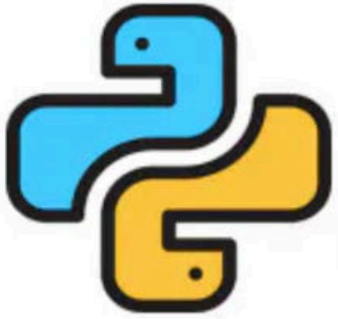
unacademy

List, Set, Tuple and Dictionary

Course on Data Structure and Algorithms Using Python

Ankush Saklecha • Lesson 4 • Dec 11, 2023

unacademy



Python Programming

String Manipulations

unacademy

Content

- String
- String Indexing & Slicing
- String Methods
- Sample Problems

unacademy

String

- In python, a string is defined as single character or group of characters.

  It is denoted as single quote ‘’ or double quote “”.

String Examples:

```
Name=‘Jhon’
Qualification=“Master of Technology”
print(Name)
print(Qualification)
print(‘Welcome to python string’)
print(“Hello STUDENTS…”)
```

unacademy

Multiline String

- String may be defined in multi-lines.

Multiline String Examples:

```
MSG = """Python is high level, object oriented programming. It  has many
advanced features such as library for machine and deep learning, IoT, Data
Science and many more"""
print(MSG)
```

unacademy

String Indexing & Slicing

- Like an array, we can also define indexes to characters of string.
- Python support two types of indexing:

  --Positive Indexing

  --Negative Indexing

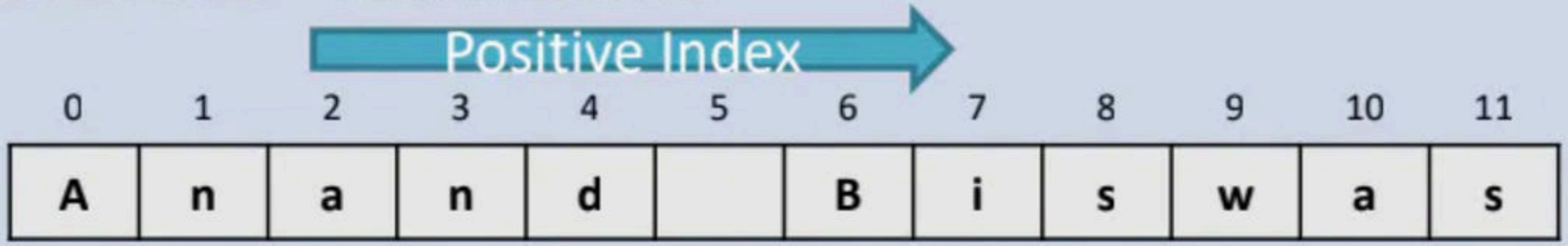
unacademy

Positive Indexing

- It begin with zero and end with string length-1 from left hand side.

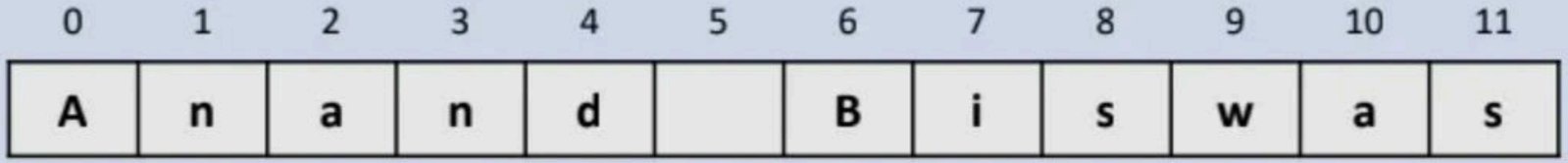
String Positive Indexing Examples:

StudentName="Anand Biswas"



Access String Character via Index:

StudentName="Anand Biswas"



print(StudentName[2])

unacademy

Let's Try…

Guess Output?

```
EmpName="Jhonson Saw"

print(EmpName[2+3])

print(EmpName[10/3])

print(EmpName[10//3])

print(EmpName[2**2])

print(EmpName[2>>1])

print(EmpName[2&4])

print(EmpName[6^3])
```

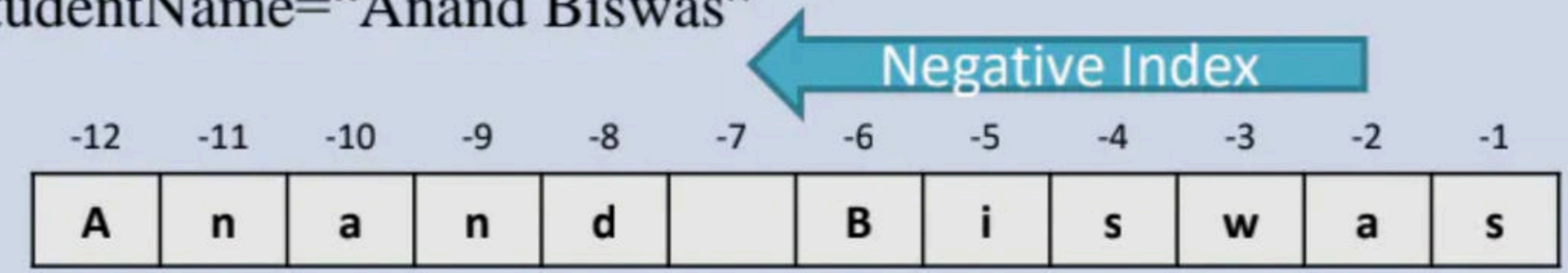
unacademy

Negative Indexing

- It begin with -1 and end with string length in negative from right hand

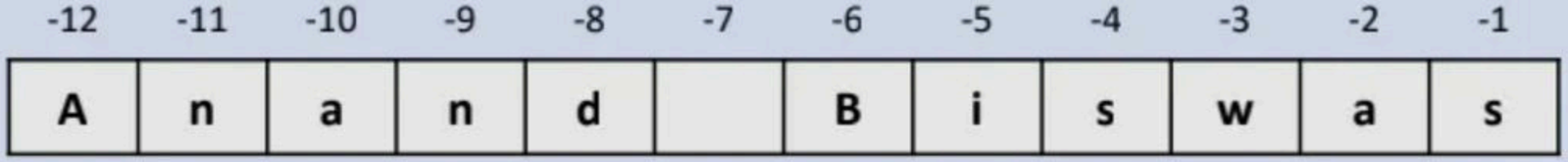
String Negative Indexing Examples:

StudentName="Anand Biswas"



Access String Character via Negative Index:

StudentName="Anand Biswas"



print(StudentName[-2])

unacademy

Let's Try…

Guess Output?

College="Indian Institute of Technology"

print(College[-4])

print(College[-6-4])

print(College[-9+10])

print(College[14-9])

print(College[-8-3])

print(College[-4+4])

print(College[-12+6])

unacademy

String Slicing

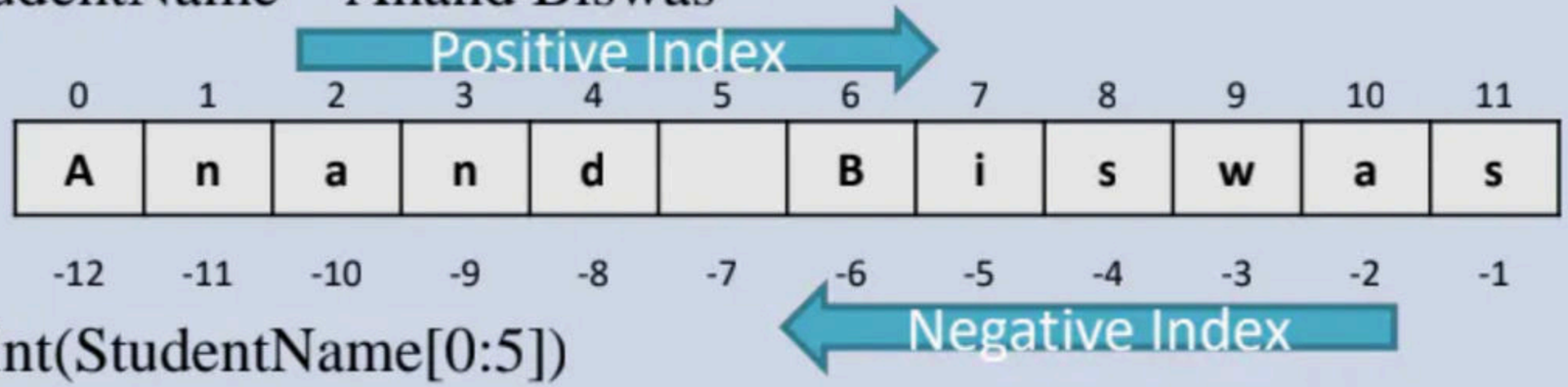
- It is way to access set of characters or substring from the given string. It is denotes as:

string_name[startindex : endindex]

- It slice string from startindex and end with endindex-1 i.e. it excludes endindex value. Slice index may be positive or negative range.

String Slicing Examples:

StudentName="Anand Biswas"




```
print(StudentName[0:5])

print(StudentName[-7:-2])
```

unacademy

Let's Try…

Guess Output?

```
College="Indian Institute of Technology"

print(College[1+1:4+0])

print(College[0:2**3])

print(College[-12:-7+3])

print(College[-1:-7)

print(College[:10])

print(College[:-6])

print(College[2:-2])

print(College[-8:14])
```

unacademy

Concatenate String

- In python, we can merge or combine two or more string using + operator.

String Concatenate Examples:

```
Fname="Anand"

Lname="Biswas"

print(Fname+Lname)

print(Fname+" "+ Lname)
```

unacademy

Let's Try…

Guess Output?

```
Fname="Anand"
Lname="Biswas"

print(Fname+ 5+ Lname)

print(Fname+ '9'+ Lname)

print(Fname+ 7.8+ Lname)

print(Fname+ .+ Lname)

print(Fname++ Lname)
```

unacademy

String Format

- In python, with the help of format() method we can concatenate any number with the string.

String Format Examples:

```
FName="Anand"
Lname="Biswas"
Age=36
print(Fname+Lname+36)
Name=Fname+Lname+" is {} year old"
print(Name.format(Age))
```

unacademy

String Format

String Format Examples:

Fname="Anand"

Lname="Biswas"

Year=36

Month=5

Name=Fname+Lname+" is {} year and {} month old"

print(Name.format(Age,Month))

unacademy

String Methods

In python, several built-in methods are available.

Method Name	Description
capitalize()	Converts the first character to upper case
casefold()	Converts string into lower case
center()	Returns a centered string
count()	Returns the number of times a specified value occurs in a string
encode()	Returns an encoded version of the string
endswith()	Returns true if the string ends with the specified value
expandtabs()	Sets the tab size of the string
find()	Searches the string for a specified value and returns the position of where it was found
format()	Formats specified values in a string
format_map()	Formats specified values in a string
index()	Searches the string for a specified value and returns the position of where it was found

unacademy

String Methods

Method Name	Description
isalnum()	Returns True if all characters in the string are alphanumeric
isalpha()	Returns True if all characters in the string are in the alphabet
isdecimal()	Returns True if all characters in the string are decimals
isdigit()	Returns True if all characters in the string are digits
isidentifier()	Returns True if the string is an identifier
islower()	Returns True if all characters in the string are lower case
isnumeric()	Returns True if all characters in the string are numeric
isprintable()	Returns True if all characters in the string are printable
isspace()	Returns True if all characters in the string are whitespaces
istitle()	Returns True if the string follows the rules of a title

66017

unacademy

String Methods

Method Name	Description
isupper()	Returns True if all characters in the string are upper case
join()	Joins the elements of an iterable to the end of the string
ljust()	Returns a left justified version of the string
lower()	Converts a string into lower case
lstrip()	Returns a left trim version of the string
maketrans()	Returns a translation table to be used in translations
partition()	Returns a tuple where the string is parted into three parts
replace()	Returns a string where a specified value is replaced with a specified value
rfind()	Searches the string for a specified value and returns the last position of where it was found
rindex()	Searches the string for a specified value and returns the last position of where it was found

unacademy

String Methods

Method Name	Description
rjust()	Returns a right justified version of the string
rpartition()	Returns a tuple where the string is parted into three parts
rsplit()	Splits the string at the specified separator, and returns a list
rstrip()	Returns a right trim version of the string
split()	Splits the string at the specified separator, and returns a list
splitlines()	Splits the string at line breaks and returns a list
startswith()	Returns true if the string starts with the specified value
strip()	Returns a trimmed version of the string
swapcase()	Swaps cases, lower case becomes upper case and vice versa
title()	Converts the first character of each word to upper case

unacademy

String Methods

capitalize( ):

```
Course="btech"

print(Course.capitalize( ))
```

casefold( ):

```
Course="BTECH"

print(Course.casefold( ))
```

center( ):

```
Course="btech"

print(Course.center(10))
```

unacademy

String Methods

count( ):

```
Course="btech in cse"
print(Course.count('e'))
print(Course.count('e',4,10))
```

encode( ):

```
Course="BTECH"
print(Course.encode( ))
```

endswith( ):

```
Course="btech"
print(Course.endswith("." ))
print(Course.endswith(".",2,7 ))
```

unacademy

Let's Try…

Guess Output?

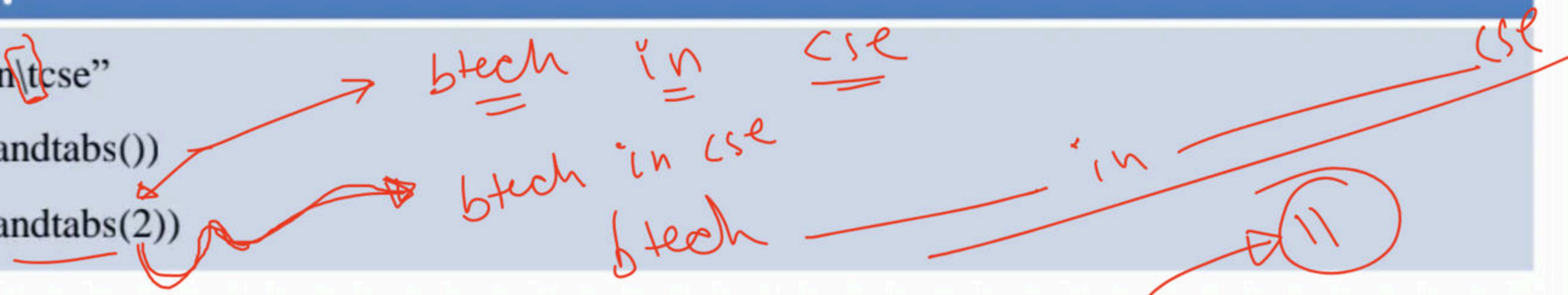
```
str="hELLO"
print(str.capitalize())

str="hELLo"
print(str.casefold())

str="hELLo how are you"
print(str.center(-4))

str="hELLo how are you"
print(str.count('LL',1,12))

str="hELLo how are you"
print(str.count('o',-11,-2))

str="hELLo how are you!"
print(str.endswith('.'))
```

unacademy

String Methods

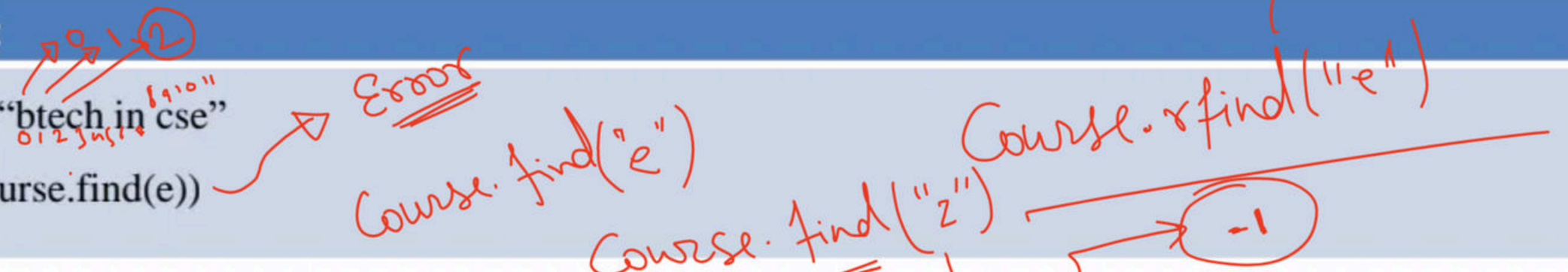
expandtabs( ):

```
Course="btech\tin\tcse"
print(Course.expandtabs())
print(Course.expandtabs(2))
```



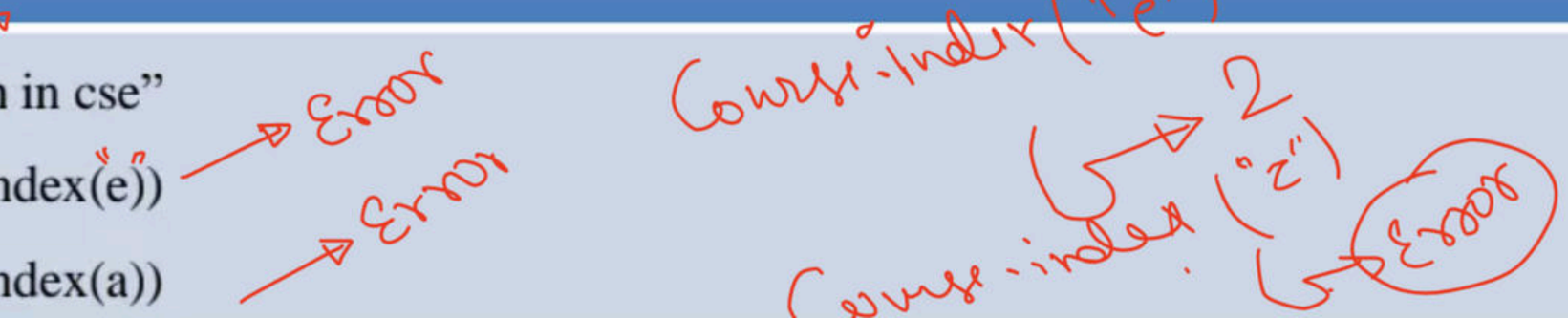
find( ):

```
Course="btech in cse"
print(Course.find(e))
```



index( ):

```
Course="btech in cse"
print(Course.index("e"))
print(Course.index(a))
```



unacademy

String Methods

isalnum( ):

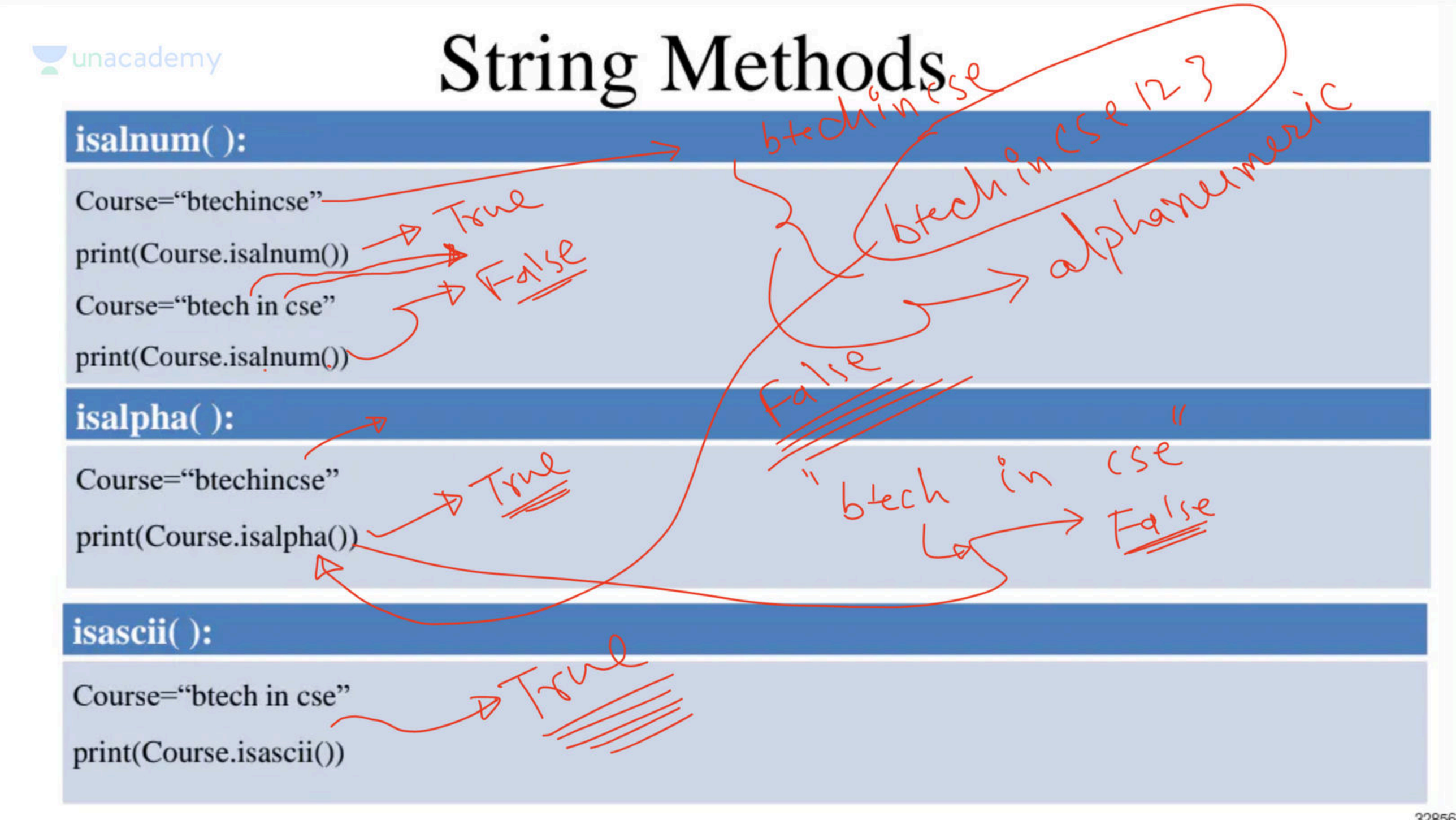
```
Course="btechincse"
print(Course.isalnum())
Course="btech in cse"
print(Course.isalnum())
```

isalpha( ):

```
Course="btechincse"
print(Course.isalpha())
```

isascii( ):

```
Course="btech in cse"
print(Course.isascii())
```



Let's Try…

Guess Output?

```
str="hELLo\tHOW\tare\tyou!"
print(str.expandtabs(10.5))

str="hELLo\tHOW\tare\tyou!“
print(str.find('o' and 'O'))

str="hELLo\tHOW\tare\tyou!“
print(str.index('T'))

str="hELLo HOW are you 9"
print(str.isalnum())

str="hELLoHOWareyou910"
print(str.isalpha())

str="hELLo HOW are you 9"
print(str.isascii())
```

unacademy

String Methods

isdecimal( ):

number="1234"

print(number.isdecimal())

number="a1234"

print(number.isdecimal())

isdigit( ):

number="102346"

print(number.isdigit())

isdentifier( ):

identifier="a b c"

print(identifier.isidentifier())

unacademy

$-abc$ → true

unacademy

String Methods

islower( ):

```
str="hello"
print(str.islower())
str="Hello"
print(str.islower())
```

True

False

```
number = 10
Print (number.isnumeric())
```

→ Error

isnumeric( ):

```
number="10"
print(number.isnumeric())
```

True

isprintable( ):

```
str="a b c"
print(str.isprintable())
```

True

unacademy

Let's Try...

Guess Output?

```
num='6.6'
print(num.isdecimal())

num='1 2 3 4 5 6'
print(num.isdigit())

ident='_abcx123'
print(ident.isidentifier())

str='helLo'
print(str.islower())

num='45667'
print(num.isnumeric())

str="Hello\tHi"
print(str.isprintable())
```

unacademy

Sample Problems

Problem:

Write a python code to check and display valid and invalid identifier names given from the user.

Example

Sample Input- var1 _xvalue abc%xy PI CircleRaduis 67abx

Sample Output-

Valid Identifier- var1 _xvalue PI CircleRaduis

Invalid Identifier- abc%xy 67abx

Solution- https://colab.research.google.com/drive/1swpewE109qQdJW7zxn56HvfDOIxxWd_6

unacademy

String Methods

isspace( ):

str="hello hru"

print(str.isspace())

→ False

str="HelloHRU"

print(str.isspace())

→ false

Str = " "

Print(str.isspace())

→ True

istitle( ):

str="Hello Hru"

print(str.istitle())

→ True

isupper( ):

str="HELLO"

print(str.isupper())

→ True

unacademy

String Methods

join( ):

```
str="hello"
str1="-".join(str)
print(str1)
```

ljust( ):

```
str="Hello"
print(str.ljust(6), "HRU")
```

lower( ):

```
str="HELLO"
print(str.lower())
```

unacademy

Print (Str . ljust (10) * ' Hru')

hello - - - - -

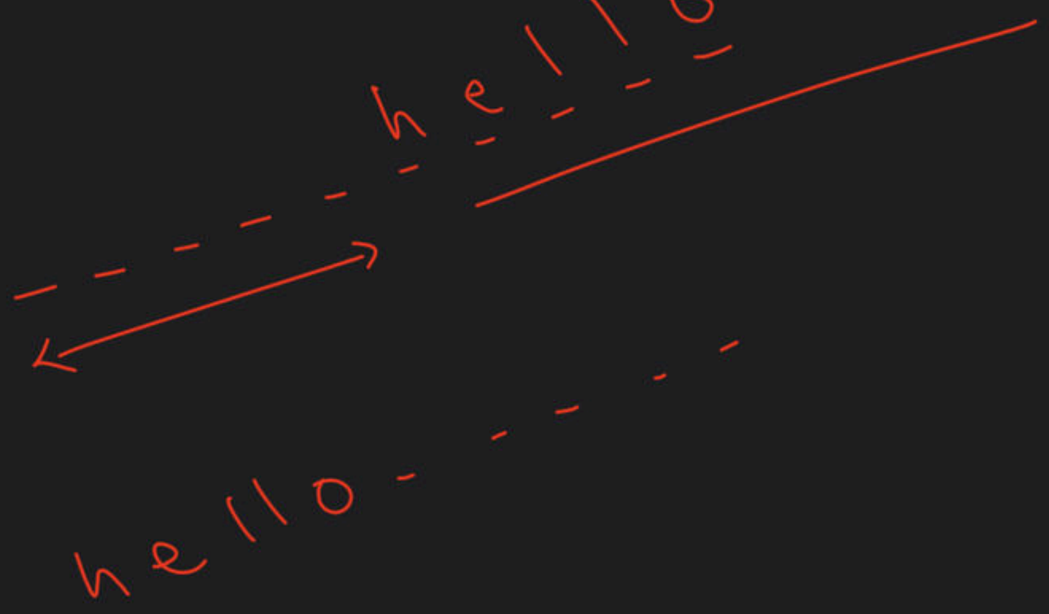
Hru

5

hello

unacademy

```
Str = "hello"
Print(Str.rjust(10))
```

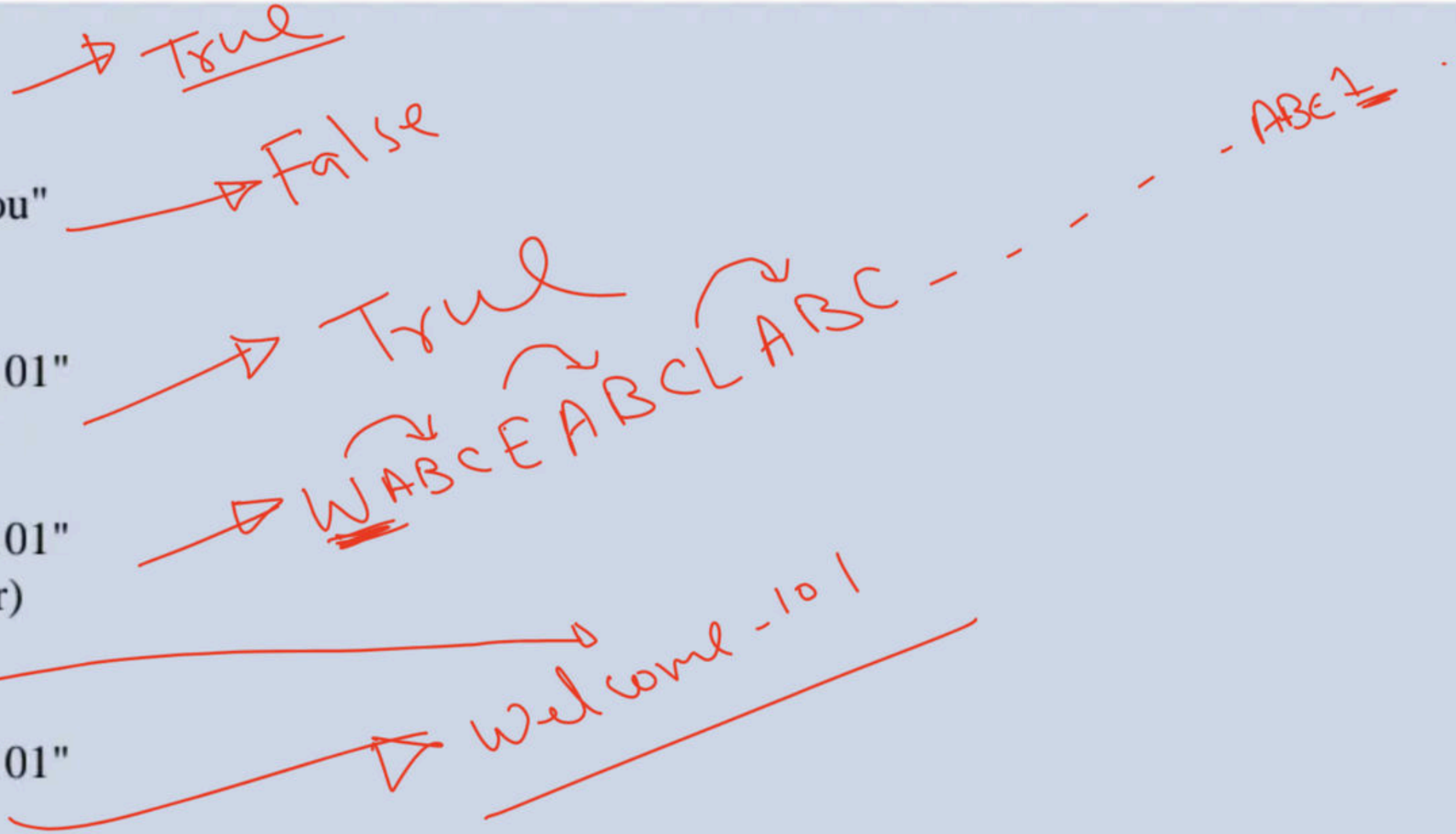


unacademy

Let's Try…

Guess Output?

```
str="  "
print(str.isspace())

str="Welcome to You"
print(str.istitle())

str="WELCOME_101"
print(str.isupper())

str="WELCOME_101"
str1="ABC".join(str)
print(str1)

str="WELCOME_101"
print(str.lower())
```



unacademy

String Methods

lstrip( ):

```
str="     hello"
print(str.lstrip)
str = ",,,,,ssaaww....hello"
print(str.lstrip(",.asw"))
```

maketrans( ):

```
str="Hello Hi"
s= str.maketrans("e","a")
print(str.translate(s))
```

partition( ):

```
str="Hello How Are You"
print(str.partition("How"))
```

97361

unacademy

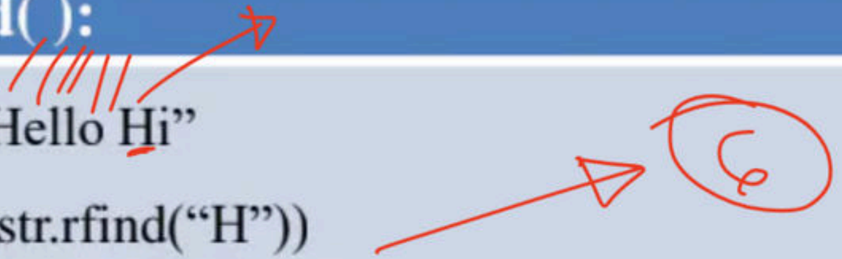
String Methods

replace( ):

```
str=“ hello”

print(str.replace(“hello”,”Welcome”)
```

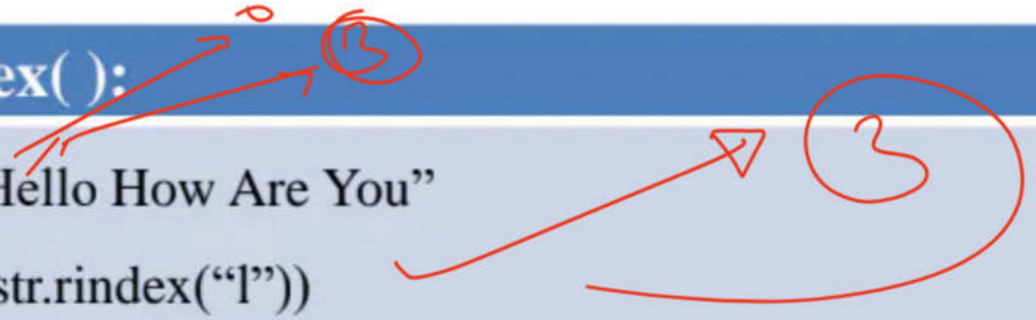
rfind( ):

```
str=“Hello Hi”

print(str.rfind(“H”))
```



rindex( ):

```
str=“Hello How Are You”

print(str.rindex(“l”))
```



unacademy

Let's Try…

Guess Output?

1.
```
str="    Hello —Hi"
print(str.lstrip())
```

2.
```
str="Hello Hi"
print(str.partition(' '))
```

3.
```
str="Hello hi"
print(str.replace("Hi","Jhon"))
```

4.
```
str="Hello hi"
print(str.rfind("L"))
```

5.
```
str="Hello hi"
print(str.rindex("h"))
```

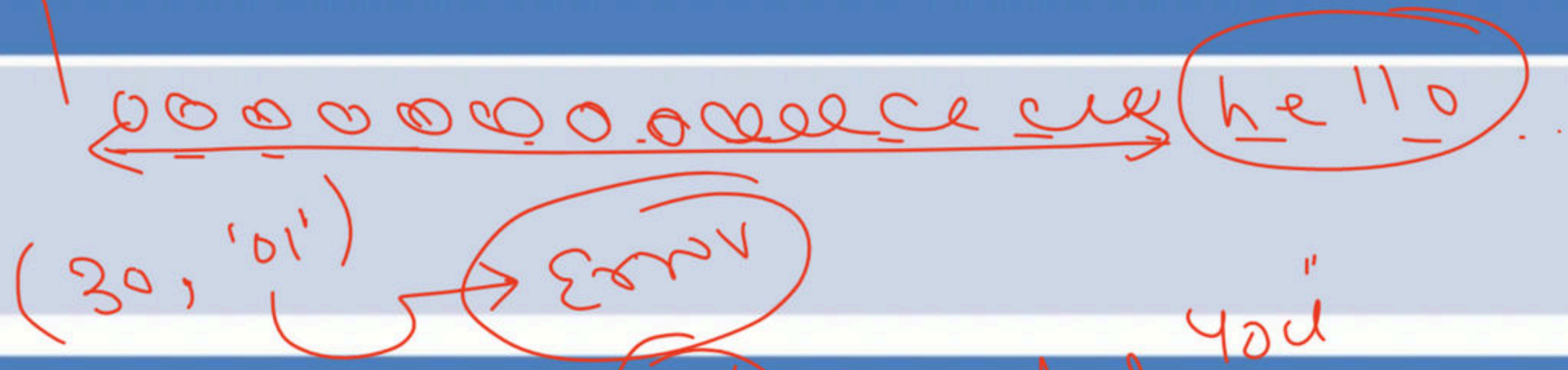
unacademy

String Methods

rjust( ):

str=“ hello”

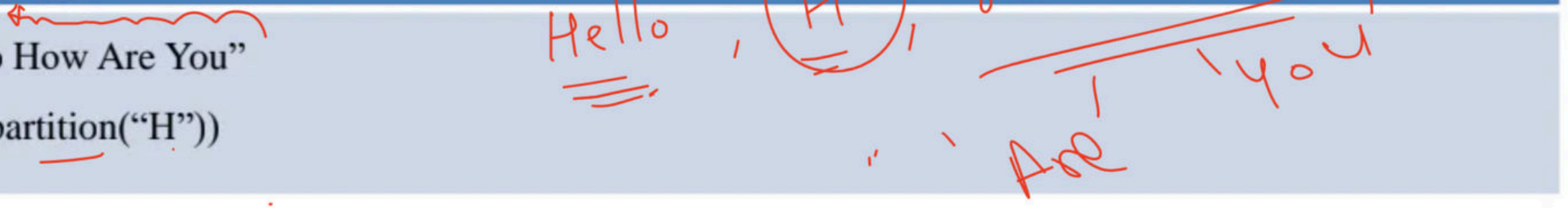
print(str.rjust(30,”0”))



rpartition( ):

str=“Hello How Are You”

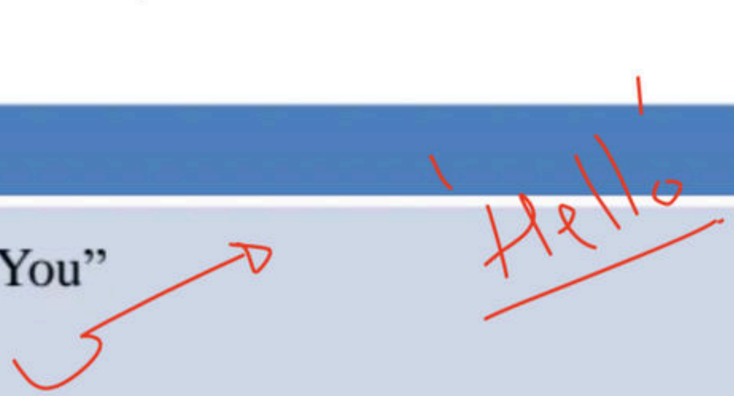
print(str.rpartition(“H”))



rsplit( ):

str=“Hello How Are You”

print(str.rsplit(“ ”))



69078

unacademy

String Methods

rstrip( ):

```
str="hello        "
print(str.rstrip())
```

split( ):

```
str="Hello How Are You"
print(str.split(" "))
```

splitlines( ):

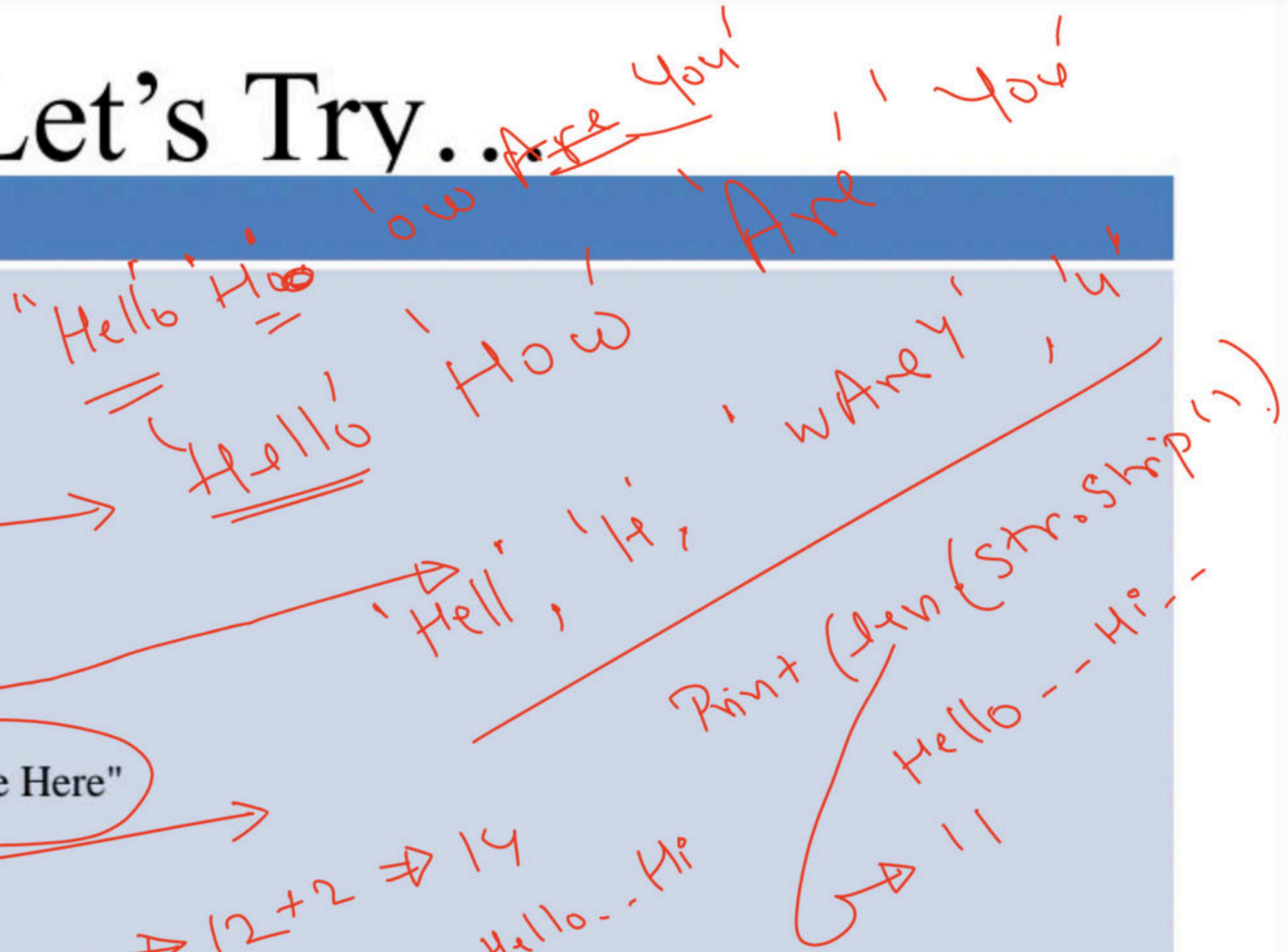
```
str="Hello How Are You\n Welcome to Python Training"
print(str.splitlines())
```

unacademy

Let's Try...

Guess Output?

```
str="Hello How Are You"
print(str.rpartition('H'))

str="Hello How Are You"
print(str.split(' '))

str="Hello How Are You"
print(str.rsplit('o'))

str="Hello How Are You\nWelcome Here"
print(str.splitlines())

str="___Hello__Hi__"
print(len(str))
print(str.rstrip())
print(len(str.rstrip()))
```



unacademy

String Methods

startswith( ):

```
str="hello how are you"
print(str.startswith('h'))
```

swapcase( ):

```
str="Hello How Are You"
print(str.swapcase())
```

strip( ):

```
str="      Hello How Are You      "
print(str.strip())
```

unacademy

String Methods

title( ):

```
str="hello how are you"
print(str.title())
```

upper( ):

```
str="Hello How Are You"
print(str.upper())
```

zfill( ):

```
str="Hello How Are You"
print(str.zfill(15))
```

unacademy

Let's Try…

Guess Output?

```
str="Hello How Are You"
print(str.startswith('h'))

str="hELlO aRe yOu"
print(str.swapcase())

str="hELlO aRe yOu"
print(str.title())

str="hELlO aRe yOu"
print(str.upper())

str="hELlO aRe yOu"
print(str.zfill(10))
```

unacademy

```
str="hello"
print(str.replace("hell0","Welcome"))
#str="Hello Hi"
#print(str.rfind("H"))
#str="Hello Hi"
#print(str.rindex("H"))
#str="hello"
#print(str.rjust(30,"0"))
```

unacademy

```
#str="Hello How Are You"
#print(str.rpartition("H"))
#str="Hello How Are You"
#print(str.rsplit(" "))
#str="hello          "
#print(str.rstrip())
#str="Hello How Are You"
#print(str.split(" "))
#str="How Are You\n Welcome to Python Training"
#print(str.splitlines())
#str="hello how are you"
#print(str.startswith('e'))
```

unacademy

```
#str="Hello How Are You"
#print(str.swapcase())
#str="       Hello How Are You       "
#print(str.strip())
#str="hello how are you"
#print(str.title())
#str="Hello How Are You"
#print(str.upper())
str="HELlO HOW ARE YOU"
print(str.zfill(10))
```

unacademy

Thank You…

unacademy

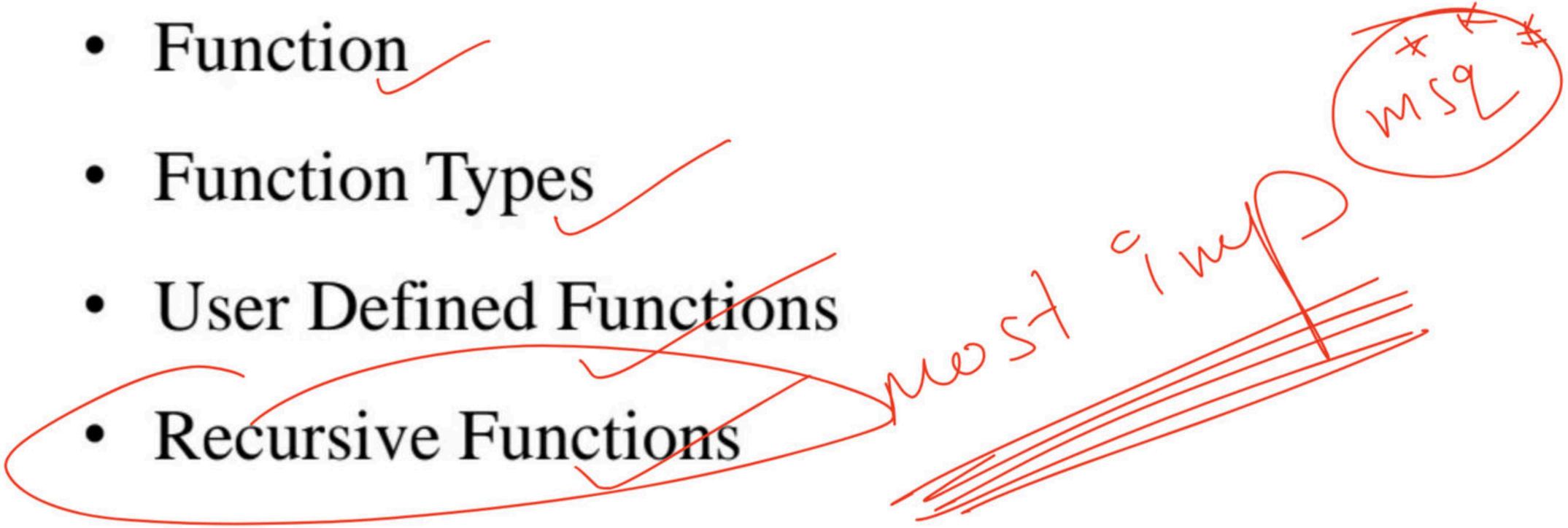
Python Programming

Function and It's Types

unacademy

Content

- Function
- Function Types
- User Defined Functions
- Recursive Functions
- Sample Problems



unacademy

Function

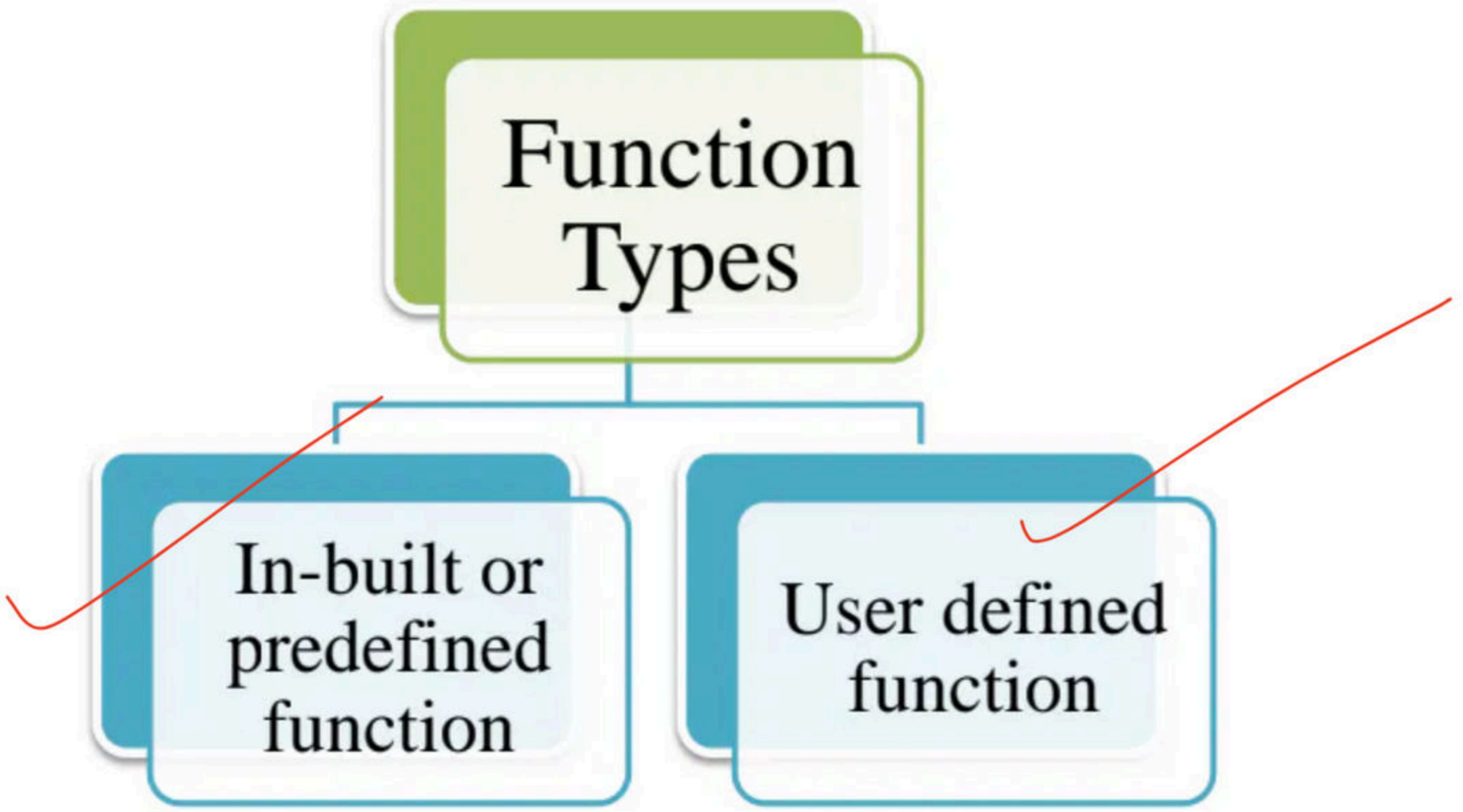
A function is group of statements which are executed when it is called.

In-built Function Examples:	User Defined Function Example:
Name='Jhon' Qualification="Master of Technology" print(len(Name)) print(len(Qualification))	Def power(x,y): print(x**y) x,y=input('Enter two numbers: ').split(' ') x=int(x) y=int(y) power(x,y)

unacademy

Function Types

In python, two types of function are exist.



unacademy

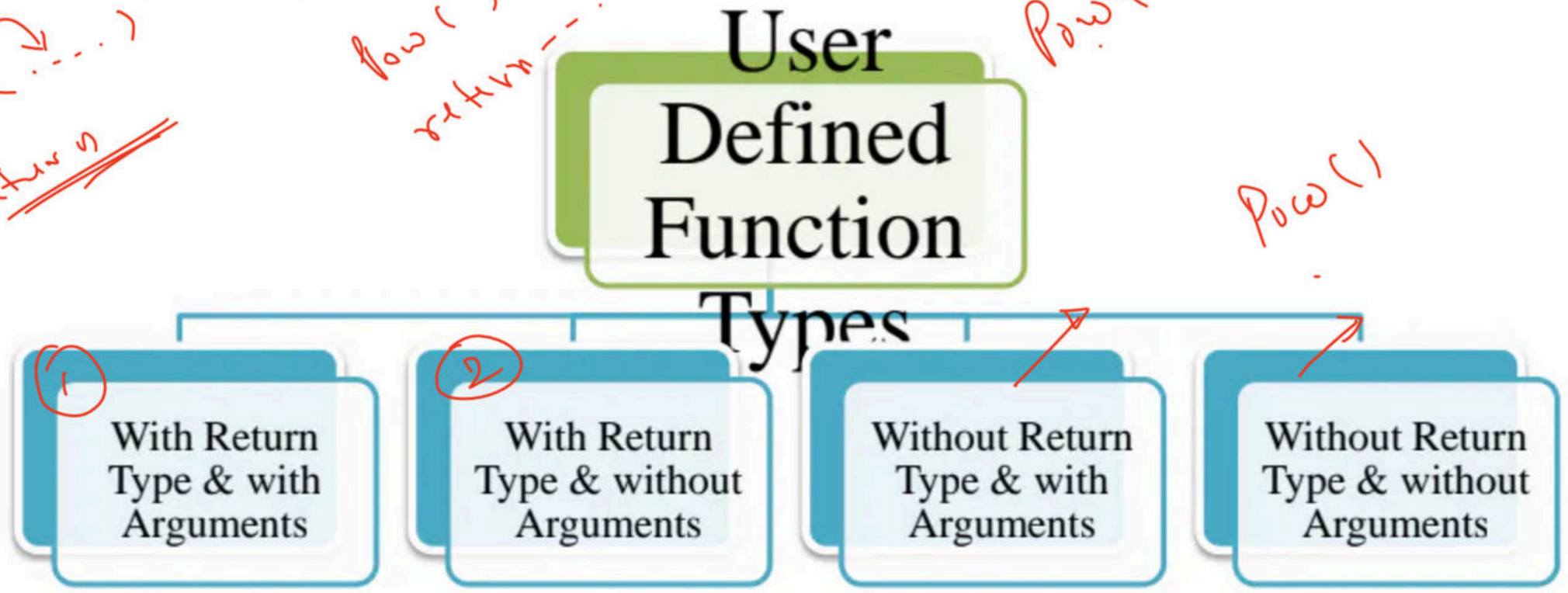
In-built Functions

There are many pre-defined or in-built functions.

General	String	List/Set/Tuples	Dictionary
Print()	Find()	Count()	Clear()
Len()	Index()	Remove()	Copy()
	Capitalize ()	Union()	Items()
	Lower()	Pop()	Keys()
	Strip()	Index()	Get()
	Isalnum ()		Pop()
	Isalpha()		Update()
	Center()		
	Isdigit()		

User Defined Functions

In python, we can also write our own function for specific needs i.e. known as user defined function.



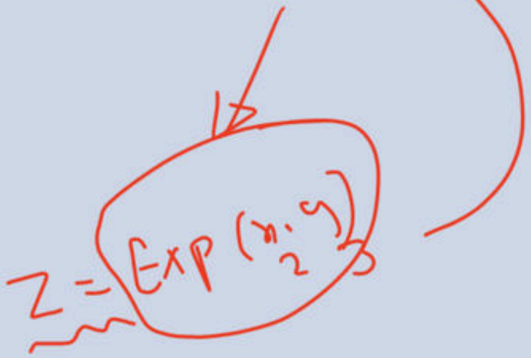
With Return Type & with Arguments

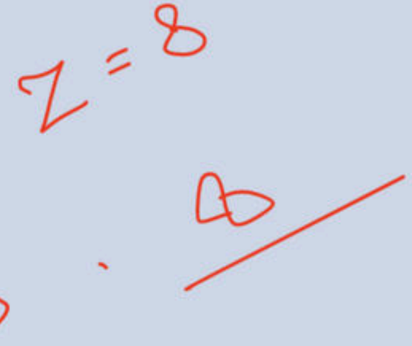
Example

```
# Python Code for With Return types and with arguments
def Exp(x, y):
 return (x**y)
x=int(input('Enter first number:'))
y=int(input('Enter second number:'))
z=Exp(x,y)
print('Exponential is: ', z)
```

Output:

```
Enter first number:10
 Enter second number:4
 Exponential is: 10000
```





With Return Type & without Arguments

Example

```
# Python Code for With Return types and without arguments
def Exp():
  x=int(input('Enter first number:'))
  y=int(input('Enter second number:'))
  return (x**y)
z=Exp()
print('Exponential is: ', z)
```

Output:

```
Enter first number:15
Enter second number:3
Exponential is: 3375
```

Without Return Type & with Arguments

Example

```
# Python Code for Without Return types and with arguments
def Exp(x, y):
    print('Exponential is: ',x**y)
x=int(input('Enter first number:'))
y=int(input('Enter second number:'))
Exp(x,y)
```

Output:

```
Enter first number:15
Enter second number:4
Exponential is: 50625
```

Without Return Type & without Arguments

Example

```
# Python Code for Without Return types and without arguments
def Exp():
  x=int(input('Enter first number:'))
  y=int(input('Enter second number:'))
  print('Exponential is: ',x**y)
Exp()
```

Output:

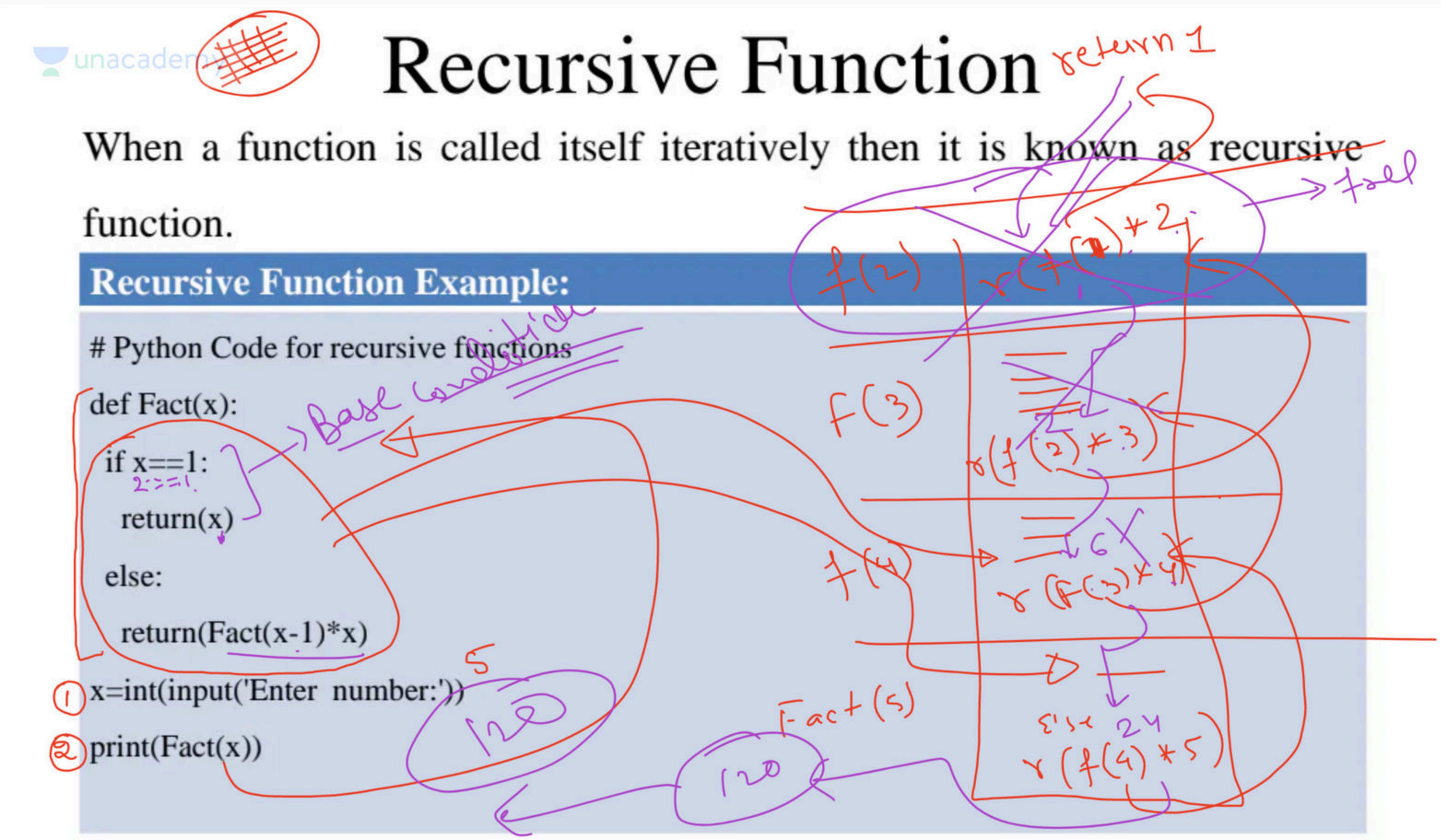
```
Enter first number:8
Enter second number:2
Exponential is: 64
```

Recursive Function

When a function is called itself iteratively then it is known as recursive function.

Recursive Function Example:

```
# Python Code for recursive functions
def Fact(x):
  if x==1:
    return(x)
  else:
    return(Fact(x-1)*x)
x=int(input('Enter  number:'))
print(Fact(x))
```



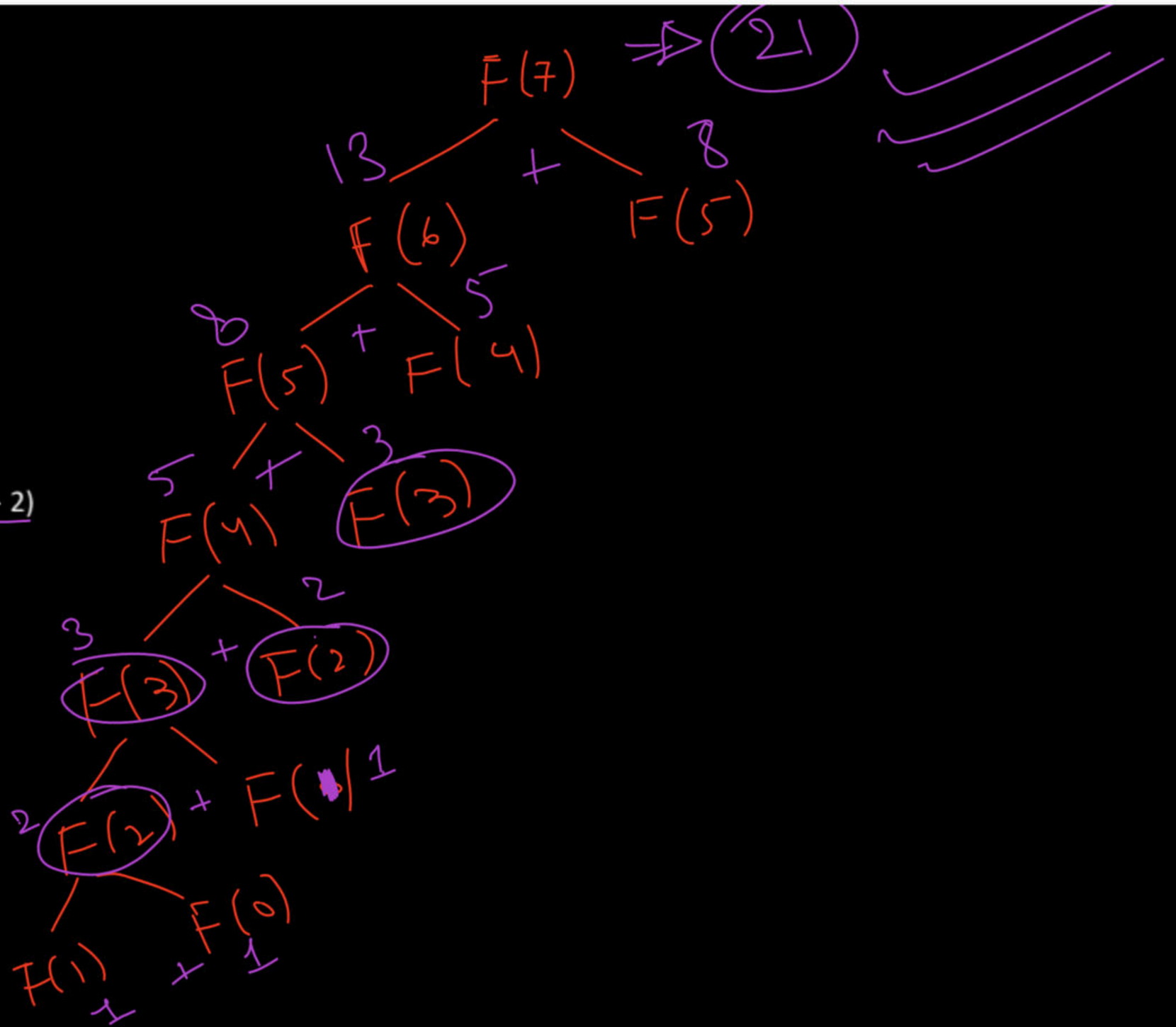
unacademy

Expected Questions
from
Python Programming

unacademy

What is output?

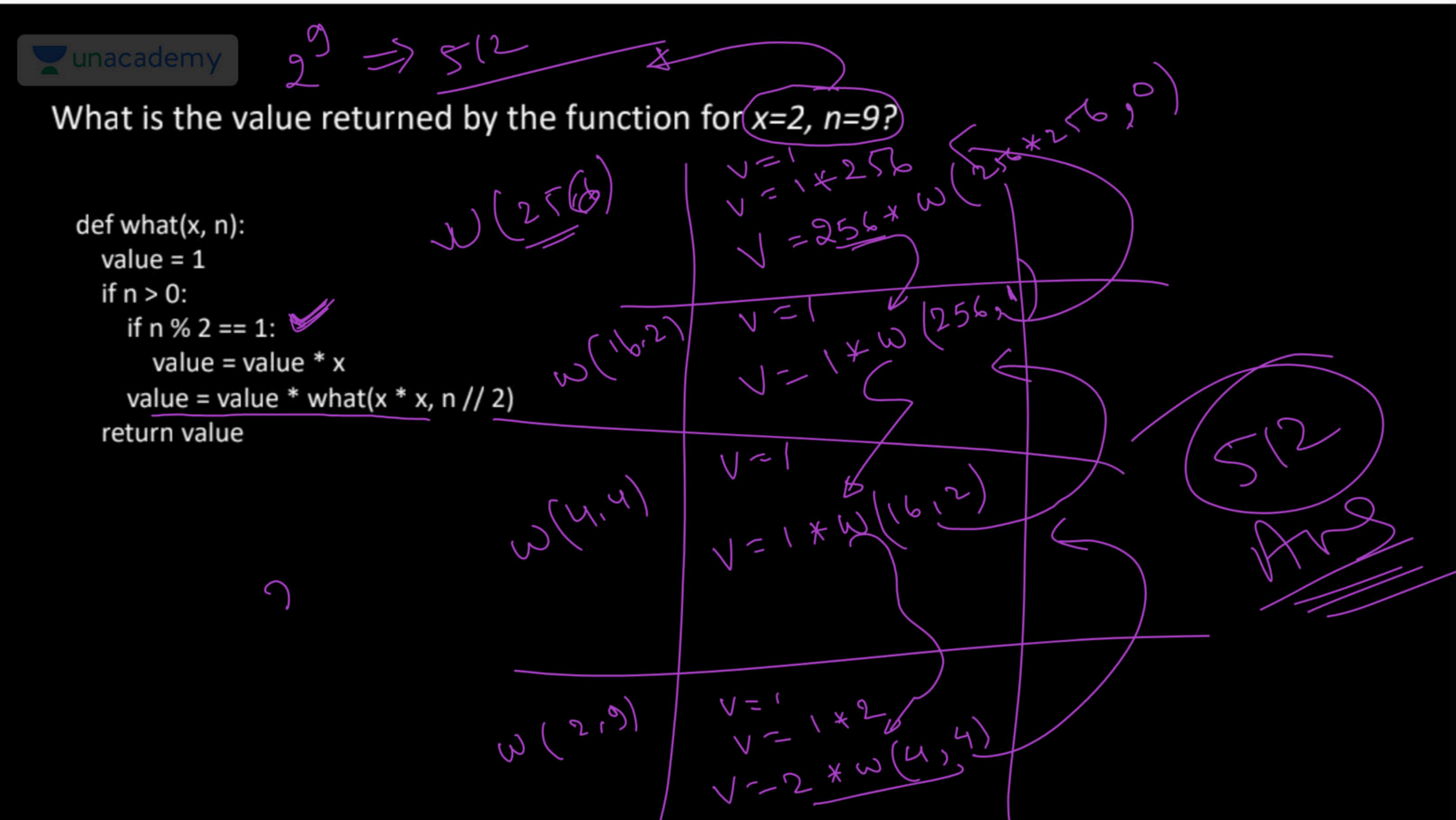
```
def fib(n):
    if n == 0 or n == 1:
        return 1
    else:
        return fib(n - 1) + fib(n - 2)

result = fib(7)
print("fib(7) =", result)
```



unacademy

What is the value returned by the function for *x=2, n=9*?

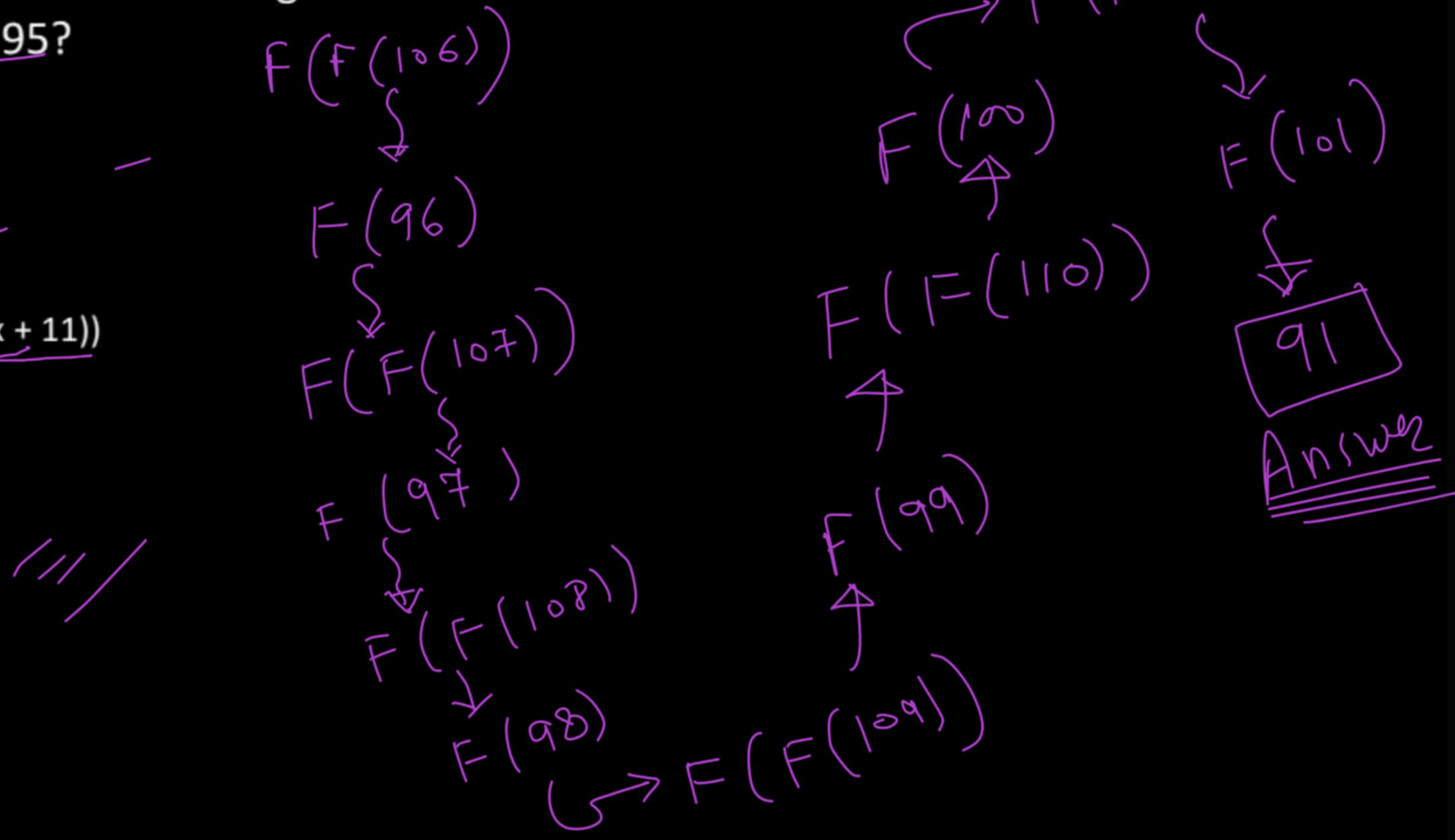
```
def what(x, n):
    value = 1
    if n > 0:
        if n % 2 == 1:
            value = value * x
        value = value * what(x * x, n // 2)
    return value
```



unacademy

What value would the following function return for the input x = 95?

```
def fun(x):
    if x > 100:
        return x - 10
    else:
        return fun(fun(x + 11))
```



unacademy

Consider the following python function definition

```
def Trial(a, b, c):
    if a >= b and c < b:
        return b
    elif a >= b:
        return Trial(a, c, b)
    else:
        return Trial(b, a, c)
```

The function trial
(A) finds the maximum of a, b and c
(B) finds the minimum of a, b and c
(C) finds the middle number of a, b, c
(D) none of the above

unacademy

What value return by j at the end of program ?

```
def incr(i):
    if not hasattr(incr, 'count'):
        incr.count = 0
    incr.count += i
    return incr.count

def main():
    i, j = 0, 0
    for i in range(5):
        j = incr(i)
        print(f"i = {i}, j = {j}")

if __name__ == "__main__":
    main()
```

unacademy

The value returned by **f(1) is**

```
def f(n):
    if n >= 5:
        return n
    global i
    n = n + i
    i += 1
    return f(n)
```

unacademy

```
def foo(n, sum):
    if n == 0:
        return
    k = n % 10
    j = n // 10
    sum += k
    foo(j, sum)
    print(f"{k},", end="")
def main():
    a = 2048
    sum = 0
    foo(a, sum)
    print(f"{sum}")
if __name__ == "__main__":
    main()
```

What does the above program print?

(A) 8, 4, 0, 2, 14 (B) 8, 4, 0, 2, 0

(C) 2, 0, 4, 8, 14 (D) 2, 0, 4, 8, 0

unacademy

```
def f(n):
    global r
    if n <= 0:
        return 1
    if n > 3:
        r = n
        return f(n - 2) + 2
    return f(n - 1) + r
r = 0
result = f(5)
print("Result:", result)
```

unacademy

```
def foo(n, r):
    if n > 0:
        return n % r + foo(n // r, r)
    else:
        return 0
```

1. What is the return value of the function foo when it is called as foo (513, 2)?
2. What is the return value of the function foo when it is called as foo (245, 8)?
3. What is the return value of the function foo when it is called as foo (245, 10)?

unacademy

```
def func(num):
    count = 0
    while num:
        count += 1
        num >>= 1
    return count

num = 513
result = func(num)
print(result)
```

unacademy

What is output ?

```
m = 10
n, n1 = 0, 0
m += 1
n = m
m += 1
n1 = m
n -= 1
n -= n1
print(n)
```

unacademy

```
def jumble(x, y):
    x = 2 * x + y
    return x

def main():
    x = 2
    y = 5
    y = jumble(y, x)
    x = jumble(y, x)
    print(x)

if __name__ == "__main__":
    main()
```

unacademy

```python
def fun(n):
    if n <= 1:
        return n
    else:
        return fun(n - 1) + fun(n - 1)

# Example usage:
result = fun(5)  # Replace 5 with the desired value
print(result)
```

unacademy

```
def fun(n):
    if n <= 1:
        return n
    else:
        return 2 * fun(n - 1)

# Example usage:
result = fun(5)  # Replace 5 with the desired value
print(result)
```

unacademy

Let's Try…

Guess Output?

College="Institute of Engineering and Technology"

print(College[1+1:4+0])

print(College[0:2**3])

print(College[-12:-7+3])

print(College[-1:-7)

print(College[:10])

print(College[:-6])

print(College[2:-2])

print(College[-8:14])

unacademy

Thank You…