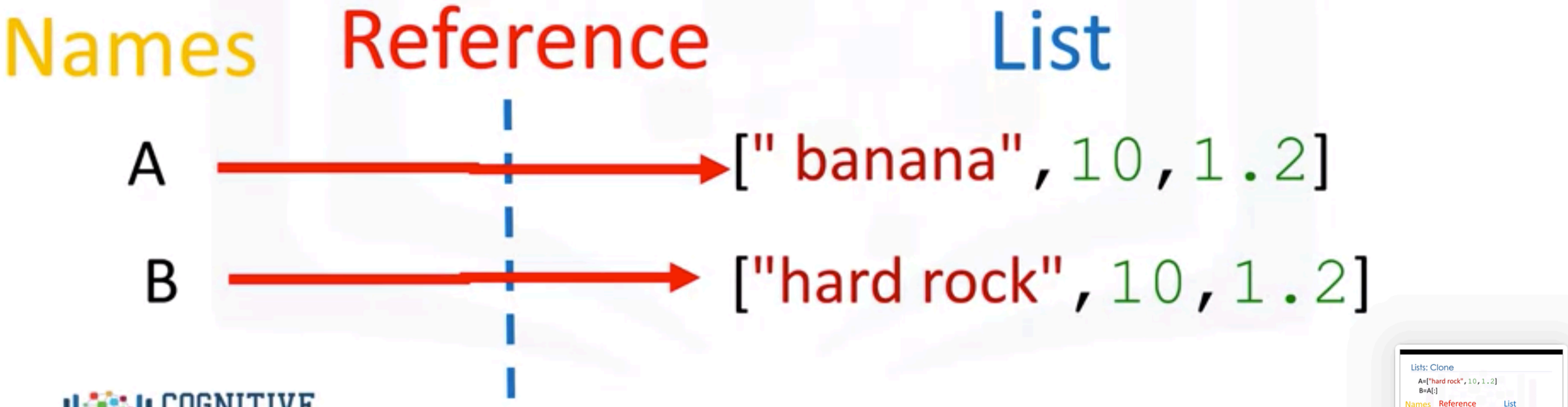


Lists: Clone

```
A=["hard rock",10,1.2]
```

A[0]= " banana "

B[0]: " hard rock "



COGNITIVE CLASS.ai

Lists

```
A=["hard rock",10,1.2]
```

```
help(A)
```